



HAL
open science

Modification et couverture de graphes : Approches paramétrées

Maël Dumas

► **To cite this version:**

Maël Dumas. Modification et couverture de graphes : Approches paramétrées. Algorithme et structure de données [cs.DS]. Université d'Orléans, 2023. Français. NNT : 2023ORLE1090 . tel-04638902

HAL Id: tel-04638902

<https://theses.hal.science/tel-04638902>

Submitted on 8 Jul 2024

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

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

UNIVERSITÉ D'ORLÉANS

*École Doctorale Mathématiques, Informatique, Physique
Théorique et Ingénierie des Systèmes*

Laboratoire d'informatique Fondamentale d'Orléans

THÈSE présentée par :

Maël DUMAS

soutenue le : **15 décembre 2023**

pour obtenir le grade de : **Docteur de l'Université d'Orléans**

Discipline / Spécialité : **Informatique**

**Modification et couverture de graphes :
Approches paramétrées**

THÈSE dirigée par :

M. PEREZ Anthony
M. TODINCA Ioan

Maître de Conférences, Université d'Orléans
Professeur, Université d'Orléans

RAPPORTEURS :

Mme BAZGAN Cristina
M. SAU VALLS Ignasi

Professeure, Université Paris Dauphine
Chargé de recherches, CNRS

EXAMINATEURS :

M. CRESPELLE Christophe
Mme LAGOUTTE Aurélie

Professeur, Université Côte d'Azur, *Président du jury*
Maître de Conférences, Université Grenoble Alpes

Remerciements

Je souhaite avant toute chose remercier Ioan et Anthony pour leur encadrement exceptionnel tout au long de ma thèse. Merci pour tout ce que vous m'avez transmis, pour les opportunités que vous m'avez données, pour toujours avoir été là pour m'aider. Travailler avec vous ces années a été un réel plaisir.

Merci à Cristina Bazgan et Ignasi Sau de m'avoir fait l'honneur de rapporter ma thèse, ainsi qu'à Christophe Crespelle et Aurélie Lagoutte pour m'avoir fait égal honneur de faire partie de mon jury. Merci pour vos précieux retours sur mon travail.

Merci à tous les membres LIFO et du département informatique qui ont pu m'aider et avec qui j'ai eu l'occasion d'échanger. Merci à Mathieu pour ton aide avec les enseignements, merci à Isabelle pour toute ton aide administrative.

Merci à Samuel mon co-bureau, sans toi ces années n'auraient pas été aussi plaisantes, merci pour ces après-midi studieuses, pour avoir essayé de rectifier mon inculture, merci d'avoir été là pour m'aider et me soutenir. Merci à Shahrzad pour ta gentillesse, pour tes cours de persan et merci pour la motivation à finir mon manuscrit. Merci également à tous mes autres camarades doctorants du LIFO pour tous ces moments que nous avons partagés au quotidien, merci à Arthur, Aymeric, Darine, Florian, Jordan, Mathieu, Nicolas, Olivier, Placido, Sébastien, Sofiane et Thibaut (liste non-exhaustive par ordre de préférence).

Merci à toutes celles et ceux avec qui j'ai eu la chance collaborer au cours de ma thèse, votre rencontre aura été des plus enrichissantes. Merci à Mathis d'avoir subi les idées d'Anthony avec moi, à Florent pour tes super problèmes, à Clément pour tous tes projets et ton franc-parler. Merci à Claire, Virginia, Marthe, Amadeus et Valentin pour les projets passés ou en cours. Merci également à toutes celles et ceux que j'ai eu la chance de rencontrer à l'occasion de conférences, écoles ou autres rencontres.

Merci aux enseignants de l'université de Montpellier qui ont su éveiller mon intérêt pour l'informatique théorique, et plus particulièrement pour la théorie des graphes et l'algorithmique, je n'en serais pas là aujourd'hui sans vous. Merci notamment à Stéphane Bessy, David Delahaye, Hervé Dicky et Mickael Montassier.

Merci à mes amis. Merci à Marien pour toutes ces longues discussions et pour toujours avoir répondu à mes questions de math. Merci à Julie, pour ton soutien et le partage de nos galères de thèse. Merci à Noah, Galdric, Morgan, Arthur et Maxime pour toutes ces soirées partagées, particulièrement pendant le confinement.

Pour finir, merci à ma famille, à mes parents et à mon frère de toujours être là pour me soutenir.

Table des matières

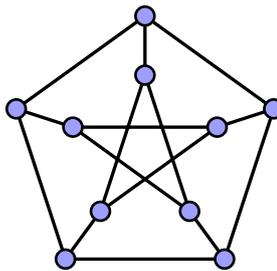
1	Introduction	1
2	Préliminaires	13
2.1	Graphes	13
2.1.1	Définitions générales	13
2.1.2	Classes de graphes	14
2.1.3	Modification d'arêtes	17
2.1.4	Paramètres de graphes	17
2.2	Complexité paramétrée	19
2.2.1	Classes de complexité paramétrée	19
2.2.2	Algorithme de noyau	20
2.2.3	Méta-théorèmes	23
2.3	Analyse de processus de branchement	24
I	Noyaux pour la modification d'arêtes	27
3	Introduction et généralités	29
3.1	Règles de réduction classiques	30
3.2	Reconnaissance et application polynomiale	32
3.3	Approche par décomposition et sommets affectés	33
4	Graphes trivialement parfaits	37
4.1	Définitions et propriétés	38
4.2	Noyau pour TRIVIALY PERFECT EDITION	41
4.2.1	Règles de réduction classiques	42
4.2.2	Peignes : définition et propriétés structurelles	43
4.2.3	Réduction des manches	44
4.2.4	Réduction des dents	48
4.2.5	Réduction exhaustive du graphe	50
4.2.6	Borner la taille d'une instance réduite	52
4.2.7	Noyaux pour la version complétion et délétion	53
4.3	Noyau pour THRESHOLD EDITION	54
4.3.1	Définition et structures	54
4.3.2	Règles de réduction pour THRESHOLD EDITION	55
4.3.3	Borner la taille d'une instance réduite	57
5	Graphes blocs et strictement cordaux	59
5.1	Définitions et propriétés	60
5.2	NP-complétude	64
5.3	Noyau pour BLOCK GRAPH EDITION	65
5.3.1	Règles de réduction classiques	66
5.3.2	Réduction des BG-branches	66
5.3.3	Borner la taille d'une instance réduite	69
5.4	Noyau pour STRICTLY CHORDAL EDITION	70

5.4.1	SC-branches : définition et propriétés structurelles	71
5.4.2	Réduction des 1-SC-branches	73
5.4.3	Réduction des 2-SC-branches	76
5.4.4	Borner la taille d'une instance réduite	78
5.4.5	Noyaux pour les versions complétion et délétion	80
Partie I : conclusion et perspectives		82
II Couverture de graphes		85
6	Couverture par plus courts chemins	87
6.1	Préliminaires et notations	89
6.2	Couverture des arêtes avec k plus courts chemins	91
6.2.1	Borne factorielle	92
6.2.2	Borne simplement exponentielle	94
6.3	Couverture des sommets avec k plus courts chemins	99
6.4	Conséquences algorithmiques	106
6.5	Conclusion et perspectives	107
7	Conclusion et perspectives de la thèse	109
Bibliographie		111

Chapitre 1

Introduction

Les **graphes** sont une structure extrêmement simple qui est fondamentale en informatique. Un graphe est composé d'un ensemble d'éléments, les *sommets*, et de connexions entre ces sommets, les *arêtes*. Ils permettent de modéliser un grand nombre de problèmes pouvant être exprimés comme un ensemble de données ayant des relations entre elles. Nous retrouvons notamment des problèmes de réseaux (par exemple des réseaux routiers, des réseaux sociaux ou encore des réseaux informatiques) ou de représentation de connaissances dans des bases de données. Ils trouvent aussi des applications dans le domaine de la biologie, notamment en génétique ou dans le domaine de la chimie pour la représentation de molécules.



Un premier exemple : le graphe de Petersen.

Une application utilisée dans la vie de tous les jours est par exemple la recherche d'un plus court chemin pour aller d'un point A à un point B . Un autre problème classique est celui du voyageur de commerce. Dans ce problème sont données n villes et l'ensemble des distances entre ces dernières, et il y est recherché la tournée la plus courte possible qui passe au moins une fois par chacune des n villes puis revient à son point de départ, et ce sans contrainte d'ordre sur les villes.

Il existe de nombreux algorithmes qui permettent de trouver un plus court chemin d'un point à un autre de manière efficace. Ce problème est considéré comme « facile » à résoudre dans le sens où le temps nécessaire pour trouver une solution optimale est au plus polynomial en la taille du graphe d'entrée, c'est-à-dire son nombre de sommets et d'arêtes. Cependant, le problème du voyageur de commerce est lui sensiblement plus « difficile » à résoudre. Un algorithme naïf consisterait à tester tous les ordres possibles pour visiter les n villes, ce qui fait $n!$ possibilités. Les meilleurs algorithmes connus prennent un temps exponentiel en n [6, 11, 82], ce qui est préférable à un temps factoriel en n . Toutefois, il n'existe pas à ce jour d'algorithme qui permet de résoudre le problème du voyageur de commerce en un temps polynomial en n .

Il semblerait donc que certains problèmes n'admettent pas d'algorithme polynomial pour les résoudre. Cela avait notamment été pressenti dans une longue digression de Jack Edmonds dans un papier datant de 1965 [62]. Il y constate que pour certains problèmes l'existence d'algorithmes s'exécutant en temps polynomial en la taille de l'entrée n'est en aucun cas évidente et suggère l'étude de la question suivante :

Does there or does there not exist an algorithm of given order of difficulty for a given class of problems?

La différence de difficulté entre les problèmes est l'un des objets d'étude de la **théorie de la complexité**. Elle permet d'organiser les problèmes par classe de complexité en fonction du temps de calcul et de l'espace nécessaires pour les résoudre. Le temps et l'espace sont mesurés en fonction de la taille de l'entrée, dans les graphes cela correspond au nombre n de sommets et au nombre m d'arêtes. Deux classes de complexité fondamentales sont P et NP :

- P est la classe des problèmes pour lesquels on peut décider si une solution existe en temps polynomial.
- NP est la classe des problèmes pour lesquels on peut vérifier qu'une solution proposée en est bien une en temps polynomial.

Ces classes de complexité concernent les problèmes de décision, qui sont les problèmes dont la réponse est soit « oui », soit « non ». Le problème du voyageur de commerce est lui un problème d'optimisation : la tournée la plus courte possible est recherchée. Néanmoins nous pouvons reformuler ce problème comme un problème de décision en remplaçant l'objectif par la question : « existe-t-il une tournée passant par les n villes de longueur au plus D ? ». Cette version du problème appartient à la classe de complexité NP, en effet, étant donné une tournée, il est aisé de vérifier qu'elle passe par les n villes et qu'elle est de longueur au plus D en temps polynomial. Notons tout de même que dans la pratique, nous sommes surtout intéressés par des problèmes d'optimisation, des solutions « optimales » sont recherchées.

Il est clair que la classe P est incluse dans la classe NP, mais nous ne savons pas si NP est inclus dans P et donc si $P = NP$ ou si $P \neq NP$.

Conjecture. $P \neq NP$: Il existe un problème de la classe NP qui n'appartient pas à la classe P.

C'est sur cette conjecture que se fonde un grand pan de la théorie de la complexité. Cette conjecture est encouragée par des résultats du début des années 1970 de Cook [38] et Levin [98] qui ont montré qu'il existe des problèmes appartenant à NP qui sont au moins aussi difficiles que tous les autres problèmes de cette classe. Cela implique que si ces problèmes admettent un algorithme polynomial pour les résoudre, alors tous les problèmes de NP en admettent un. Les problèmes qui sont au moins aussi difficiles que tous ceux de NP sont dit *NP-difficiles*, si en plus ils sont dans NP, alors ils sont dit *NP-complets*.

Depuis Cook et Levin, de nombreux problèmes ont été montrés NP-complets, dès 1972 où Karp a démontré la NP-complétude d'un ensemble de 21 problèmes [85]. Parmi ces problèmes, nous retrouvons notamment des problèmes de graphes classiques tels que :

- VERTEX COVER : Existe-t-il un ensemble de k sommets qui couvrent toutes les arêtes ?
- INDEPENDENT SET : Existe-t-il un ensemble de k sommets deux à deux non adjacents ?
- CHROMATIC NUMBER : Est-il possible de colorer les sommets du graphe avec k couleurs de manière à ce que deux sommets adjacents n'aient pas la même couleur ?
- HAMILTONIAN CYCLE : Existe-t-il un cycle passant par tous les sommets du graphe exactement une fois ?

Pour montrer qu'un problème A est NP-difficile, il suffit de montrer qu'il est au moins aussi difficile à résoudre qu'un problème B déjà connu NP-difficile. Par exemple, un algorithme résolvant le problème décisionnel du voyageur de commerce permet aussi de résoudre le problème HAMILTONIAN CYCLE, si des modifications (en temps polynomial) ont été effectuées au préalable sur le graphe d'entrée pour l'adapter au problème. Cela implique que le problème du voyageur de commerce décisionnel est lui aussi NP-complet. Notons que si NP porte sur des problèmes de décision, les problèmes d'optimisation qui sont au moins aussi durs que tous les problèmes de NP sont eux aussi qualifiés de NP-difficiles usuellement. En résumé, qu'un problème soit NP-difficile signifie qu'il n'existe a priori pas d'algorithmes efficaces pour le résoudre.

Malgré leur difficulté apparente, il y a un besoin clair de pouvoir résoudre les problèmes NP-difficiles aussi efficacement que possible. Pour cela différentes approches peuvent être envisagées. La première est simplement de chercher les meilleurs algorithmes exponentiels possibles pour résoudre ces problèmes de façon exacte. Par exemple pour le problème MINIMUM VERTEX COVER

un algorithme naïf testant toutes les 2^n solutions possibles prend un temps $O^*(2^n)$ ¹, mais actuellement le meilleur algorithme connu, basé sur le principe « mesurer et conquérir », prend un temps $O^*(1.1996^n)$ [128]. Une seconde approche est de chercher non pas une solution exacte, mais une solution approchée en temps polynomial. Pour le problème MINIMUM VERTEX COVER, il existe un algorithme simple qui garantit une solution au plus deux fois plus grande que celle optimale. Il consiste à prendre les deux extrémités d'une arête, les ajouter dans la solution, puis de les supprimer, et ce tant que le graphe contient une arête.

En pratique, on retrouve beaucoup de méthodes que l'on qualifie habituellement d'heuristiques : elles n'ont pas nécessairement des garanties théoriques sur le temps d'exécution ou sur l'optimalité des solutions trouvées, mais leur efficacité peut être observée dans la pratique. Par exemple pour le problème MINIMUM VERTEX COVER, une heuristique simple est de choisir en premier dans la solution un sommet de grand degré, intuitivement, un tel sommet a de grandes chances d'être dans une solution optimale ; cette approche *gloutonne* peut être généralisée à de nombreux problèmes. Les techniques de programmation linéaire en nombres entiers ou encore de programmation par contraintes fournissent en pratique de bons résultats. À titre d'exemple, en combinant diverses méthodes, au début des années 2000, le problème du voyageur de commerce a pu être résolu sur des instances de plus de 15 000 sommets, correspondant aux villes d'Allemagne [7].

Une dernière approche est celle de la **complexité paramétrée**, qui a été formalisée par Downey et Fellows [51] dans les années 1990. Avec cette approche on s'intéresse à la complexité d'un problème par rapport à sa taille, mais également par rapport à un autre paramètre, supposé petit. C'est cette approche que nous adoptons pour étudier les problèmes considérés dans cette thèse. Différents paramètres peuvent être considérés, par exemple un paramètre naturel pour de nombreux problèmes est la taille de la solution. Ce paramètre correspond à la taille de l'ensemble de sommets sélectionnés dans VERTEX COVER ou au nombre de couleurs autorisées dans CHROMATIC NUMBER. D'autres paramètres liés à la structure de l'instance peuvent être considérés, pour un graphe par exemple, son degré maximum, la taille de sa couverture par sommets minimum, sa largeur arborescente ou linéaire, ou encore sa profondeur d'arbre. Nous définirons ces paramètres dans le prochain chapitre. Une instance d'un problème paramétré est notée (x, k) où x est l'entrée du problème non paramétré, un graphe par exemple, et $k \in \mathbb{N}$ un paramètre indépendant de la taille de x . L'objectif dans le cadre de la complexité paramétrée est de développer des algorithmes dont les temps d'exécution ont une faible dépendance en la taille de l'entrée x , mais pouvant avoir une forte dépendance en k , le paramètre.

Définition. Un problème paramétré admettant un algorithme pour le résoudre en temps $f(k) \cdot \text{poly}(|x|)$ est dit **fixed-parameter tractable** (résoluble efficacement à paramètre fixé).

Le temps d'exécution d'algorithmes fixed-parameter tractable a une dépendance polynomiale en la taille de l'entrée x mais qui peut être exponentielle en fonction du paramètre k . Par exemple le problème VERTEX COVER, paramétré par la taille k de la solution, admet un algorithme simple qui s'exécute en temps $2^k \cdot \text{poly}(n)$ et est donc fixed-parameter tractable (cet algorithme est détaillé dans le chapitre 2). La classe des problèmes paramétrés fixed-parameter tractable est appelée **FPT**. Qu'un problème paramétré soit dans FPT signifie qu'il peut être résolu efficacement si le paramètre est petit.

Cependant, à l'instar des problèmes NP-difficiles, il existe des problèmes paramétrés qui sont conjecturés comme n'étant pas dans FPT. C'est le cas par exemple des problèmes INDEPENDENT SET paramétré par la taille de la solution et CHROMATIC NUMBER paramétré par le nombre de couleurs autorisées. Pour le premier problème il existe néanmoins un algorithme paramétré qui s'exécute en temps $n^{f(k)}$, ce problème est dans la classe de complexité paramétrée appelée XP. Le second problème quant à lui n'est même pas dans XP et ne peut a priori pas admettre d'algorithme ayant une dépendance polynomiale en la taille de l'entrée, en effet, savoir si un graphe peut être coloré par k couleurs est NP-complet dès que $k \geq 3$.

Remarquons que des problèmes polynomialement équivalents en complexité classique peuvent ne pas l'être en complexité paramétrée. C'est le cas par exemple des problèmes VERTEX COVER et INDEPENDENT SET, où une solution de taille k pour l'un donne une solution complémentaire de taille $n - k$ pour l'autre problème. Cependant, si nous souhaitons résoudre le problème INDEPENDENT SET, paramétré par la taille k de la solution, à l'aide de l'algorithme fixed-parameter

¹La notation O^* omet le facteur polynomial d'une fonction exponentielle, par exemple $2^n \cdot \text{poly}(n) = O^*(2^n)$.

tractable pour VERTEX COVER, son temps d'exécution est $2^{n-k} \cdot \text{poly}(n)$. Le temps d'exécution étant exponentiel en n cela ne permet pas de montrer que INDEPENDENT SET est FPT, et nous rappelons en particulier que ce problème est conjecturé comme ne l'étant pas.

Pour finir, nous présentons un type d'algorithmes paramétrés auxquels nous apportons une attention particulière dans cette thèse : les **algorithmes de noyaux**, plus simplement appelés **noyaux**. Ce sont des algorithmes de pré-traitement qui permettent de réduire une instance d'entrée (x, k) en temps polynomial en une instance équivalente (x', k') ayant une taille bornée supérieure-ment par une fonction $g(k)$. Cette fonction $g(k)$ qui ne dépend que du paramètre k est appelée la *taille* du noyau. Les deux instances sont dites équivalentes dans le sens où (x', k') répond positivement au problème si et seulement si (x, k) aussi.

L'approche classique pour mettre au point un noyau est d'analyser la structure de l'instance pour essayer de détecter et simplifier des parties de cette dernière qui contraignent soit faiblement soit fortement les éventuelles solutions. Prenons par exemple le cas du problème VERTEX COVER paramétré par la taille de la solution. Pour une instance (G, k) , nous pouvons faire deux observations simples qui permettent de réduire cette instance et de définir ce que nous appelons des règles de réduction :

- Un sommet qui n'est incident à aucune arête n'a besoin d'être dans aucune solution.
Règle 1. Si G contient un sommet isolé v , retirer ce sommet du graphe.
- Un sommet qui a au moins $k + 1$ voisins doit être dans toute solution de taille au plus k , si ce n'était pas le cas ses $k + 1$ voisins devraient être dans la solution.
Règle 2. Si un sommet a au moins $k + 1$ voisins, retirer ce sommet et décrémenter k de 1.

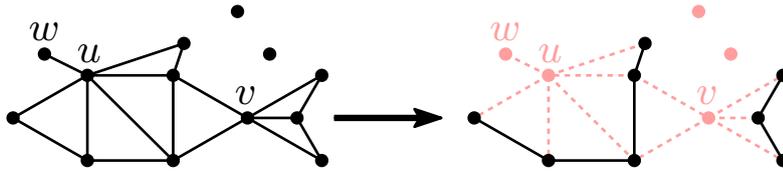


Figure 1.1: Une instance $(G, 5)$ (à gauche) et $(G', 3)$ son instance réduite par les règles 1 et 2 (à droite). La règle 1 a été appliquée sur les sommets u et v . Le sommet w devient isolé après la suppression du sommet u et peut donc être retiré à l'aide de la règle 2. Cette règle permet également de retirer les deux sommets isolés.

Ces deux règles sont suffisantes pour définir un noyau pour le problème VERTEX COVER de taille $k^2 + k$. En effet, soit une instance (G, k) où aucune de ces deux règles ne peut être appliquée. Chaque sommet est adjacent à au plus k autres sommets. Si le graphe G admet une solution de taille k , alors il ne peut pas contenir plus de $k^2 + k$ sommets. En particulier si le graphe contient strictement plus que $k^2 + k$ sommets, nous pouvons conclure que l'instance (G, k) ne peut pas admettre de solution de taille au plus k .

Notons que les algorithmes paramétrés que nous avons mentionnés pour VERTEX COVER ne sont pas les meilleurs connus. En effet, il existe pour ce problème un algorithme fixed-parameter tractable qui s'exécute en temps $O(1.2738^k + kn)$ [36] et un noyau de taille $2k - c \cdot \log k$ [97] pour n'importe quelle constante $c > 0$.

Une dernière remarque est que la notion de noyau est une définition alternative des algorithmes fixed-parameter tractable. En particulier, un problème est FPT si et seulement s'il admet un noyau [30]. Néanmoins, cela n'indique rien sur la taille de ce noyau qui peut être exponentielle. La question pertinente pour les problèmes FPT est donc de savoir s'ils admettent un noyau de taille polynomiale. Malheureusement des résultats mettent en évidence que ce n'est pas toujours le cas sous certaines hypothèses de complexité classiques [17, 20].

Cette thèse est divisée en deux parties, chacune dédiée à un type différent de problèmes de graphes abordés sous l'angle de la complexité paramétrée. Dans la première partie nous étudions l'existence de noyaux polynomiaux pour des problèmes de modification de graphes. Dans la deuxième partie nous étudions l'existence d'algorithmes fixed-parameter tractable pour des problèmes

de couverture de graphes par des plus courts chemins. Dans ce qui suit nous introduisons ces deux types de problèmes et faisons un bref aperçu des résultats connus pour ces questions dans le domaine de complexité paramétrée.

Modification de graphes

Les problèmes de modification de graphes sont des problèmes où, étant donné en entrée un graphe quelconque, nous cherchons à le transformer en un nouveau graphe satisfaisant certaines propriétés en effectuant un minimum de modifications. Les opérations de modifications les plus classiques sont : retirer des sommets, retirer des arêtes ou ajouter des arêtes. De nombreux problèmes classiques sont ou peuvent être vus comme des problèmes de modification de graphes. Par exemple, le problème `MINIMUM VERTEX COVER` qui peut être défini comme la recherche du nombre minimum de sommets à supprimer afin que le graphe obtenu ne contienne aucune arête. Un autre exemple de problème classique est le `MINIMUM FEEDBACK VERTEX SET` où nous cherchons à obtenir un graphe sans cycle (une forêt) en retirant des sommets du graphe initial. Enfin, un exemple plus proche de ceux qui seront étudiés dans cette thèse est le problème du `MINIMUM FILL-IN`, dans lequel nous cherchons à ajouter un minimum d'arêtes pour obtenir un graphe cordal, aussi appelé graphe triangulé (illustré dans la [figure 1.2](#)). Ce sont les graphes dont tous les cycles d'au moins quatre sommets possèdent une arête (appelée corde) reliant deux sommets non consécutifs.

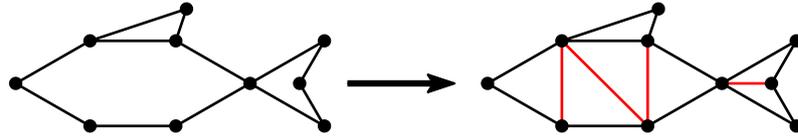


Figure 1.2: Exemple d'ajout d'arêtes à un poisson pour qu'il soit cordal.

Pour la plupart des classes de graphes classiques, les problèmes de modification d'un graphe quelconque afin d'obtenir un graphe de cette classe sont NP-complets. Si nous cherchons uniquement à supprimer des sommets, Lewis et Yannakakis [99] ont montré que le problème est NP-complet pour toute classe de graphes *héréditaire* non triviale. Une classe est héréditaire si, pour un graphe appartenant à cette classe, tous ses sous-graphes induits y appartiennent également, et une classe est *non triviale* si elle contient et ne contient pas un nombre infini de graphes. Bien que cela ne couvre pas toutes les classes de graphes, la plupart de celles étudiées dans la littérature sont héréditaires, certaines exceptions étant par exemple les classes ayant des contraintes de connexité, de degré ou de densité.

Pour les problèmes où nous cherchons à modifier (ajouter ou supprimer) des arêtes, il n'existe pas de résultat aussi général, mais ils ont été montrés NP-complets pour de nombreuses classes de graphes. Les rares exemples de classes pour lesquels ces problèmes sont faciles sont des graphes ayant une structure très simple comme par exemple les arbres et les unions de cliques disjointes. Un aperçu des classes pour lesquelles ces problèmes sont connus difficiles a été proposé par Burzyn *et al.* [26].

Dans le domaine de la complexité paramétrée, ces problèmes ont été principalement étudiés avec le paramètre « nombre de modifications autorisées ». Ce sera ce paramètre que nous utiliserons. Commençons par un résultat majeur, montré par Cai [28].

Théorème 1.1. *Tout problème de modification de graphes vers des classes de graphes héréditaires qui peuvent être caractérisées par des familles finies de sous-graphes induits interdits est FPT.*

Les sous-graphes interdits sont appelés *obstructions*. L'idée pour obtenir un algorithme fixed-parameter tractable pour un tel problème de modification de graphes est simple : tant que le graphe contient une obstruction, nous branchons sur l'ensemble fini des possibilités pour détruire cette obstruction avec les opérations disponibles. Observons qu'il est possible de détecter la présence d'une obstruction (de taille fixée) dans un graphe en temps polynomial, qu'au plus k modifications peuvent être faites, et qu'il y a un nombre constant d de possibilités pour détruire une obstruction. La procédure décrite peut être effectuée en temps $O(d^k) \cdot \text{poly}(n)$ et est donc bien FPT, une analyse détaillée pour obtenir cette complexité est proposée dans la [section 2.3](#).

Pour de nombreuses autres classes, les problèmes ont été montrés FPT, notamment pour les graphes cordaux [33, 69, 109], pour les graphes planaires [87, 111] ou les forêts pour la suppression de sommets [44]. Les classes de graphes pour lesquelles ces problèmes sont connus comme n'étant pas FPT sont beaucoup plus rares, c'est notamment le cas pour la suppression de sommets et d'arêtes vers les graphes sans roues [102]. Un aperçu plus général des résultats FPT pour les problèmes de modification d'arêtes a été proposé par Crespelle *et al.* [41].

Naturellement, comme les problèmes de modification de graphes sont en grande partie FPT, l'existence de noyaux polynomiaux a été particulièrement étudiée pour ces derniers. Pour les problèmes de suppression de sommets, les classes héréditaires caractérisées par des familles finies de sous-graphes induits interdits, admettent un noyau polynomial. En effet, ces problèmes se ramènent au problème d -HITTING SET, où étant donné une famille de sous-ensembles de cardinal au plus d , est cherché un ensemble minimum d'éléments qui intersecte tous ces sous-ensembles, pour lequel il existe un noyau de taille $O(k^{d-1})$ [1]. L'existence de noyaux polynomiaux a été montrée pour de nombreuses autres classes, nous pouvons citer sans être exhaustifs les forêts [126], les graphes cordaux [3], les graphes d'intervalles [4], les graphes blocs [2] et les graphes distances héréditaires [90].

Pour les problèmes de modification d'arêtes, ils ne se ramènent malheureusement pas au problème d -HITTING SET, en effet l'ajout ou la suppression d'une arête pour éliminer une obstruction peut en créer d'autres, comme illustré dans la figure 1.3.



Figure 1.3: Exemple d'un graphe où ajouter l'arête rouge pour éliminer un cycle de longueur 4 crée un nouveau cycle de longueur 4.

Néanmoins de nombreuses classes pouvant être caractérisées par des familles finies d'obstructions admettent des noyaux polynomiaux, nous pouvons citer les cographes [43, 75], les graphes cluster [32, 37, 52], les graphes split [10] et les graphes threshold [52]. L'existence de noyaux polynomiaux est connue pour peu d'autres classes, on y retrouve par exemple les graphes bipartis [94] et les forêts de chemins [63]. Cependant, la question reste ouverte pour des classes telles que les graphes planaires, les graphes d'intervalles ou encore pour les graphes cordaux pour lesquels un noyau polynomial n'est connu que pour l'ajout d'arêtes. Nous nous intéresserons particulièrement à l'existence de noyaux polynomiaux pour les sous-classes des graphes cordaux et autres classes proches, les résultats connus sont détaillés dans la table 3.1 du chapitre 3. Un aperçu plus général des résultats connus sur les algorithmes de noyaux pour les problèmes de modification d'arêtes est aussi proposé par Crespelle *et al.* [41].

Terminons en mentionnant qu'il existe des classes des graphes pour lesquelles les problèmes de modifications d'arêtes n'admettent pas de noyaux polynomiaux, sous certaines hypothèses de complexité. C'est notamment le cas pour des classes de graphes H -free, c'est-à-dire caractérisées par exactement une seule obstruction H [29, 75, 93, 110]. En particulier Cai et Cai [29] ont montré que c'était le cas quand H est un chemin ou un cycle de longueur au moins 4 ou un graphe 3-connexe non complet.

Contributions Dans cette thèse nous nous sommes particulièrement intéressés aux problèmes de modification d'arêtes vers des sous-classes des graphes cordaux. Les graphes cordaux sont les graphes sans trous, c'est-à-dire sans cycles induits de longueur 4 ou plus. Dans ce sens, ils peuvent donc être vus comme une généralisation des arbres. Notamment, les graphes cordaux peuvent être caractérisés comme les graphes d'intersection de sous-arbres d'un arbre. Cela implique que les graphes cordaux et leurs sous-classes ont des structures arborescentes, comme nous le verrons à plusieurs reprises dans ce document. De telles structures sont intéressantes pour mettre au point des noyaux polynomiaux pour les problèmes de modification d'arêtes. Nous pouvons aussi observer que si nous sommes restreints à l'ajout d'arêtes, alors les graphes cordaux sont en réalité caractérisés par une famille finie d'obstructions : un graphe contenant un cycle sans corde de

longueur plus de $k + 3$ ne peut pas avoir de solution. Par conséquent, le problème est FPT d'après le [théorème 1.1](#). En lien avec cette observation et les résultats connus, Bessy et Perez [14] ont formulé la conjecture suivante portant sur des sous-classes des graphes cordaux.

Conjecture. Soit une classe de graphes caractérisée par une famille d'obstructions contenant les trous plus un ensemble fini de graphes. Alors le problème d'ajout d'arêtes vers cette classe de graphes admet un noyau polynomial.

De nombreux résultats positifs allant dans le sens de cette conjecture ont été obtenus au fil des années, utilisant souvent des techniques similaires, mais qui dépendent toujours des spécificités de la classe étudiée. Ces résultats sont présentés plus en détail dans la [partie I](#) de ce document.

Notre objectif initial dans cette thèse était d'unifier les approches et techniques connues pour en mettre au point des génériques qui pourraient être appliquées à un plus grand nombre de classes de graphes. Pour aller dans ce sens nous avons étudié les problèmes de modification d'arêtes vers différentes sous-classes des graphes cordaux. Nous avons mis au point des noyaux polynomiaux pour plusieurs sous-classes des graphes cordaux. Pour tous ces noyaux nous utilisons une approche très similaire, qui se base sur l'exploitation d'une décomposition du graphe en une structure arborescente spécifique à chacune des classes. Néanmoins, à chaque fois nous avons besoin de propriétés structurelles fortes qui dépendent des classes de graphes considérées. En conséquence, nous n'avons pas réussi à définir de technique générique unifiant tous les résultats.

Nous présentons dans la première partie de cette thèse les noyaux que nous avons élaborés pour des problèmes de modification d'arêtes vers différentes classes de graphes. Nous appelons \mathcal{G} -EDITION le problème de modification d'arêtes vers la classe de graphes \mathcal{G} et où l'ajout et la suppression sont autorisés. Le problème \mathcal{G} -COMPLETION (resp. \mathcal{G} -DELETION) est défini de manière similaire en n'autorisant que l'ajout (resp. la suppression) d'arêtes. Nos résultats sont les suivants :

- TRIVIALY PERFECT EDITION ainsi que ses version complétion et délétion admettent un noyau qui contient $O(k^2)$ sommets,
- BLOCK GRAPH EDITION et BLOCK GRAPH DELETION admettent un noyau qui contient $O(k^2)$ sommets,
- STRICTLY CHORDAL EDITION admet un noyau qui contient $O(k^4)$ sommets,
- STRICTLY CHORDAL COMPLETION et STRICTLY CHORDAL DELETION admettent un noyau qui contient $O(k^3)$ sommets.

Notons qu'un noyau avec un nombre quadratique de sommets était déjà connu pour TRIVIALY PERFECT COMPLETION [10]. Pour les problèmes TRIVIALY PERFECT EDITION et TRIVIALY PERFECT DELETION, notre résultat améliore un précédent noyau qui contient $O(k^7)$ sommets [54]. Dans l'article présentant ce noyau, il était laissé comme question ouverte l'existence d'un noyau avec un nombre cubique de sommets (ou plus petit), nous répondons donc positivement à cette question. Nous montrons également que les techniques utilisées peuvent être adaptées de manière immédiate au problème THRESHOLD EDITION et ses versions complétion et délétion pour obtenir un noyau avec un nombre quadratique de sommets. Cependant un noyau avec un tel nombre de sommets était déjà connu pour ces problèmes [52].

Couverture par plus courts chemins

Parmi les problèmes de graphes les plus fondamentaux, nombreux sont des problèmes de couverture de graphes, nous y retrouvons notamment des problèmes comme VERTEX COVER, DOMINATING SET ou encore HAMILTONIAN PATH. Dans ce dernier problème est recherché un chemin qui passe exactement une fois par chacun des sommets du graphe. Il peut être généralisé au problème PATH COVER où est recherché un nombre minimum de chemins permettant de couvrir tous les sommets du graphe. En ajoutant à ce dernier problème la contrainte que les chemins soient disjoints les uns des autres, nous définissons le problème PATH PARTITION. Ce problème a particulièrement été étudié dans le contexte des graphes orientés, par exemple avec le théorème de Gallai-Milgram [71]

qui montre que la taille minimum d'une partition d'un graphe en chemins est bornée supérieurement par la taille de son ensemble indépendant maximum. Nous pouvons aussi mentionner la conjecture de Berge [12, 81], qui relie les partitions en chemins à une collection d'ensembles indépendants.

En retirant la contrainte que tous les sommets du graphe doivent être couverts, nous définissons le problème DISJOINT PATHS où sont données k paires de sommets, les *terminaux*, et est recherché un ensemble de k chemins disjoints les uns des autres, joignant chacun les terminaux d'une paire. Ce problème est un problème fondamental et précurseur dans le domaine de la complexité paramétrée en raison de l'algorithme FPT, paramétré par le nombre de paires de terminaux, proposé par Robertson et Seymour [118]. Ce résultat est fondamental pour la théorie des mineurs de graphes car il permet de montrer que le problème de tester si un graphe G admet un mineur H fixé est polynomial en la taille de G . Le graphe H est un mineur de G s'il peut être obtenu à partir de G en supprimant des sommets et des arêtes, et en contractant des arêtes.

Enfin, pour les différents problèmes que nous avons présentés, nous pouvons définir tout un ensemble de variantes en imposant des conditions sur les chemins, comme par exemple que les chemins soient de taille bornée [122], que ce soient des chemins induits ou encore que ce soient des plus courts chemins. Un aperçu général de différents problèmes de couverture ou de partition par différents types de chemins ainsi que les résultats connus sur ces derniers est proposé par Manuel [104]. Une étude des problèmes de partition par chemins sous l'angle de la complexité paramétrée est elle proposée par Fernau *et al.* [64].

Dans cette thèse, nous nous intéressons particulièrement à des problèmes de couverture de graphes par des *plus courts chemins*, parfois appelés chemins isométriques dans la littérature. La question : « Un graphe peut-il être couvert par au plus k plus courts chemins ? » peut être déclinée de différentes manières, selon que l'on souhaite couvrir toutes les arêtes ou seulement tous les sommets du graphe, selon que les extrémités des chemins sont données en entrée ou non, ou selon que les plus courts chemins sont disjoints ou non. Cela fait apparaître une famille de problèmes pour lesquels les réponses en termes d'algorithmique et de complexité paramétrée ne sont pas forcément identiques.

Le problème où est recherché un ensemble de k plus courts chemins couvrant tous les sommets du graphe est appelé ISOMETRIC PATH COVER (illustré dans la figure 1.4). Ce problème a été introduit par Fisher *et al.* [65] dans le contexte du jeu des gendarmes et du voleur [5]. Dans ce jeu, des gendarmes tentent d'attraper un voleur, chaque personnage est sur un sommet du graphe et à chaque tour peut se déplacer vers un sommet adjacent. Une couverture du graphe par des plus courts chemins permet de définir une stratégie gagnante pour attraper le voleur : chaque gendarme est assigné à un des plus courts chemins, et à chaque tour un gendarme se rapproche du voleur tout en restant sur le chemin qui lui est assigné. Cette stratégie permet de garantir l'arrestation du voleur et implique donc que le nombre de plus courts chemins nécessaires pour couvrir un graphe est une borne supérieure au nombre de gendarmes nécessaires pour attraper le voleur.

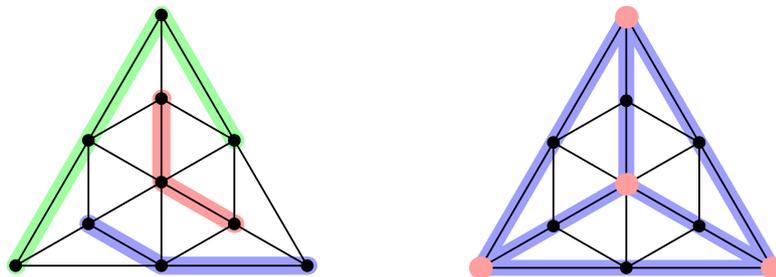


Figure 1.4: Exemple d'un graphe qui peut être couvert par 3 plus courts chemins (illustré à gauche), et qui a une solution de taille 4 pour STRONG GEODETIC SET (illustré à droite).

La version du problème où les plus courts chemins doivent relier des paires de sommets données en entrée est appelée ISOMETRIC PATH COVER WITH TERMINALS. Un problème proche, STRONG GEODETIC SET, a été récemment introduit par Manuel *et al.* [106, 107] (illustré dans la figure 1.4). Ce problème est une restriction du problème GEODETIC SET [80], dans lequel est recherché un ensemble de k terminaux tel que l'ensemble des plus courts chemins entre chaque paire de terminaux

couvre tous les sommets du graphe. La version « strong » de ce problème contraint à choisir un unique chemin pour chaque paire de terminaux. Nous pouvons, cette fois ci encore, définir une version où les terminaux sont donnés en entrée, à savoir STRONG GEODETIC SET WITH TERMINALS.

De manière surprenante, la complexité algorithmique de ces problèmes n'a été étudiée dans la littérature que très récemment. Ces problèmes ont tous été montrés NP-complets [34, 45, 107], mais ils peuvent être résolus en temps polynomial pour certaines classes de graphes [46, 112, 116]. Nous aborderons les résultats connus plus en détail dans l'introduction du [chapitre 6](#).

Contributions Nous montrons dans le [chapitre 6](#) que ISOMETRIC PATH COVER WITH TERMINALS et STRONG GEODETIC SET WITH TERMINALS admettent des algorithmes FPT paramétrés par le nombre de terminaux. De ces algorithmes FPT découlent des algorithmes XP pour ISOMETRIC PATH COVER et STRONG GEODETIC SET. En effet, il suffit d'appliquer pour tous les ensembles de k terminaux possibles l'algorithme FPT de la version avec terminaux. De plus, nous montrons que ces résultats sont aussi vérifiés dans les variantes de ces problèmes où nous cherchons à couvrir les arêtes du graphe ou bien que les plus courts chemins partitionnent le graphe. Nous répondons donc positivement à une question ouverte posée dans [100], demandant si STRONG GEODETIC SET WITH TERMINALS est FPT.

Pour obtenir ces résultats, nous montrons des propriétés structurelles sur les graphes dont les sommets ou les arêtes peuvent être couverts par un nombre fixé k de plus courts chemins. En particulier, nous montrons que dans un tel graphe G , étant donné un sommet arbitraire a et un entier D , le nombre de sommets à distance exactement D de a est borné par une fonction de k , et donc que la borne ne dépend pas de la taille du graphe d'entrée. Ce résultat implique que la largeur linéaire, et donc que la largeur arborescente de tels graphes est bornée par une fonction de k , cette borne pouvant être obtenue via une décomposition construite par un simple parcours en largeur. C'est cette borne sur la largeur arborescente qui nous permet d'utiliser une extension du théorème de Courcelle [8, 39] pour obtenir notre algorithme FPT.

Structure du document

Nous terminons ce chapitre en décrivant la structure de ce document. Dans le [chapitre 2](#), nous présentons les définitions et notions de graphes qui seront essentielles à cette thèse. Nous présentons également les classes et paramètres de graphes utiles pour la suite. Enfin nous formalisons les notions de complexité paramétrée évoquées dans l'introduction : les classes de complexité classiques telles FPT et XP, les algorithmes de noyaux qui seront au centre de la première partie de ce document, ainsi que le « méta-théorème » de Courcelle qui nous permettra d'obtenir les résultats algorithmiques de la seconde partie.

La [partie I](#) est consacrée à nos résultats d'algorithmes de noyaux pour différents problèmes de modification d'arêtes. Dans le [chapitre 3](#), nous commençons par présenter les noyaux polynomiaux connus pour les problèmes de modification d'arêtes vers des sous-classes des graphes cordaux. Ensuite, nous présentons des règles de réduction classiques pour les noyaux de problèmes de modification d'arêtes. Nous finissons par décrire l'approche par « décomposition et sommets affectés » que nous utilisons pour obtenir tous nos noyaux.

Dans le [chapitre 4](#) nous présentons un noyau avec un nombre quadratique de sommets pour le problème TRIVIALY PERFECT EDITION ainsi que ses variantes complétion et délétion. Un noyau avec un nombre quadratique était déjà connu pour le cas de la complétion [10], cependant pour l'édition et de la délétion, notre noyau améliore le meilleur noyau connu qui contenait $O(k^7)$ sommets [54]. Nous montrons ensuite que le noyau que nous avons défini peut être adapté de manière presque immédiate au problème THRESHOLD EDITION, obtenant ainsi un noyau avec un nombre quadratique de sommets pour ce problème. Cependant, ce noyau n'améliore que d'un facteur constant le meilleur résultat connu [52].

Dans le [chapitre 5](#) nous considérons les problèmes BLOCK GRAPH EDITION et sa version délétion, ainsi que STRICTLY CHORDAL EDITION et ses versions délétion et complétion. Nous commençons par montrer que tous ces problèmes sont NP-difficiles. Nous présentons ensuite un noyau avec un nombre quadratique de sommets pour le problème BLOCK GRAPH EDITION et sa version

délétion. En utilisant des idées similaires, nous montrons un noyau avec $O(k^4)$ sommets pour le problème STRICTLY CHORDAL EDITION. Ce problème porte sur les graphes strictement cordaux qui sont une généralisation des graphes blocs. Enfin, nous montrons que pour les versions de complétion et de délétion de ce problème, le noyau peut être aisément amélioré pour obtenir une borne cubique sur le nombre de sommets.

Nous concluons la [partie I](#) en présentant divers problèmes de modification d'arêtes vers des sous-classes de graphes cordaux pour lesquels l'existence de noyaux polynomiaux reste une question ouverte.

La [partie II](#), composée uniquement du [chapitre 6](#), est consacrée aux graphes dont les sommets ou les arêtes peuvent être couverts par k plus courts chemins. Nous montrons que dans les graphes couvrables par k plus courts chemins, le nombre de sommets à une distance donnée d'un sommet arbitraire est borné par une fonction exponentielle en k . Ce résultat combinatoire nous permet de conclure simplement que la largeur linéaire de ces graphes est également bornée par une fonction exponentielle en k . En utilisant une version étendue du théorème de Courcelle [8], nous montrons que les problèmes ISOMETRIC PATH COVER WITH TERMINALS et STRONG GEODETIC SET WITH TERMINALS sont FPT. De ces résultats découlent des algorithmes XP pour ISOMETRIC PATH COVER et STRONG GEODETIC SET.

Dans le [chapitre 7](#) nous concluons cette thèse et rappelons plusieurs pistes de recherche en lien avec nos travaux, en faisant une synthèse des questions ouvertes évoquées dans les [parties I et II](#).

Publications

La majorité des résultats présentés dans ce manuscrit sont publiés ou acceptés pour publication dans des colloques internationaux, et ont été également soumis ou acceptés à des journaux. Ci-après la liste des publications dans lesquelles j'ai été impliqué au cours de ma thèse.

Conférences Internationales :

- [59] **A Cubic Vertex-Kernel for Trivially Perfect Editing**
Avec Anthony Perez et Ioan Todinca. MFCS 2021.
- [60] **Polynomial kernels for strictly chordal edge modification problems**
Avec Anthony Perez et Ioan Todinca. IPEC 2021.
- [55] **On graphs coverable by k shortest paths**
Avec Florent Foucaud, Anthony Perez et Ioan Todinca. ISAAC 2022.
- [57] **An improved kernelization algorithm for Trivially Perfect Editing**
Avec Anthony Perez. IPEC 2023.

Journaux :

- [61] **A cubic vertex-kernel for Trivially Perfect Editing**
Avec Anthony Perez et Ioan Todinca. Algorithmica. 2023.
Version complète de [59].
- **On graphs coverable by k shortest paths**
Avec Florent Foucaud, Anthony Perez et Ioan Todinca. SIAM Journal on Discrete Mathematics (accepté). 2023.
C'est une version améliorée de [55], disponible sur [arXiv](#) [56]. Cette version améliore la borne supérieure sur la largeur linéaire (pathwidth) des graphes qui peuvent être couverts par k plus courts chemins. L'ancienne borne était factorielle en k , la nouvelle simplement exponentielle en k .

Soumissions en journal :

- **Polynomial kernels for edge modification problems towards block and strictly chordal graphs**

Avec Anthony Perez, Mathis Rocton et Ioan Todinca.

C'est une version complète et étendue de [60], disponible sur [arXiv](#) [58]. Cette version ajoute un noyau polynomial pour les problèmes BLOCK GRAPH EDITION et BLOCK GRAPH DELETION.

Chapitre 2

Préliminaires

2.1 Graphes

Dans cette section nous donnons les principales définitions et notations utilisées pour les graphes dans ce document. Nous présentons également des classes et paramètres de graphes, ainsi que les problèmes de modification de graphes essentiels au reste du document. Pour une introduction plus complète des notions classiques de théorie des graphes, nous renvoyons le lecteur au livre de Diestel [48].

2.1.1 Définitions générales

Un **graphe non-orienté** est une paire $G = (V, E)$ d'ensembles finis où $E \subseteq [V]^2$ est un ensemble de paires d'éléments¹ de V . L'ensemble V est l'ensemble des **sommets** et E l'ensemble des **arêtes**. Lorsque le contexte n'est pas clair, nous utilisons respectivement $V(G)$ et $E(G)$ pour désigner l'ensemble des sommets et des arêtes de G . Deux sommets u et v sont dit **adjacents** ou **voisins** si $\{u, v\} \in E$. Les sommets u et v d'une arête $e = \{u, v\}$ sont appelés les **extrémités** de e , cette arête est dite **incidente** à u et à v .

Le **voisinage (ouvert)** d'un sommet $u \in V$ est l'ensemble $N_G(u) = \{v \in V \mid \{u, v\} \in E\}$. Le **voisinage fermé** de u est défini comme $N_G[u] = N_G(u) \cup \{u\}$. Pour un sous-ensemble de sommets $S \subseteq V$, $N_G[S]$ est l'ensemble $\bigcup_{v \in S} N_G[v]$ et $N_G(S)$ est l'ensemble $N_G[S] \setminus S$. Nous omettons la mention de G quand le contexte est clair. Le **degré** d'un sommet u est son nombre de voisins.

Le **complémentaire** de G est le graphe $\overline{G} = (V, \overline{E})$ tel que $\overline{E} = \{\{u, v\} \mid u, v \in V \wedge u \neq v \wedge \{u, v\} \notin E\}$. Une paire de sommets $\{u, v\}$ est une **non-arête** de G si $\{u, v\} \notin E$. Un **sous-graphe** $H = (V_H, E_H)$ de G est un graphe tel que $V_H \subseteq V$ et $E_H \subseteq \{\{u, v\} \in E \mid u, v \in V_H\}$. Le sous-graphe **induit** par $S \subseteq V$ est défini par $G[S] = (S, E_S)$ où $E_S = \{\{u, v\} \in E \mid u, v \in S\}$. Pour faciliter la lisibilité, nous notons $G \setminus S$ le graphe $G[V \setminus S]$ obtenu à partir de G en supprimant les sommets de S . Soit H un sous-graphe induit de G , nous disons que G **contient** H .

Un **chemin** P d'un graphe G est une suite de sommets (u_1, \dots, u_l) de G telle que $\{u_i, u_{i+1}\}$, $1 \leq i \leq l-1$, est une arête de G . Un **cycle** C d'un graphe G est une suite de sommets $(u_1, \dots, u_{l-1}, u_l)$ de G telle que $\{u_i, u_{i+1}\}$, $1 \leq i \leq l-1$, et $\{u_1, u_l\}$ sont des arêtes de G . La **longueur** d'un chemin ou d'un cycle X correspond au nombre de ses arêtes, elle est notée $|X|$. Un **plus court** chemin est un chemin de longueur minimum entre deux sommets. La **distance** entre deux sommets u et v dans G est notée $\text{dist}(u, v)$ et correspond à la longueur d'un plus court chemin entre u et v . Une **corde** d'un chemin (d'un cycle) est une arête entre deux sommets non consécutifs du chemin (du cycle). Un chemin sans corde sur n sommets est noté P_n , un cycle sans corde sur n sommets est noté C_n (observons que $|P_n| = n - 1$ et $|C_n| = n$). Un cycle sans corde de longueur au moins 4 est appelé un **trou**.

Un graphe G est **connexe** si pour toute paire de sommets u, v de G , il existe un chemin connectant u et v . Un sous-graphe C est une **composante connexe** de G s'il est un sous-graphe connexe maximal de G . Un graphe est **biconnexe** si le graphe reste connexe après la suppression

¹Ici $[X]^2$ désigne l'ensemble des paires d'un ensemble X . Nous utilisons $X \times Y$ pour désigner l'ensemble des paires (non ordonnées) ayant un élément dans X et l'autre dans Y quand les deux ensembles sont disjoints.

d'un de ses sommets. Un ensemble $S \subseteq V$ est un **séparateur** de G si $G \setminus S$ n'est pas connexe. Étant donné deux sommets u et v de G , S est un uv -séparateur si u et v se trouvent dans des composantes connexes distinctes de $G \setminus S$. De plus, S est un uv -séparateur *minimal* si aucun sous-ensemble propre de S n'est un uv -séparateur. Enfin, un **séparateur minimal** est un séparateur S tel qu'il existe une paire $\{u, v\}$ telle que S est un uv -séparateur minimal. Cette notion est illustrée dans la figure 2.1.

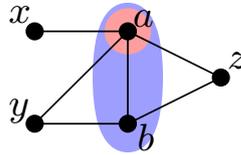


Figure 2.1: Les séparateurs minimaux de ce graphe sont $\{a, b\}$ et $\{a\}$, en particulier $\{a, b\}$ est un yz -séparateur minimal, $\{a\}$ est un xy -séparateur minimal.

Un sous-ensemble de sommets $K \subseteq V$ est une **clique** si $\{u, v\} \in E$ pour tout $u, v \in K$. Un graphe dont les n sommets forment une clique est noté K_n . Un sous-ensemble de sommets $I \subseteq V$ est appelé **indépendant** ou **stable** si $\{u, v\} \notin E$ pour tout $u, v \in I$. Un graphe dont les n sommets forment un indépendant est noté I_n . Un sous-ensemble de sommets $M \subseteq V$ est un **module** de G si $N_G(u) \setminus M = N_G(v) \setminus M$ pour tout $u, v \in M$.

Deux sommets u et v sont **jumeaux** si $N_G(u) \setminus \{u\} = N_G(v) \setminus \{v\}$. Ce sont des **vrais jumeaux** si u et v sont adjacents, des **faux jumeaux** sinon. Un ensemble maximal de vrais jumeaux $K \subseteq V$ est une **clique critique**. Remarquons que K est un module clique et que l'ensemble des cliques critiques d'un graphe partitionne son ensemble de sommets. Un sommet $u \in V$ est **universel** si $N_G[u] = V$. L'ensemble des sommets universels forme une clique, appelée la **clique universelle** de G . Remarquons que la clique universelle est une clique critique.

Une **couverture par sommets** de G est un ensemble de sommets $S \subseteq V$ tel que pour tout $\{u, v\} \in E$, $u \in S$ ou $v \in S$. Un **ensemble dominant** de G est un ensemble de sommets $D \subseteq V$ tel que pour tout $v \in V$, il existe $u \in D$ tel que $\{u, v\} \in E$.

Le **graphe d'intersection** d'une famille $\{S_1, \dots, S_n\}$ de sous-ensembles d'un ensemble S est le graphe $G = (V, E)$ tel que $V = \{v_1, \dots, v_n\}$ et $E = \{\{v_i, v_j\} \mid i \neq j, S_i \cap S_j \neq \emptyset\}$. Une famille \mathcal{X} d'ensembles est **imbriquée** si pour tout $A, B \in \mathcal{X}$, $A \subseteq B$ ou $B \subseteq A$. Une famille \mathcal{X} d'ensembles est **laminaire** si pour tout $A, B \in \mathcal{X}$, $A \subseteq B$ ou $B \subseteq A$ ou $A \cap B = \emptyset$. Ces familles sont illustrées dans la figure 2.2.

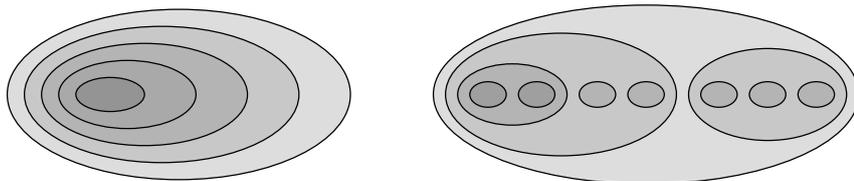


Figure 2.2: Illustration à gauche d'une famille imbriquée; à droite d'une famille laminaire.

2.1.2 Classes de graphes

Une classe de graphes \mathcal{G} est **héréditaire** si pour un graphe $G \in \mathcal{G}$, tout sous-graphe induit de G appartient à \mathcal{G} . Une classe de graphes \mathcal{G} est **close** pour une certaine opération si tout graphe obtenu à partir de graphes de \mathcal{G} par cette opération est dans \mathcal{G} . Par exemple, cette opération peut être l'union disjointe de graphes ou encore l'ajout de jumeaux d'un sommet. Soit \mathcal{H} une famille de graphes, éventuellement infinie, un graphe G est dit **\mathcal{H} -free** s'il ne contient aucun graphe de \mathcal{H} comme sous-graphe induit. Une classe de graphes \mathcal{G} est caractérisée par une *famille de sous-graphes induits interdits* s'il existe une famille de graphes \mathcal{H} telle que \mathcal{G} est formée précisément des

graphes \mathcal{H} -free. Les sous-graphes interdits de la famille \mathcal{H} sont appelés les **obstructions** de \mathcal{G} . La classe **complémentaire** d'une classe de graphes \mathcal{G} , appelée $\text{co-}\mathcal{G}$, est la classe obtenue en prenant le complémentaire de tous les graphes de \mathcal{G} , c'est-à-dire $\{\bar{G} \mid G \in \mathcal{G}\}$.

Nous présentons dans ce qui suit une liste des classes de graphes qui sont étudiées ou mentionnées dans cette thèse. La plupart de ces classes sont des sous-classes des graphes cordaux. Nous commençons par introduire des classes classiques, puis celles que nous étudions dans cette thèse et enfin des classes proches de ces dernières. La figure 2.4 contient le diagramme des inclusions entre les classes présentées. Notons que toutes ces classes (à l'exception des arbres) sont héréditaires et peuvent être caractérisées par une famille de sous-graphes induits interdits. Un plus grand aperçu des classes de graphes, leurs propriétés et leurs relations d'inclusion peut être retrouvé dans le livre de Brandstädt, Le et Spinrad [24].

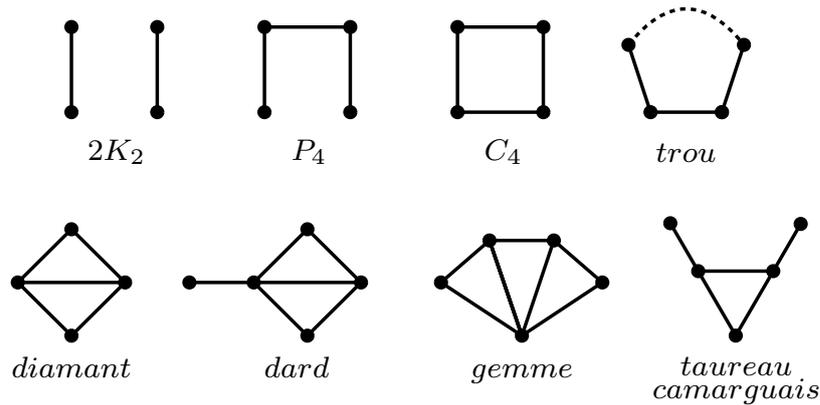


Figure 2.3: Illustration des petits graphes utilisés pour caractériser les classes de graphes présentées dans cette section.

Arbre et forêt Un *arbre* est un graphe connexe sans cycle. Un sommet d'un arbre est appelé un *noeud*, s'il est de degré 1 c'est une *feuille* de l'arbre, sinon c'est un *noeud interne*. Un arbre *enraciné* est un arbre dont un sommet est étiqueté comme la racine. La *hauteur* d'un arbre enraciné est la longueur du plus long chemin de sa racine à une de ses feuilles. Une *forêt* est un graphe dont chacune des composantes connexes est un arbre.

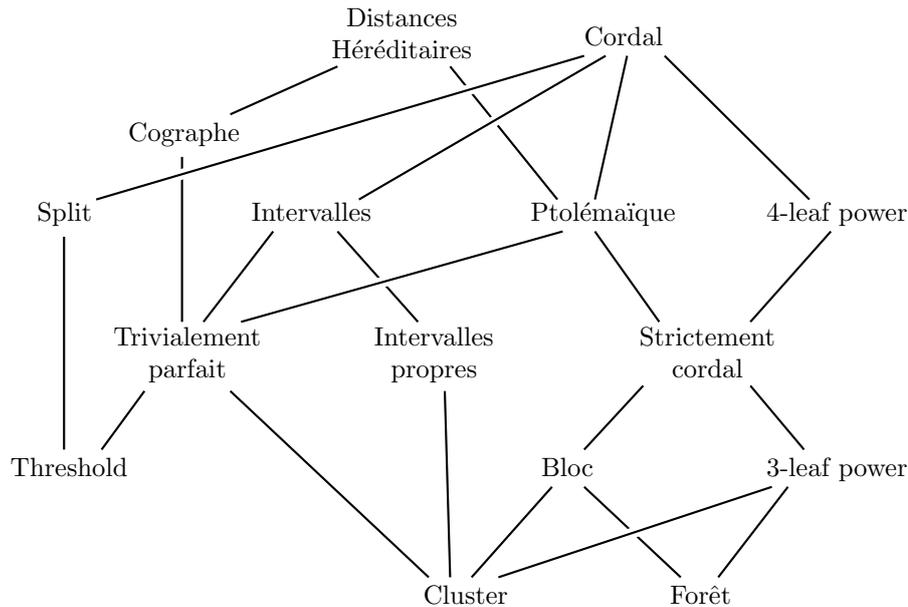
Cordal Un graphe est *cordal* ou *triangulé* s'il ne contient pas de trou, c'est-à-dire que chacun de ses cycles de longueur au moins 4 contient une corde. Les graphes cordaux peuvent aussi être caractérisés comme les graphes dont tous les séparateurs minimaux sont des cliques ou comme les graphes d'intersections de sous-arbres d'un arbre. Toutes les classes que nous étudierons dans la première partie de ce document sont des sous-classes des graphes cordaux.

Intervalles Un graphe est d'*intervalles* s'il est le graphe d'intersection d'intervalles de la droite réelle. Les graphes d'intervalles peuvent aussi être caractérisés comme les graphes cordaux sans *triplets astéroïdaux*, qui sont des ensembles de trois sommets pour lesquels, pour chaque paire de sommets, il existe un chemin les reliant qui ne passe pas par le voisinage du troisième sommet.

Intervalles propres Un graphe est d'*intervalles propres* ou d'*intervalles unitaires* s'il est le graphe d'intersection d'intervalles unitaires de la droite réelle.

Cluster Un graphe est *cluster* s'il est $\{P_3\}$ -free. Ce sont les graphes formés par une union disjointe de cliques.

Split Un graphe est *split* s'il est $\{2K_2, C_4, C_5\}$ -free. Ce sont les graphes qui peuvent être partitionnés en une clique et un indépendant.

Figure 2.4: Diagramme d'inclusion des classes présentées dans la [section 2.1.2](#).

Threshold Un graphe est *threshold* s'il est $\{P_4, C_4, 2K_2\}$ -free. Les graphes threshold peuvent aussi être caractérisés comme les graphes split dont les voisinages des sommets de chacune des parties forment une famille imbriquée.

Trivialement parfait Un graphe est *trivialement parfait* s'il est $\{P_4, C_4\}$ -free. Ces graphes sont également connus sous le nom *quasi-threshold*. Les graphes trivialement parfaits peuvent aussi être caractérisés comme les graphes tels que chacun de leurs sous-graphes induits connexes possède un sommet universel, ou encore comme les graphes d'intervalles où les intervalles forment une famille laminaire.

Chaîne Un graphe est *chaîne* s'il est $\{K_3, 2K_2, C_5\}$ -free. De manière similaire aux graphes threshold, ils peuvent être partitionnés en deux indépendants tels que les voisinages des sommets de chacun des indépendants forment une famille imbriquée. Les graphes *co-chaînes* sont les graphes $\{I_3, C_4, C_5\}$ -free, contrairement aux graphes chaînes, ils sont cordaux.

Bloc Un graphe est *bloc* s'il est $\{trou, diamant\}$ -free. Les graphes bloc peuvent aussi être caractérisés comme les graphes dont les séparateurs minimaux sont de taille 1 et les graphes dont les composantes biconnexes sont des cliques.

Strictement cordal Un graphe est *strictement cordal* s'il est $\{trou, gemme, dard\}$ -free. Ces graphes sont également connus sous le nom *block duplicate*. Les graphes strictement cordaux peuvent aussi être caractérisés comme les graphes cordaux dont les séparateurs minimaux sont deux à deux disjoints, en particulier ce sont des cliques critiques.

Cographe Un graphe est un *cographe* s'il est $\{P_4\}$ -free. Ce sont les graphes où le complémentaire d'un sous-graphe induit connexe est non-connexe. Ce sont aussi les graphes où chaque sous-graphe induit contient une paire de jumeaux.

Distances héréditaires Un graphe est *distances héréditaires* si les distances entre ses sommets sont identiques dans tous ses sous-graphes connexes induits.

Ptolémaïque Un graphe est ptolémaïque s'il est $\{trou, gemme\}$ -free. Ce sont les graphes cordaux et distances héréditaires, et les graphes cordaux dont les séparateurs minimaux forment une famille laminaire.

r-leaf power Un graphe $G = (V, E)$ est r -leaf power s'il existe un arbre $T = (V_T, E_T)$ tel que les feuilles de T sont les sommets de G , et $\{u, v\} \in E$ si et seulement si u et v sont à distance au plus r dans T . Notons que les 1-leaf power correspondent aux graphes sans arêtes, ou au graphe formé de deux sommets adjacents. Les 2-leaf power correspondent aux graphes cluster. Les 3-leaf power sont les graphes $\{trou, gemme, dard, taureau\}$ -free.

Notons que la plupart des inclusions et non-inclusions présentées dans la [figure 2.4](#) sont immédiates par la définition des classes, en particulier en terme de sous-graphes interdits. Pour la classes des graphes 4-leaf power, il existe une caractérisation en sous-graphes induits interdits [117] qui permet de conclure sur les non-inclusions avec toutes les classes sur la gauche du diagramme. En particulier, les graphes suivants sont des obstructions des 4-leaf power : un dard auquel on a ajouté un faux jumeau aux sommets de degré 2; deux gemmes que l'on a fusionnées sur un sommet de degré 2. Le premier graphe est threshold, le second est d'intervalles propres.

2.1.3 Modification d'arêtes

Les problèmes étudiés dans la première partie de ce document sont des problèmes de modification d'arêtes. Formellement, étant donné une classe de graphes \mathcal{G} nous définissons le problème suivant :

\mathcal{G} -EDITION

Entrée : Un graphe $G = (V, E)$, un paramètre $k \in \mathbb{N}$.

Question : Existe-t-il un ensemble F de paires d'éléments de V de taille au plus k tel que le graphe $H = (V, E \Delta F)$ appartient à la classe \mathcal{G} , où $E \Delta F = (E \setminus F) \cup (F \setminus E)$?

Le problème \mathcal{G} -COMPLETION (resp. \mathcal{G} -DELETION) est défini de manière similaire en n'autorisant que l'ajout (resp. la suppression) d'arêtes, c'est-à-dire $F \cap E = \emptyset$ (resp. $F \subseteq E$). Étant donné une instance $(G = (V, E), k)$ de \mathcal{G} -EDITION, un ensemble de paires $F \subseteq [V]^2$ tel que $H = (V, E \Delta F)$ est dans la classe \mathcal{G} est une **édition** de G . Quand F est contraint à être disjoint de E (resp. contenu dans E), l'ensemble F est une **complétion** (resp. une **délétion**) de G . Pour plus de simplicité les graphes $(V, E \Delta F)$, $(V, E \cup F)$ et $(V, E \setminus F)$ sont notés respectivement $G \Delta F$, $G + F$ et $G - F$. Un sommet est **affecté** par F s'il est contenu dans une des paires appartenant à F . L'ensemble F est une k -édition (resp. k -complétion, k -délétion) quand $|F| \leq k$. Un tel ensemble F est **optimal** quand il est de taille minimum.

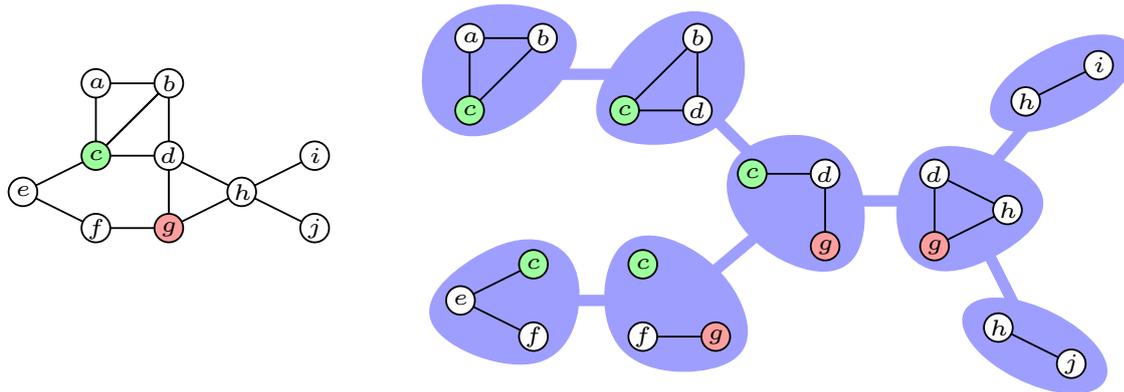
2.1.4 Paramètres de graphes

Dans cette sous-section nous présentons des paramètres de graphes classiques. Ces paramètres permettent de mieux comprendre la structure des graphes et sont souvent des outils essentiels pour la mise au point d'algorithmes (paramétrés) efficaces.

Une **décomposition arborescente** de $G = (V, E)$ est un couple (T, \mathcal{X}) où $T = (V_T, E_T)$ est un arbre et \mathcal{X} est une famille $\{X_t \subseteq V \mid t \in V_T\}$ de sous-ensembles de V , appelés *sacs*, telle que :

1. L'union de tous les sacs de \mathcal{X} est égale à V , soit $\bigcup_{t \in V_T} X_t = V$,
2. Pour toute arête $\{u, v\} \in E$, il existe $t \in V_T$ tel que $u, v \in X_t$,
3. Pour tout sommet $u \in V$, les noeuds de T dont les sacs contiennent u forment un sous-arbre connexe.

La largeur de (T, \mathcal{X}) est $\max\{|X_t| - 1 \mid t \in V_T\}$, et la **largeur arborescente** (treewidth) de G , notée $\text{tw}(G)$, est la largeur minimum parmi toutes les décompositions arborescentes de G .

Figure 2.5: Un graphe G et une décomposition arborescente de G de largeur 2.

Les notions de **décomposition linéaire** et **largeur linéaire** (pathwidth) sont définies de manière analogue en ajoutant la condition que l'arbre T doit être un chemin. La largeur linéaire de G est notée $\text{pw}(G)$.

Une **forêt d'élimination** de $G = (V, E)$ est une forêt enracinée $T = (V_T, E_T)$ telle qu'il existe une bijection $f: V \rightarrow V_T$ respectant la propriété : pour toute arête $\{u, v\} \in E$, $f(u)$ est un descendant de $f(v)$ dans T ou inversement. La **profondeur d'arbre** (treedepth), notée $\text{td}(G)$, est la hauteur minimale d'une forêt d'élimination de G .

Ces trois paramètres sont reliés par l'inégalité : $\text{tw}(G) \leq \text{pw}(G) \leq \text{td}(G) - 1$ [19]. La première partie est immédiate vu que largeur linéaire est la restriction de la largeur arborescente aux chemins. Pour la seconde partie, à partir d'une forêt d'élimination F de hauteur h pour un graphe G , il est aisé de construire une décomposition linéaire de largeur h . Soit $\{v_1, \dots, v_l\}$ les feuilles de F , ordonnées selon un parcours en profondeur de la forêt F . Pour chaque feuille v_i , nous créons un sac X_i qui contient tous les ancêtres de cette feuille. Alors, le chemin sur l sommets auquel on associe les sacs X_1, \dots, X_l est une décomposition linéaire de largeur h . En effet, par définition dans F les deux extrémités d'une arête du graphe G sont associées à des noeuds tels que l'un est ancêtre de l'autre. Ainsi, il existe un sac qui contient ses deux extrémités. De plus, par l'ordre sur les feuilles, il est immédiat que les sacs qui contiennent un sommet forment un sous-chemin connexe. Enfin, par construction la taille maximale d'un sac correspond à la hauteur h de la forêt.

Les valeurs de ces trois paramètres pour certaines classes de graphes simples sont immédiates. Les cliques sont de largeur arborescente, largeur linéaire et profondeur d'arbre $n - 1$. Les arbres sont de largeur arborescente 1, de largeur linéaire et profondeur d'arbre $O(\log n)$. Les chemins sont de largeur linéaire 1 et de profondeur d'arbre $O(\log n)$. Les étoiles (arbres de hauteur 1) sont de profondeur d'arbre 1.

Nous concluons cette section en observant qu'il existe un lien fort entre les paramètres de graphes présentés et les problèmes de modification de graphes. Définissons d'abord $\omega(G)$ comme étant la taille maximum d'une clique d'un graphe G . Rappelons qu'un graphe cordal G peut être caractérisé comme le graphe d'intersection de sous-arbres $\{T_1, T_2, \dots, T_n\}$ d'un arbre $T = (V_T, E_T)$. Définissons l'ensemble $\mathcal{X} = \{X_t \mid t \in V_T\}$ où le sac X_t contient tous les sommets v_i de G associés aux sous-arbres T_i qui contiennent le noeud t . Le couple (T, \mathcal{X}) est par construction une décomposition arborescente du graphe G . Une observation immédiate est que la largeur de cette décomposition est égale à la taille de la clique maximum de G moins 1. Ainsi, la largeur arborescente d'un graphe cordal G est égale à $\omega(G) - 1$ puisque tous les sommets d'une clique sont contenus dans un même sac pour toute décomposition arborescente. Une conséquence est que pour tout graphe G , sa largeur arborescente est égale à la plus petite valeur prise par $\omega(H) - 1$ parmi toutes les complétions en graphe cordal H de G . Pour observer cela, il suffit de prendre une décomposition arborescente de largeur minimum de G , et de compléter chaque sac de cette décomposition en une clique, il en résulte que le graphe H obtenu est cordal et que $\omega(H) - 1 = \text{tw}(H) = \text{tw}(G)$. Par exemple, dans la [figure 2.5](#), le graphe contient un cycle sans corde (c, d, g, f, e) , compléter chaque sac en clique revient à ajouter les arêtes $\{c, g\}$ et $\{c, f\}$, ce qui rend le graphe cordal.

Une relation similaire peut être observée pour les deux autres paramètres, en effet, la largeur linéaire (resp. la profondeur d'arbre) est égale à la plus petite valeur prise par $\omega(H) - 1$ parmi toutes les complétions en graphe d'intervalles (resp. graphe trivialement parfait) H de G .

2.2 Complexité paramétrée

Dans cette section nous définissons formellement des notions centrales de complexité paramétrée qui sont utiles tout au long de ce document. Nous commençons par présenter les problèmes fixed-parameter tractable et les classes de complexité paramétrée, ensuite nous présentons les algorithmes de noyau et nous finissons par la présentation du théorème de Courcelle qui permet d'obtenir des algorithmes fixed-parameter tractable pour des graphes de largeur arborescente bornée. Pour plus de détails, nous renvoyons le lecteur au livre de Downey et Fellows [51] pour des notions générales de complexité paramétrée, et au livre de Fomin et al. [68] pour un aperçu plus général sur les algorithmes de noyau.

Définition 2.1. Un *problème paramétré* est un langage $L \subseteq \Sigma^* \times \mathbb{N}$ où Σ est un alphabet fini. Pour une instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k est appelé le *paramètre*.

Tout au long de cette section, nous utiliserons le problème VERTEX COVER comme exemple pour illustrer les différentes notions introduites. Ce problème correspond au langage de l'ensemble des graphes qui admettent une couverture de sommets de taille au plus k . Nous le définissons comme suit :

VERTEX COVER

Entrée : Un graphe $G = (V, E)$, un paramètre $k \in \mathbb{N}$.

Question : Existe-t-il une couverture par sommets S de G de taille au plus k ?

2.2.1 Classes de complexité paramétrée

Définition 2.2. Un problème paramétré est *fixed-parameter tractable* s'il existe un algorithme qui décide si $(x, k) \in L$ en temps $f(k) \cdot |x|^{O(1)}$, où f est une fonction calculable. La classe de complexité contenant tous les problèmes fixed-parameter tractable est appelée FPT.

Une classe de complexité paramétrée plus grande est la classe XP qui contient tous les problèmes qui peuvent être décidés en temps $f(k) \cdot |x|^{g(k)}$, où f et g sont des fonctions calculables. Cette classe correspond aux problèmes dits *polynomial par tranche*, c'est-à-dire les problèmes pour lesquels il existe un algorithme polynomial pour les résoudre pour chaque valeur de k , mais dont le polynôme peut être différent pour chaque valeur de k .

Nous présentons à présent un exemple d'algorithme FPT pour VERTEX COVER. Par définition, toute arête d'un graphe G a au moins une de ses extrémités dans une couverture par sommets de G . Cette propriété permet de définir un algorithme FPT simple pour ce problème. Soit l'algorithme $VC(G, k)$, qui renvoie vrai si le graphe G a une couverture par sommets de taille au plus k , faux sinon. Nous pouvons le définir comme suit :

Algorithme 1 : $VC(G, k)$

```

si  $G$  ne contient aucune arête alors
   $\perp$  renvoyer « vrai »
si  $k = 0$  et  $G$  contient une arête alors
   $\perp$  renvoyer « faux »
  Choisir  $\{u, v\} \in E(G)$ 
  renvoyer  $VC(G \setminus \{u\}, k - 1) \vee VC(G \setminus \{v\}, k - 1)$ 

```

Analysons maintenant le temps d'exécution de cet algorithme. Soit $T(n, k)$ le temps d'exécution de $VC(G, k)$, si $k = 0$, nous avons $T(n, 0) = poly(n)$, sinon

$$T(n, k) \leq 2 \cdot T(n - 1, k - 1) + poly(n).$$

Vu que $k \leq n$ (autrement la réponse est triviale), nous avons : $T(n, k) \leq 2^k \cdot \text{poly}(n)$. Ainsi cet algorithme est bien FPT. Nous présenterons des outils pour analyser le temps d'exécution d'algorithmes de branchement de manière plus détaillée dans la [section 2.3](#).

Certains problèmes paramétrés sont conjecturés comme n'étant pas FPT, et plus encore, il semblerait qu'il y ait différents niveaux de difficultés de complexité paramétrée. Par exemple les problèmes INDEPENDENT SET ou DOMINATING SET sont conjecturés comme n'étant pas FPT, de plus le second serait plus difficile que le premier. Pour comparer la difficulté de problèmes paramétrés, la notion classique de réduction d'un problème (non paramétré) à un autre peut être adaptée. Soit L et L' deux problèmes paramétrés, une **réduction paramétrée** de L à L' est un algorithme prenant $(x, k) \in L$ en entrée et qui renvoie une instance (x', k') de L' tel que :

1. $(x, k) \in L$ si et seulement si $(x', k') \in L'$,
2. $k' \leq g(k)$ pour une fonction calculable g ,
3. son temps d'exécution est $f(k) \cdot |x|^{O(1)}$ où f est une fonction calculable.

Pour classifier les problèmes paramétrés, Downey et Fellows [51] ont introduit la notion de **W-hiérarchie**. Informellement, $W[0]$ correspond à la classe FPT, $W[1]$ correspond aux problèmes qui peuvent se réduire depuis INDEPENDENT SET paramétré par la taille de la solution, $W[2]$ correspond aux problèmes qui peuvent se réduire depuis DOMINATING SET paramétré par la taille de la solution.

Plus formellement, un problème paramétré L est dans $W[t]$ si toute instance (x, k) peut être réduite en temps FPT à un circuit booléen C de profondeur constante (indépendante de l'entrée), de « weft » au plus t et tel que $(x, k) \in L$ si et seulement s'il existe une assignation satisfaisante assignant 1 à exactement k entrées. La « weft » d'un circuit est le nombre maximum de portes ayant plus de deux arcs entrants, situées sur un même chemin d'une entrée vers la sortie. Nous avons :

$$FPT \subseteq W[1] \subseteq W[2] \subseteq W[3] \subseteq \dots \subseteq XP$$

Ces inclusions sont conjecturées comme étant strictes. Ainsi, qu'un problème soit $W[1]$ -difficile est un fort indicateur qu'il ne soit pas FPT.

Nous terminons en rappelant qu'un problème non paramétré a autant de versions paramétrées qu'il existe de paramètres différents. Il existe de nombreux paramètres, comme par exemple la taille de la solution. C'est notamment le paramètre qui est majoritairement utilisé pour des problèmes de modification de graphes. D'autres paramètres liés à la structure du graphe peuvent aussi être considérés comme par exemple le degré maximum, la largeur arborescente, la largeur linéaire, la profondeur d'arbre, la taille de la couverture par sommets minimum ou bien la somme de plusieurs paramètres. Le théorème de Courcelle, présenté dans la [section 2.2.3](#) montre qu'une grande partie de problèmes classiques des graphes sont FPT quand ils sont paramétrés par la largeur arborescente du graphe en entrée. Ce théorème permet de montrer en particulier que INDEPENDENT SET et DOMINATING SET sont FPT quand ils sont paramétrés par la largeur arborescente. Cela nous permet de remarquer que l'appartenance d'un problème à une classe de complexité paramétrée dépend de son paramétrage.

2.2.2 Algorithme de noyau

Définition 2.3. Un *algorithme de noyau*, ou plus simplement un *noyau*, pour un problème paramétré L est un algorithme qui prend en entrée une instance (x, k) et retourne une instance (x', k') en temps polynomial en $|x| + k$ tel que :

$$((x, k) \in L \iff (x', k') \in L) \text{ et } |x'|, k' \leq h(k),$$

où h est une fonction calculable. La fonction h est appelée la *taille* du noyau, si h est polynomiale alors le noyau est dit polynomial.

Un noyau est usuellement présenté sous la forme d'un ensemble de *règles de réduction*, permettant de réduire l'instance en entrée à une instance équivalente. Informellement, elles permettent la simplification de la structure de l'entrée. Cette réduction est le plus souvent liée à une diminution de la taille de l'entrée et du paramètre, mais il arrive dans certains cas qu'une règle augmente leur taille si cela permet d'obtenir des propriétés intéressantes sur la nouvelle instance.

Définition 2.4 (sûreté). Une règle de réduction prenant en entrée une instance (x, k) et produisant en sortie une instance réduite (x', k') est *sûre* pour un problème paramétré L si :

$$(x, k) \in L \Leftrightarrow (x', k') \in L.$$

Un ensemble de règles est dit être *appliqué exhaustivement* sur une instance si aucune des règles de cet ensemble ne peut être appliquée sur l'instance réduite. Algorithmiquement, cela revient généralement à appliquer une règle tant qu'elle est applicable. Cependant, les règles d'un noyau doivent parfois être appliquées dans un certain ordre pour garantir que le temps d'exécution de l'algorithme soit polynomial.

La taille $h(k)$ d'un noyau est définie par la taille maximum d'une instance positive réduite exhaustivement par l'ensemble des règles. La preuve de la taille d'un noyau consiste donc à montrer cette taille maximum. Cela nous permet de conclure que si la taille d'une instance réduite est plus grande que la taille $h(k)$ du noyau, alors cette instance est nécessairement négative. Nous pouvons donc définir la règle qui suit.

Règle. Soit (x, k) une instance d'un problème paramétré L réduite exhaustivement par un ensemble de règles. Si $|x| > h(k)$, alors renvoyer la plus petite instance $(y, 0) \notin L$.

Cette règle nous permet de borner la taille de *toutes* les instances, même négatives. La preuve de la taille d'un noyau correspond dans les faits à montrer la sûreté de cette règle. Étant implicite, nous ne mentionnons pas cette règle pour les algorithmes de noyaux que nous présentons.

La notion de noyau peut être vue comme une définition alternative des algorithmes FPT comme le montre le théorème qui suit.

Théorème 2.1 ([30]). *Un problème paramétré admet un algorithme FPT si et seulement s'il admet un noyau.*

Preuve. Soit L un problème paramétré, supposons qu'il admet un noyau s'exécutant en temps polynomial $(|x| + k)^c$ pour une instance (x, k) et une constante c . Soit $f(k)$ la taille de l'instance (x', k') réduite par le noyau. Appliquer un algorithme de décision pour décider si $(x', k') \in L$ prend un temps $g(|x'|) \leq g(f(k))$ pour une certaine fonction g . Ainsi il est possible de décider si $(x, k) \in L$ en temps $g(f(k)) \cdot (|x| + k)^c$ et donc le problème L est FPT.

Supposons maintenant qu'il existe un algorithme décidant si $(x, k) \in L$ en temps $f(k) \cdot |x|^c$ pour une certaine fonction f et une constante c . Si $|x| \leq f(k)$ alors renvoyer (x, k) . Sinon $|x| \geq f(k)$, appliquer l'algorithme FPT sur l'instance, il s'exécute en temps polynomial $f(k) \cdot |x|^c \leq |x|^{c+1}$. Si l'algorithme répond oui, renvoyer une instance positive de taille constante, sinon renvoyer une instance négative de taille constante. Cette procédure correspond à un noyau de taille $f(k)$ pour le problème L . \square

Ce théorème montre l'existence d'un noyau pour tout problème FPT. Néanmoins le noyau qui est construit dans la preuve est de taille exponentielle si le problème est NP-difficile. Ainsi, étant donné un problème FPT, la question pertinente est de savoir s'il admet ou pas un noyau *de taille polynomiale*.

Nous terminons cette sous-section par un exemple classique de noyau polynomial pour VERTEX COVER. Il a été présenté de manière informelle dans l'introduction de cette thèse, nous le décrivons formellement ici afin de présenter le formalisme que nous utilisons pour les noyaux.

Ce noyau se base sur deux règles intuitives (illustrées dans la [figure 1.1](#)). Soit $(G = (V, E), k)$ une instance de VERTEX COVER. La première règle découle de l'observation qu'aucune couverture par sommets minimale ne contient de sommet isolé, en effet, ce sommet ne permet de couvrir aucune arête du graphe. Ainsi, la règle qui suit est trivialement sûre.

Règle 2.1. Soit v un sommet isolé de G . Retirer le sommet v .

La seconde règle suit de l'observation que si un sommet v a un degré plus grand que k , alors il doit être contenu dans toute couverture par sommets de taille au plus k . En effet, si ce sommet v n'est pas dans une telle couverture par sommets S , alors, tout son voisinage $N_G(v)$ doit être contenu dans S , ce qui est une contradiction car $|N_G(v)| > k$ et $|S| \leq k$.

Règle 2.2. Soit v un sommet de degré au moins $k+1$ de G , retirer le sommet v de G et décrémenter le paramètre k de 1.

La sûreté des deux règles présentées est immédiate, néanmoins pour l'exemple nous montrons formellement la sûreté de la [règle 2.2](#).

Lemme 2.1. *La [règle 2.2](#) est sûre.*

Preuve. Soit (G, k) une instance de VERTEX COVER et soit $(G \setminus \{v\}, k-1)$ l'instance obtenue à partir de (G, k) après l'application de la [règle 2.2](#). Nous devons montrer que (G, k) est une instance positive si et seulement si $(G \setminus \{v\}, k-1)$ est une instance positive.

Supposons d'abord que (G, k) est une instance positive du problème, il existe S une couverture par sommets de taille au plus k de G . Observons que le degré du sommet v dans G est au moins $k+1$ par définition de la règle, cela implique que $v \in S$. Observons maintenant que l'ensemble $S \setminus \{v\}$ est une couverture par sommets du graphe $G \setminus \{v\}$, en effet les arêtes de G' sont exactement les arêtes de G qui n'étaient pas incidentes à v et sont donc toutes incidentes à un sommet de $S \setminus \{v\}$. Nous pouvons conclure que $S \setminus \{v\}$ est une couverture par sommets de taille $k-1$ de $G \setminus \{v\}$ et donc que $(G \setminus \{v\}, k-1)$ est une instance positive.

Supposons maintenant que $(G \setminus \{v\}, k-1)$ admette une couverture par sommets S' de taille au plus $k-1$. Par définition toutes les arêtes de G non incidentes à v sont incidentes à S' . Ainsi, toutes les arêtes de G sont incidentes à $S' \cup \{v\}$, et donc (G, k) est une instance positive. En conclusion nous avons montré que (G, k) est une instance positive si et seulement si $(G \setminus \{v\}, k-1)$ l'est aussi, et donc que la [règle 2.2](#) est sûre. \square

Les deux règles présentées sont suffisantes pour monter l'existence d'un noyau avec $k^2 + k$ sommets pour VERTEX COVER.

Théorème 2.2. *VERTEX COVER admet un noyau avec $k^2 + k$ sommets.*

Preuve. Soit (G', k') une instance de VERTEX COVER obtenue à partir de (G, k) par l'application exhaustive des [règles 2.1](#) et [2.2](#). Vu que les deux règles sont sûres, (G', k') admet une couverture par sommets de taille au plus $k' \leq k$ si et seulement si (G, k) admet une couverture par sommets de taille au plus k . Supposons que (G', k') est une instance positive de VERTEX COVER, c'est-à-dire qu'il existe une couverture par sommets de G' de taille au plus k' . Comme G' a été réduit par la [règle 2.1](#), il ne contient pas de sommet isolé, ainsi, tout sommet de G' est contenu dans ou adjacent à une couverture par sommets. De plus, comme G' a été réduit par la [règle 2.2](#), chacun de ses sommets est de degré au plus k' . Ainsi, une couverture par sommets de taille au plus k' peut être adjacente à au plus $k' \cdot k'$ sommets. Nous pouvons donc conclure que G' contient au plus $k'(k'+1) \leq k(k+1)$ sommets, sinon (G', k') serait une instance négative.

Finalement, nous pouvons constater que les [règles 2.1](#) et [2.2](#) peuvent être appliquées exhaustivement à l'instance (G, k) en temps polynomial facilement, il suffit de calculer le degré de chacun des sommets. \square

Rappelons que si la taille d'une instance (G, K) de VERTEX COVER réduite par les [règles 2.1](#) et [2.2](#) contient plus que $k^2 + k$ sommets, alors cette instance est négative et peut être simplement remplacée par une instance trivialement négative (par exemple $(K_2, 0)$).

Remarque 2.1. Dans les problèmes portant sur des graphes, la taille d'un noyau est habituellement donnée qu'en fonction du nombre de sommets, sans prendre en compte le nombre d'arêtes.

2.2.3 Méta-théorèmes

De nombreux problèmes classiques qui sont difficiles à résoudre sur des graphes quelconques, sont en revanche faciles à résoudre sur les arbres. Cela est aussi souvent vérifié dans le cas des graphes de « petite » largeur arborescente, cette dernière mesurant leur similitude avec des arbres. En particulier, les propriétés des décompositions arborescentes de « petite » largeur peuvent être exploitées pour mettre au point des algorithmes efficaces. Un résultat majeur de théorie des graphes est le théorème de Courcelle [39], il garantit que toute propriété pouvant être exprimée en logique monadique du second ordre (abrégée MSO_2) peut être vérifiée en temps linéaire pour les graphes de largeur arborescente bornée. Ce résultat est considéré comme un « *méta-théorème* » car il permet de résoudre toute une classe de problèmes exprimés par une certaine logique, pour toute une classe de graphes caractérisée par un paramètre structural. Dans le cadre de la complexité paramétrée, ce théorème montre que tous les problèmes exprimables en logique monadique du second ordre sont FPT paramétrés par la largeur arborescente. Cette logique permet d'exprimer des problèmes de décision classiques tels que HAMILTONIAN CYCLE ou 3-COLORING, et nous verrons que ce résultat s'adapte également à de nombreux problèmes d'optimisation tels que VERTEX COVER et DOMINATING SET. Dans cette section nous commençons par présenter MSO_2 et le théorème de Courcelle, puis nous présentons une version de ce théorème étendue aux problèmes d'optimisation.

Dans la logique du second ordre, il est possible de quantifier sur les prédicats. Dans la logique monadique du second ordre des graphes (abrégée MSO_2) cette quantification est restreinte aux prédicats unaires. Les prédicats unaires correspondent à des sous-ensembles, ainsi, MSO_2 peut être vue comme l'extension de la logique du premier ordre dans laquelle il est possible de quantifier sur des ensembles d'arêtes ou de sommets. Cette logique est composée :

- Des connecteurs logiques \vee , \wedge , \neg , \Leftrightarrow et \Rightarrow ,
- De variables pour les sommets, les arêtes, les ensembles de sommets et les ensembles d'arêtes,
- Des quantificateurs \forall et \exists qui peuvent porter sur ces variables,
- Des relations :
 - $\text{adj}(u, v)$, où u et v sont des variables de sommet et l'interprétation est que u et v sont adjacents,
 - $\text{inc}(v, e)$, où v est une variable de sommet et e une variable d'arête et l'interprétation est que v est incident à e ,
 - $v \in V'$, où v est une variable de sommet et V' une variable d'ensemble de sommets,
 - $e \in E'$ où e est une variable d'arête et E' une variable d'ensemble d'arêtes,
 - l'égalité de deux variables de même nature.

Commençons par observer qu'avec les relations d'appartenance, il est aisé de définir la relation $X' \subseteq X$, où X et X' sont des variables d'ensembles. Cette logique permet d'exprimer de nombreuses propriétés classiques de graphe. Par exemple, la connexité d'un sous-graphe (V', E') de G où V' est une variable d'ensemble de sommets et E' une variable d'ensemble d'arêtes :

$$\begin{aligned} \text{Connexe}(V', E') &= \forall X \subseteq V', \forall Y \subseteq V' (\text{Partition}(X, Y, V') \Rightarrow \text{Adjacent}(X, Y, E')) \\ \text{Partition}(X, Y, V') &= \forall x \in V' ((x \in X \vee x \in Y) \wedge \neg(x \in X \wedge x \in Y)) \\ \text{Adjacent}(X, Y, E') &= \exists x \in X, \exists y \in Y, \exists e \in E' (\text{inc}(x, e) \wedge \text{inc}(y, e)) \end{aligned}$$

Ou bien la couverture des sommets du graphe par des ensembles de sommets V_1, V_2, \dots, V_k :

$$\text{Couverture}(V_1, V_2, \dots, V_k) = \forall x \in V (x \in V_1 \vee x \in V_2 \vee \dots \vee x \in V_k)$$

Les prédicats qui ont été présentés ici seront utilisés dans le [chapitre 6](#). Nous présentons maintenant le théorème de Courcelle [39] qui implique que tout problème exprimable en MSO_2 est FPT paramétré par la largeur arborescente du graphe d'entrée.

Théorème 2.3 (Courcelle [39]). *Tout problème exprimable en MSO_2 peut être résolu en temps $f(\text{tw}) \cdot n$ sur les graphes à n sommets et de largeur arborescente tw et où f est une fonction dépendant du problème.*

La preuve de ce théorème fournit un algorithme qui prend en entrée un graphe et sa décomposition arborescente (qui peut être obtenue en temps $f(\text{tw}) \cdot O(n)$ [16, 92]) ainsi que la formule MSO_2 correspondant au problème, et calcule la solution. La complexité de cet algorithme dépend de la largeur arborescente mais également de la formule MSO_2 , elle dépend donc du problème.

Les problèmes exprimables en MSO_2 correspondent à des problèmes de décision. Pour couvrir un plus grand ensemble de problèmes, Arnborg et Lagergren [8] proposent une preuve alternative au théorème de Courcelle et montrent que des résultats similaires peuvent être obtenus pour une extension de la logique monadique du second ordre (abrégée EMSO_2), permettant notamment d'exprimer des problèmes d'optimisation. Nous présentons une version simplifiée de EMSO_2 , qui sera suffisante pour l'utilisation qu'on en fera dans ce document. Cette extension peut être définie comme des paires composées d'une formule MSO_2 $\varphi(X_1, \dots, X_l)$ avec des variables d'ensembles X_1, \dots, X_l libres et d'une fonction d'évaluation linéaire $F(|X_1|, \dots, |X_l|)$ portant sur les cardinaux des variables d'ensembles libres. Les problèmes EMSO_2 d'optimisation consistent donc en la recherche d'une assignation des variables X_1, \dots, X_l , satisfaisant $\varphi(X_1, \dots, X_l)$ tout en minimisant ou maximisant l'évaluation de la fonction $F(|X_1|, \dots, |X_l|)$. Prenons comme exemple le problème `MINIMUM VERTEX COVER`, il correspond au problème EMSO_2 , qui cherche à minimiser $F(|S|) = |S|$ avec S vérifiant la propriété :

$$\text{VertexCover}(S) = \forall e \in E, \exists v \in S, \text{inc}(v, e).$$

En outre, dans la version du théorème montrée par Arnborg et Lagergren il est permis d'étiqueter un nombre fini de sommets ou d'arêtes du graphe. Cela permet par exemple d'exprimer la propriété que deux sommets s et t fixés sont dans une même composante connexe :

$$\text{MCC}() = \exists V' \subseteq V, \exists E' \subseteq E (\text{Connexe}(V', E') \wedge s \in V' \wedge t \in V').$$

Le théorème qui suit montre que le [théorème 2.3](#) se généralise à un grand nombre de problèmes d'optimisation. Précisons qu'une fonction d'évaluation est dite *linéaire* si elle est de la forme $F(|X_1|, \dots, |X_l|) = a_0 + \sum_{i=1}^l a_i |X_i|$ pour certaines valeurs $a_0, a_1, \dots, a_l \in \mathbb{R}$.

Théorème 2.4 ([8]). *Tout problème exprimable en un problème EMSO_2 d'optimisation, où la fonction d'évaluation est linéaire, peut être résolu en temps $f(\text{tw}) \cdot n$ sur les graphes à n sommets et de largeur arborescente tw et où f est une fonction dépendant du problème.*

La version plus générale de EMSO_2 présentée par Arnborg et Lagergren [8] permet notamment aux fonctions d'évaluation de prendre en compte des poids sur les sommets et les arêtes du graphe. Ils montrent aussi un théorème similaire pour une classe de fonction d'évaluation plus générale, mais où le temps de résolution est de la forme $f(\text{tw}) \cdot \text{poly}(n)$.

Nous utilisons le [théorème 2.4](#) dans le [chapitre 6](#), où nous montrons que le problème `ISOMETRIC PATH COVER WITH TERMINALS` peut se réduire à un problème EMSO_2 d'optimisation et ainsi montrer qu'il est FPT.

2.3 Analyse de processus de branchement

Les processus de branchement sont fondamentaux et se retrouvent à la fois pour le développement d'algorithmes (notamment paramétrés), ou bien pour analyser des structures arborescentes. Nous présentons dans cette section des techniques classiques pour l'analyse de processus de branchement, en nous basant sur le livre de Fomin et Kratsch [66, Chapitre 2].

Pour résoudre un problème, un algorithme de branchement va brancher sur des sous-problèmes, les résoudre récursivement, puis combiner leurs solutions en une solution du problème initial. Un tel algorithme peut être décrit par un ensemble de règles de branchement et de règles d'arrêt. Les règles de branchement réduisent le problème en différents sous-problèmes plus petits. L'exécution d'un algorithme de branchement peut être caractérisée par un arbre de recherche, dont chaque noeud correspond à un problème (la racine étant le problème initial) et ses fils aux sous-problèmes

créés par une règle de branchement. Si pour chaque noeud, le temps d'exécution pour appliquer la règle est borné par $f(n)$, où n est la taille de l'entrée, alors le temps d'exécution de l'algorithme peut être borné par $|V(T)| \cdot f(n)$. Ainsi pour analyser le temps d'exécution de ces algorithmes, il est nécessaire de savoir borner la taille de leurs arbres de recherche.

Soit $T(k)$ le nombre maximum de feuilles d'un arbre de recherche sur une entrée avec un paramètre k pour un certain algorithme de branchement. Pour borner $T(k)$ l'approche générale est d'analyser les différentes règles de branchement indépendamment. Soit R une règle qui branche en $r \geq 2$ sous-problèmes de paramètres $k - t_1, k - t_2, \dots, k - t_r$. Nous appelons (t_1, t_2, \dots, t_r) le *vecteur de branchement* de la règle R . Pour toute valeur de k suffisamment grande, nous pouvons en déduire la récurrence linéaire suivante :

$$T(k) \leq T(k - t_1) + T(k - t_2) + \dots + T(k - t_r).$$

La solution d'une telle récurrence est de la forme $T(k) = \alpha^k$ où α , appelé *facteur de branchement*, est l'unique solution réelle positive de l'équation :

$$x^k - x^{k-t_1} - x^{k-t_2} - \dots - x^{k-t_r} = 0.$$

Ainsi, si seule la règle R est appliquée, le nombre de feuilles de l'arbre de recherche est $O(\alpha^k)$. Par exemple, dans le cas de l'algorithme pour VERTEX COVER dans la [section 2.2.1](#), la règle branche en deux sous-problèmes de paramètre $k - 1$, elle a donc pour vecteur de branchement $(1, 1)$. La racine réelle positive de $x^k - 2x^{k-1} = 0$ est 2, nous en concluons que l'arbre de recherche a $O(2^k)$ feuilles.

Maintenant, si plusieurs règles peuvent être appliquées, vu que nous cherchons une borne supérieure à $T(k)$, il est suffisant de considérer uniquement la règle avec le plus grand facteur de branchement. Nous en déduisons donc le théorème qui suit.

Théorème 2.5 ([66]). *Soit un algorithme de branchement \mathcal{A} avec les règles R_1, R_2, \dots, R_q . Soit α_i le facteur de branchement de la règle R_i . Le nombre de feuilles de l'arbre de recherche de \mathcal{A} est $O(\alpha^k)$ où $\alpha = \max_{1 \leq i \leq q} \alpha_i$.*

Dans l'introduction de ce document, nous avons présenté un algorithme, proposé par Cai [28], pour résoudre les problèmes de modification de graphes vers des classes de graphes héréditaires qui peuvent être caractérisées par une famille finie d'obstructions. Le principe de l'algorithme est simple : tant que le graphe contient une obstruction, nous branchons sur l'ensemble (fini) des possibilités pour détruire cette obstruction avec les opérations disponibles. Rappelons qu'au plus k modifications sont autorisées. Pour chaque obstruction H_i il existe une règle R_i , qui branche sur d_i sous-problèmes de paramètre $k - 1$. Ainsi le facteur de branchement α_i de chaque règle est égal à d_i , la racine réelle de $x^k - d_i x^{k-1} = 0$. Nous pouvons donc conclure, à l'aide du [théorème 2.5](#), que l'arbre de recherche a $O(d^k)$ feuilles où $d = \max_i d_i$. Vu qu'il est possible de détecter la présence d'une obstruction dans un graphe en temps polynomial, l'algorithme s'exécute en temps $O(d^k) \cdot \text{poly}(n)$ et est donc bien FPT.

Les outils présentés dans cette section sont usuellement utilisés pour de l'analyse d'algorithmes de branchement. Néanmoins, dans ce document nous les utilisons dans le [chapitre 6](#) pour borner la taille d'arbres respectant certaines règles de branchement.

Partie I

Noyaux pour la modification d'arêtes

Chapitre 3

Introduction et généralités

Nous commençons ce chapitre en présentant les résultats connus pour les noyaux de modification d'arêtes vers des sous-classes des graphes cordaux et autres classes de graphes proches. Nous présenterons ensuite des techniques classiques pour élaborer des noyaux pour des problèmes de modification d'arêtes. Enfin nous présenterons l'approche que nous utilisons pour mettre au point les noyaux qui seront présentés dans le reste de cette partie.

Un des premiers noyaux que l'on trouve dans la littérature est un noyau pour MINIMUM FILL-IN [84], qui est le problème de la complétion vers les graphes cordaux. Kaplan *et al.* [84] ont défini le noyau et ont montré une borne $O(k^3)$ sur sa taille, une analyse plus fine proposée par Natanzon *et al.* [115] a ensuite montré que la taille du noyau pouvait être bornée à $O(k^2)$ sommets. Depuis lors, de nombreux noyaux polynomiaux pour des problèmes de modification d'arêtes ont été proposés. La [table 3.1](#) liste les noyaux connus pour des sous-classes de graphes cordaux et autres classes proches. Cette table s'appuie en grande partie sur l'étude de Crespelle *et al.* [41]. Notons que la plupart des classes listées sont présentées dans le diagramme de classes de la [figure 2.4](#). Nous rappelons que la taille des noyaux désigne le nombre de sommets d'une instance réduite, sans prendre en compte le nombre d'arêtes.

La plupart des problèmes de modification d'arêtes sont NP-complets, à l'exception de certaines classes dont la structure est simple. C'est le cas notamment pour la complétion en forêts d'étoiles ou de chemins où la réponse est trivialement « oui » ou « non », en effet ajouter des arêtes ne permet pas de faire devenir un graphe une forêt s'il ne l'était pas déjà. Encore pour la complétion, le problème est facile pour les graphes cluster et les graphes bloc, en effet il suffit d'ajouter toutes les arêtes manquantes aux composantes connexes et biconnexes, respectivement, pour les transformer en cliques. Enfin, de manière surprenante, le problème est polynomial pour l'édition vers les graphes split [78], alors que dans la version complétion et délétion le problème est difficile [114].

Observons que pour une classe très restreinte, à savoir les graphes composés d'une clique et d'un indépendant, il a été récemment montré qu'il existe un noyau avec $O(k/\log k)$ sommets [10], qui est à notre connaissance le seul noyau avec un nombre sous-linéaire de sommets pour les problèmes de modification d'arêtes.

Nous pouvons mentionner un noyau de taille $O(k^3)$ pour le problème FLIP CONSENSUS TREE montré par Komusiewicz et Uhlmann [91]. Ce problème revient au problème de l'édition vers des graphes bipartis $G = (V = V_c \cup V_t, E)$ (V_c et V_t forment des indépendants) où les voisinages des sommets de V_c forment une famille laminaire. Bien que ces graphes ne soient pas cordaux, ils sont proches des graphes trivialement parfaits et des graphes chaînes. De plus, l'approche utilisée pour définir le noyau pour l'édition vers ces graphes dans [91] est très proche de celle que nous utilisons pour le noyau des graphes trivialement parfaits.

Nous pouvons observer que rares sont les noyaux polynomiaux connus pour des familles de graphes caractérisées par des ensembles infinis d'obstructions. Rappelons que dans le cas de la complétion, les trous se comportent comme une famille finie d'obstructions. En effet, tout trou de longueur au moins $k + 4$ nécessite au moins $k + 1$ ajouts d'arêtes pour être rendu cordal, ce qui n'est pas possible pour une complétion de taille k . Ainsi les seules familles caractérisées par des ensembles infinis d'obstructions qui admettent un noyau polynomial sont, dans le cas édition et délétion seulement, les graphes blocs, les graphes 3-leaf power, les graphes strictement cordaux

Classe	Complétion	Délétion	Édition
Cordal	$O(k^2)$ [84, 115]	Ouvert	Ouvert
Cographe	$O(k^3)$ [75]	$O(k^3)$ [75]	$O(k^2 \log k)$ [43]
Trivialement parfait	$O(k^2)$ [10]	$O(k^2)$ [57]	$O(k^2)$ [57]
Threshold	$O(k^2)$ [52]	$O(k^2)$ [52]	$O(k^2)$ [52]
Co-chaîne	$O(k^2)$ [14]	$O(k^2)$ [52]	$O(k^2)$ [52]
Split	$O(k)$ [10]	$O(k)$ [10]	P [78]
Cluster	P	$2k$ [10]	$2k$ [32, 37]
Forêt d'étoiles	P	$4k$ [10]	$4k$ [10]
Forêt de chemins	P	$9k$ [63]	$9k$ [63]
Clique + Indépendant	P	$O(k/\log k)$ [10]	$2k$ [41, Table 1]
Bloc	P	$O(k^2)$ [58]	$O(k^2)$ [58]
3-leaf power	$O(k^3)$ [13]	$O(k^3)$ [13]	$O(k^3)$ [13]
4-leaf power	Ouvert	Ouvert	Ouvert
Strictement cordal	$O(k^3)$ [60]	$O(k^3)$ [60]	$O(k^4)$ [60]
Intervalles	Ouvert	Ouvert	Ouvert
Intervalles propres	$O(k^3)$ [14]	Ouvert	Ouvert
Distances héréditaires	Ouvert	Ouvert	Ouvert
Ptolémaïque	$O(k^4)$ [42]	Ouvert	Ouvert

Table 3.1: Liste des tailles des noyaux polynomiaux connus pour les trois problèmes de modification d'arêtes vers des sous-classes des graphes cordaux et autres classes proches. P signifie que le problème est polynomial à résoudre. Les contributions réalisées au cours de cette thèse sont indiquées en orange.

ainsi que les forêts de chemins. L'existence de noyau polynomial reste ouverte pour des classes caractérisées par des ensembles infinis d'obstructions, telles que les graphes d'intervalles, les graphes distances héréditaires ou les graphes cordaux.

3.1 Règles de réduction classiques

Nous présentons dans cette section des règles de réduction classiques qui seront pour certaines cruciales pour définir nos noyaux. Nous commençons avec une règle qui permet de supprimer toutes les composantes connexes appartenant déjà à la classe de graphes cible.

Règle 3.1. Soit C une composante connexe de G telle que C est dans la classe de graphes \mathcal{G} . Retirer les sommets de C .

Lemme 3.1. La règle 3.1 est sûre pour \mathcal{G} -EDITION (resp. \mathcal{G} -COMPLETION, \mathcal{G} -DELETION) où \mathcal{G} est une classe de graphes héréditaire close par union disjointe.

Preuve. Nous montrons le lemme pour l'édition, les mêmes arguments peuvent être appliqués pour les problèmes de complétion et délétion.

Soit une instance (G, k) et (G', k) l'instance réduite obtenue par l'application de la règle 3.1. Nous montrons que (G, k) est une instance positive de \mathcal{G} -EDITION si et seulement si (G', k) est une instance positive de \mathcal{G} -EDITION. Puisque G' est un sous-graphe induit de G et que \mathcal{G} est héréditaire, toute k -édition de G est aussi une k -édition de G' .

Supposons maintenant que G' admette une k -édition F' et soit $H' = G' \triangle F'$, nous montrons que F' est une k -édition de G . Soit C la composante connexe retirée par la règle 3.1. Rappelons que C est dans la classe de graphes \mathcal{G} . Puisque \mathcal{G} est close par union disjointe, le graphe H résultant

de l'union de H' et C est dans \mathcal{G} . De plus $H = G\Delta F'$ par construction. Nous concluons que F' est une k -édition de G . \square

Nous présentons maintenant un lemme de Bessy *et al.* [13] sur les éditions, complétions et délétions dans les classes de graphes héréditaires closes pour l'ajout de vrais jumeaux. Il montre que pour ces classes, il existe toujours une édition optimale qui édite de la même manière tous les sommets d'une clique critique.

Lemme 3.2 ([13]). *Soit \mathcal{G} une classe de graphes héréditaire close pour l'ajout de vrais jumeaux. Pour tout graphe $G = (V, E)$, il existe une édition (resp. complétion, délétion) optimale F en un graphe $H = G\Delta F$ (resp. $H = G + F$, $H = G - F$) de \mathcal{G} telle que toute clique critique de G est un module clique de H .*

Preuve. Nous prouvons le lemme pour l'édition, les mêmes arguments peuvent être appliqués pour les problèmes de complétion et délétion.

Soit F une édition optimale de G en un graphe de \mathcal{G} , nous allons construire une édition optimale F^* de G en un graphe de \mathcal{G} qui respecte l'énoncé du lemme. Soit K une clique critique de G et soit v un sommet de K contenu dans un nombre minimum de paires de F . Considérons le graphe $H_v = G \setminus (K \setminus \{v\})$, qui appartient à \mathcal{G} par hérédité. Considérons maintenant le graphe H' obtenu à partir de H_v en rajoutant les sommets de $K \setminus \{v\}$ en tant que vrais jumeaux de v . Ce graphe est dans \mathcal{G} car cette classe est close pour l'ajout de vrais jumeaux. Observons que K est un module clique dans H' (mais n'est pas nécessairement maximal). Soit F' l'édition telle que $H' = G\Delta F'$, par le choix de v nous avons que $|F'| \leq |F|$ donc F' est optimale.

En appliquant ce processus sur chacune des cliques critiques de G , nous pouvons obtenir une édition optimale F^* de G en un graphe de \mathcal{G} telle que toute clique critique de G est un module clique dans $H = G\Delta F$. \square

Une édition, complétion ou délétion qui respecte le [lemme 3.2](#) est dite *homogène*. Observons que pour une édition homogène F de G , pour toute paire de cliques critiques K et K' de G , soit $(K \times K') \subseteq F$ soit $(K \times K') \cap F = \emptyset$.

Remarque 3.1. Dans le reste de ce document, nous supposons que toutes les éditions (resp. complétions, délétions) pour les classes de graphes héréditaires closes pour l'ajout de vrais jumeaux sont homogènes.

Cette propriété sur les classes de graphes que nous considérons nous permet aussi de montrer la sûreté de la règle suivante, qui est cruciale pour nos algorithmes de noyaux.

Règle 3.2. Soit $K \subseteq V$ une clique critique de G telle que $|K| > k + 1$. Retirer $|K| - (k + 1)$ sommets de K .

Lemme 3.3 ([13]). *La règle 3.2 est sûre pour \mathcal{G} -EDITION (resp. \mathcal{G} -COMPLETION, \mathcal{G} -DELETION) où \mathcal{G} est une classe de graphes héréditaire close pour l'ajout de vrais jumeaux.*

Preuve. Nous montrons le lemme pour l'édition, les mêmes arguments peuvent être appliqués pour les problèmes de complétion et délétion.

Soit une instance (G, k) et (G', k) l'instance réduite obtenue par l'application de la [règle 3.2](#). Nous montrons que (G, k) est une instance positive de \mathcal{G} -EDITION si et seulement si (G', k) est une instance positive de \mathcal{G} -EDITION. Puisque G' est un sous-graphe induit de G et que \mathcal{G} est héréditaire, toute k -édition de G est aussi une k -édition de G' .

Supposons maintenant que G' admette une k -édition F' et soit $H' = G'\Delta F'$, nous montrons que F' est une k -édition de G . Soit K' les sommets de K qui n'ont pas été retirés par la [règle 3.2](#). Observons que K' est une clique module de H' (rappelons que nous supposons que F' est homogène). Soit H le graphe obtenu à partir de H' en rajoutant les sommets de $K \setminus K'$ en tant que vrais jumeaux des sommets de K' . Ce graphe est dans \mathcal{G} car cette classe est close pour l'ajout de vrais jumeaux. Puisque $|F'| \leq k$ et que $|K'| > k$ nous en déduisons que F' n'affecte aucun sommet de K' sinon $|F'| > k$. Par conséquent F' est l'édition telle que $H = G\Delta F'$. Nous déduisons que F' est une k -édition de G en un graphe de \mathcal{G} , ce qui conclut la preuve du lemme. \square

Remarquons que si la classe de graphes est close pour l'ajout de *faux* jumeaux, alors une version de cette règle portant sur les indépendants critiques (modules indépendants maximaux) peut être utilisée. Cette version de la règle a été montrée indépendamment de la version vrais jumeaux par Komusiewicz et Uhlmann [91].

Le lemme qui suit est une conséquence immédiate de la [règle 3.2](#). Il nous permet de borner simplement le nombre de sommets contenus dans les cliques critiques qui contiennent des sommets affectés par une k -édition. Notons que ce lemme est aussi vérifié dans le cas de la complétion et de la déletion.

Lemme 3.4. *Soit (G, k) une instance de \mathcal{G} -EDITION réduite par la [règle 3.2](#) et F une k -édition de G vers la classe \mathcal{G} . Alors le nombre de sommets contenus dans des cliques critiques affectées de $H = G\Delta F$ est borné par $O(k^2)$.*

Preuve. Le graphe H contient au plus $2k$ sommets affectés par F et donc contient au plus $2k$ cliques critiques affectées par F . Chacune de ces cliques critiques contient au plus $k + 1$ sommets non affectés, en effet, ces sommets forment un module clique de G qui a été réduit par la [règle 3.2](#). Ainsi, les cliques critiques affectées de H contiennent au total $2k(k + 1) + 2k$ sommets. \square

Nous terminons en présentant une règle classique pour les algorithmes de noyaux, qui porte sur les *tournesols* (sunflower). Un *tournesol* \mathcal{F} avec r pétales et un centre Y est une collection d'ensembles non vides $\{S_1, \dots, S_r\}$ telle que $S_i \cap S_j = Y$ pour tout $i \neq j$. Une règle dite de tournesol pour un problème de modification d'arêtes est de la forme suivante :

Règle 3.3. Soit (G, k) une instance de \mathcal{G} -EDITION et soit $\mathcal{F} = \{S_1, \dots, S_r\}$ un tournesol de centre $\{u, v\}$ avec $r > k$. Si S_1, \dots, S_r sont des obstructions de la classe de graphes \mathcal{G} , alors réduire l'instance à $(G\Delta\{u, v\}, k - 1)$.

Observons que cette règle est sûre. En particulier, la paire $\{u, v\}$ doit être dans toute k -édition de G vers la classe \mathcal{G} . En effet, une édition doit contenir au moins une paire de sommets de chaque obstruction du graphe initial. Ainsi, pour une k -édition F de G vers la classe \mathcal{G} , si $\{u, v\} \notin F$, pour chacune des obstructions S_i , il existe deux sommets $x, y \in S_i$ tels que $\{x, y\} \in F$. Puisque $S_i \cap S_j = \{u, v\}$ pour tout $i \neq j$, cela implique $|F| \geq r > k$, ce qui est une contradiction. Nous pouvons conclure que F doit contenir la paire $\{u, v\}$.

Dans les noyaux que nous présentons dans les [chapitres 4 et 5](#), l'utilisation de règles de tournesol n'est pas nécessaire pour obtenir les bornes sur la taille de ces noyaux. Cependant, d'un point de vue pratique, s'il fallait implémenter les algorithmes, il serait intéressant de s'en servir afin d'obtenir un noyau plus compact. Cette règle permet notamment de diminuer la taille du paramètre, ce qui est particulièrement intéressant puisque les algorithmes FPT, qui sont utilisés pour résoudre une instance réduite par un noyau, ont une dépendance exponentielle en le paramètre. Notons néanmoins que cette règle n'est pas intrinsèquement polynomiale par sa définition, mais il est souvent possible d'en trouver des déclinaisons polynomiales.

3.2 Reconnaissance et application polynomiale

Dans cette section nous abordons en premier la reconnaissance des classes de graphes en temps polynomial. Nous présentons ensuite la notion de décomposition modulaire et montrons que les [règles 3.1 et 3.2](#) introduites dans la section précédente peuvent être appliquées de manière exhaustive en temps polynomial. Toutefois, tout au long de cette partie du document qui porte sur les noyaux, nous n'attachons pas une importance particulière à l'efficacité de nos algorithmes, pourvu qu'ils soient polynomiaux (tous nos efforts seront concentrés en revanche pour minimiser la taille des noyaux).

Dans les noyaux pour les problèmes de modifications de graphes, une des idées principale est de réduire les « parties » des graphes qui appartiennent déjà à la classe de graphes cible. C'est par exemple le cas de la [règle 3.1](#). Il est donc essentiel de pouvoir *reconnaître* une classe de graphes \mathcal{G} donnée (c'est-à-dire vérifier l'appartenance d'un graphe à \mathcal{G}) en temps polynomial.

Pour détecter la présence d'un sous-graphe induit de taille d dans un graphe, une approche naïve consiste en tester tous les ensembles de d sommets du graphe, ce qui prend un temps $O(n^d)$. Avec cette approche, si une classe de graphes est caractérisée par une famille finie d'obstructions il

est possible de tester l'appartenance d'un graphe G à cette dernière en temps polynomial. En effet, il suffit de vérifier la présence ou non de chacune des obstructions dans G . Néanmoins il existe en général des algorithmes plus efficaces pour tester l'appartenance à ces classes de graphes. Notons que les classes de graphes que nous avons présentées dans la [section 2.1.2](#) peuvent être reconnues pour la plupart en temps linéaire, sinon en temps polynomial. Par exemple les graphes cordaux et les graphes d'intervalles peuvent être reconnus en temps linéaire [77, 119]. Il a été récemment montré que les r -leaf power peuvent être reconnus en temps polynomial pour tout r fixé [96].

Décomposition modulaire Nous rappelons qu'un *module* M de G est un ensemble de sommets qui ont le même voisinage à l'extérieur de M , c'est-à-dire $N_G(u) \setminus M = N_G(v) \setminus M$ pour tout $u, v \in M$. Un graphe peut contenir un nombre exponentiel de modules, c'est notamment le cas de la clique où tout sous-ensemble de sommets est un module. Un module M est dit *fort* si pour tout autre module M' , $M \cap M' = \emptyset$, $M \subseteq M'$ ou $M' \subseteq M$. Notons que l'ensemble des modules forts d'un graphe forment une famille laminaire. La *décomposition modulaire* correspond à l'arbre d'inclusion des modules forts où la racine est étiquetée par V et les feuilles par les singletons $\{v\}, v \in V$. Cette décomposition peut être construite en temps $O(n + m)$ [124].

Nous utilisons la décomposition modulaire pour montrer que des règles portant sur des modules peuvent être appliquées exhaustivement en temps polynomial. Nous pouvons observer que les composantes connexes et les cliques critiques d'un graphe sont des modules forts, il est donc possible de les énumérer en temps polynomial. Ces observations impliquent le lemme suivant, qui assure que les [règles 3.1](#) et [3.2](#) peuvent être appliquées en temps polynomial. Notons que les composantes connexes peuvent aussi être énumérées via des parcours en profondeur. Les cliques critiques peuvent être énumérées en partitionnant les sommets en fonction de leurs voisinages fermés.

Lemme 3.5. *La règle 3.1 peut être appliquée en temps polynomial si la classe de graphes \mathcal{G} peut être reconnue en temps polynomial. La règle 3.2 peut être appliquée en temps polynomial pour toute classe de graphes.*

3.3 Approche par décomposition et sommets affectés

Dans cette section nous présentons l'approche que nous utilisons pour élaborer les noyaux qui sont présentés dans cette partie, que nous appelons « décomposition et sommets affectés ». Cette approche exploite la structure arborescente des classes de graphes que nous considérons. Elle se base sur une observation simple mais cruciale, qui est qu'une édition de taille k d'un graphe n'affecte pas plus que $2k$ sommets de ce dernier. Si la classe de graphes est héréditaire, alors le graphe induit par les sommets qui ne sont pas affectés appartient à la classe de graphes \mathcal{G} cible. Dans un sens cela veut dire qu'une « grande partie » du graphe appartient déjà à la classe \mathcal{G} . Seulement, nous ne pouvons pas connaître ces sommets affectés sans résoudre le problème. Par conséquent pour élaborer nos noyaux, nous supposons avoir une édition et, à l'aide d'une décomposition spécifique à la classe de graphes considérée, nous étudions comment se structurent les sommets affectés par rapport au reste du graphe. L'objectif est comprendre comment détecter et réduire, dans le graphe initial, les parties qui seront affectées et les parties qui ne le seront pas.

Pour illustrer l'approche « décomposition et sommets affectés », nous prenons l'exemple du noyau cubique pour le problème 3-LEAF POWER EDITION présenté par Bessy *et al.* [13]. Commençons par rappeler que les graphes 3-leaf power sont les graphes $\{\text{trou}, \text{gemme}, \text{dard}, \text{taureau}\}$ -free. Ils peuvent également être caractérisés de manière simple en fonction de leurs cliques critiques. Rappelons que les cliques critiques sont des modules cliques maximaux, ce sont aussi des ensembles de vrais jumeaux maximaux. Remarquons que l'ensemble $\mathcal{K}(G)$ des cliques critiques d'un graphe G partitionne son ensemble de sommets, cela permet de définir le graphe des cliques critiques.

Définition 3.1 (Graphe des cliques critiques). Soit $G = (V, E)$ un graphe. Le *graphe des cliques critiques* de G est le graphe $\mathcal{C}(G) = (\mathcal{K}(G), E_{\mathcal{C}})$ avec $E_{\mathcal{C}} = \{KK' \mid \forall u \in K, \forall v \in K', uv \in E\}$.

Théorème 3.1 ([23]). *Un graphe est 3-leaf power si et seulement si son graphe des cliques critiques est une forêt.*

C'est sur le graphe des cliques critiques que repose la preuve de la taille de leur noyau. Considérons une instance positive $(G = (V, E), k)$ de 3-LEAF POWER EDITION, F une k -édition de G et soit $H = G \triangle F$. Le graphe G n'est pas 3-leaf power, donc $\mathcal{C}(G)$ n'est pas un arbre, néanmoins $\mathcal{C}(H)$ en est un (illustré dans la figure 3.1). Puisque F est de taille au plus k , il existe au plus $2k$ cliques critiques de $\mathcal{C}(H)$ qui contiennent des sommets affectés. Soit A ces cliques critiques et T le sous-arbre connexe minimal de $\mathcal{C}(H)$ qui couvre les cliques critiques de A . Notons que nous supposons ici que H est connexe, s'il ne l'était pas il suffirait d'appliquer le raisonnement sur chacune des composantes connexes. Définissons A' comme l'ensemble des cliques critiques de degré au moins 3 dans T . Considérer les cliques critiques de A' nous permettra de simplifier l'analyse de la taille de l'instance. Puisque T est un arbre, $|A'| \leq |A| \leq 2k$.

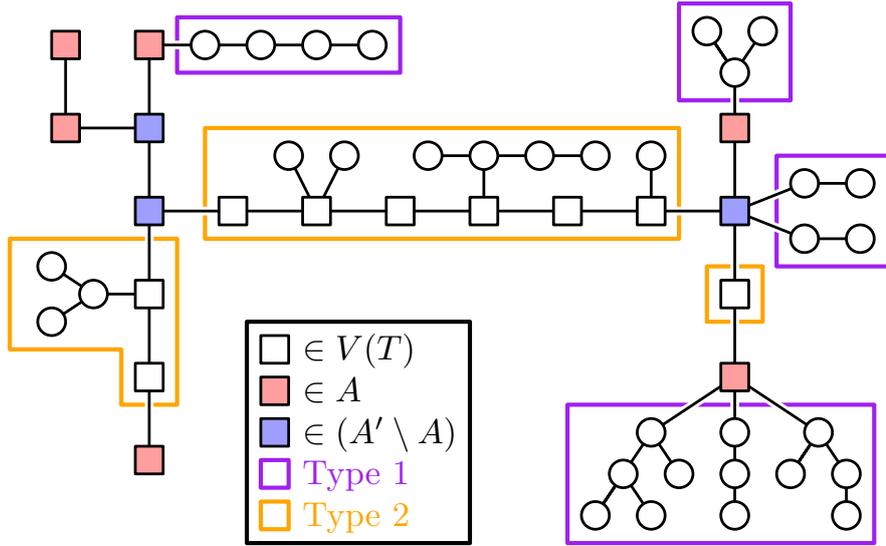


Figure 3.1: Illustration de la structure du graphe des cliques critiques $\mathcal{C}(H)$. Notons que les composantes de type 1 et type 2 n'ont pas encore été réduites.

Tout d'abord, les graphes 3-leaf power sont clos pour l'ajout de vrais jumeaux, ainsi la règle 3.2 est sûre, ce qui permet de borner par $k + 1$ la taille de chacune des cliques critiques. Ainsi, avec le lemme 3.4, nous pouvons conclure que les cliques critiques de A et A' contiennent $O(k^2)$ sommets.

Maintenant, il reste à borner la taille de $\mathcal{C}(H) \setminus (A \cup A')$. Observons que par le choix de A' , les composantes connexes de ce graphe sont adjacentes à exactement zéro, une ou deux cliques critiques de $A \cup A'$ dans $\mathcal{C}(H)$. De plus, ces composantes connexes induisent toutes des graphes 3-leaf power dans G puisqu'elles ne contiennent pas de sommets affectés par l'édition. Si une de ces composantes connexes n'est adjacente à aucun sommet de $A \cup A'$ dans $\mathcal{C}(H)$, alors elle induit une composante connexe 3-leaf power dans le graphe initial. Dans ce cas, elle peut être retirée simplement à l'aide de la règle 3.1.

Considérons les composantes connexes de $\mathcal{C}(H) \setminus (A \cup A')$ adjacentes à une même clique critique a de $A \cup A'$, nous les appelons composantes de type 1. À l'aide de règles de réduction portant sur des sous-graphes 3-leaf power, Bessy *et al.* [13] montrent que le nombre de cliques critiques contenues dans une composante de type 1 peut être borné par $O(k)$, et que cette composante contient donc au total $O(k^2)$ sommets. Puisque $|A \cup A'| = O(k)$, les composantes de type 1 contiennent au total $O(k^3)$ sommets.

Enfin, considérons les composantes connexes $\mathcal{C}(H) \setminus (A \cup A')$ adjacentes à deux sommets de $A \cup A'$, nous les appelons composantes de type 2. Une composante de type 2 contient le chemin de T d'un sommet de $A \cup A'$ à un autre. Puisque T est un arbre il existe au plus $|A \cup A'|$ composantes de type 2. Bessy *et al.* [13] montrent que le nombre de cliques critiques contenues dans une composante de type 2 peut être borné par $O(1)$, composante qui contient donc au total $O(k)$ sommets. Puisque $|A \cup A'| = O(k)$, les composantes de type 2 contiennent au total $O(k^2)$ sommets. Cette analyse permet de montrer un noyau qui contient $O(k^3)$ sommets pour le problème 3-LEAF POWER EDITION.

L'approche par « décomposition et sommets affectés » que nous utilisons pour élaborer nos algorithmes de noyaux est similaire à celle de Bessy *et al.* [13] que nous venons de présenter. Elle s'adapte aux structures arborescentes spécifiques des sous-classes des graphes cordaux que nous considérons, mais en suivant la même démarche. Notamment, dans tous les noyaux que nous présentons nous ferons apparaître un sur-ensemble des « sacs affectés » (jouant les mêmes rôles que les ensembles A et A' ci-dessus), puis nous identifierons des parties du graphe dites de « type 1 » ou de « type 2 », au sens où il s'agit des parties de la décomposition reliées à un ou deux endroits à des sacs affectés. Le challenge est de comprendre comment les détecter et les réduire.

Dans le reste de cette section, nous présentons brièvement divers problèmes de modification d'arêtes pour lesquels des noyaux polynomiaux ont été élaborés en utilisant une approche « décomposition et sommets affectés ». Cependant précisons que cette approche n'a aucune définition formelle, et dire qu'un noyau l'utilise n'est qu'une appréciation de notre part, particulièrement quand nous ne sommes pas à l'origine des noyaux en question.

Nous commençons avec les noyaux pour les problèmes BLOCK GRAPH EDITION et STRICTLY CHORDAL EDITION que nous présenterons dans le chapitre 5. Les graphes blocs peuvent être vus comme des arbres de cliques, les graphes strictement cordaux sont les graphes dont le graphe des cliques critiques est un graphe bloc. En particulier nous montrerons que les graphes blocs, comme les arbres, admettent un sous-graphe couvrant tous les sommets affectés qui possède un nombre linéaire en k de sommets de degré au moins 3. De plus, les règles de réduction que nous utilisons pour les graphes strictement cordaux peuvent être vues comme une généralisation de celles utilisées par Bessy *et al.* [13] pour les graphes 3-leaf power.

Dans le chapitre 4, nous exploitons pour le problème TRIVIALY PERFECT EDITION une caractérisation des graphes trivialement parfaits, qui est la décomposition en clique universelle. Cette décomposition correspond à un arbre enraciné étiqueté par des ensembles de sommets (qui sont des cliques critiques), de plus les sommets d'un sac d'un noeud u sont adjacents à tous les sommets dans les sacs des noeuds descendant de u . Nous montrerons également dans le chapitre que les règles que nous utilisons pour les graphes trivialement parfaits peuvent être adaptées de manière immédiate aux graphes threshold. Ces derniers sont une restriction des graphes trivialement parfaits où la décomposition en clique universelle est un chemin. Enfin nous pouvons également mentionner le noyau pour le problème FLIP CONSENSUS TREE, montré par Komusiewicz et Uhlmann [91]. Ce problème correspond à un problème d'édition pour lequel la classe de graphes cible admet une décomposition proche de la décomposition en clique universelle, à la différence que les sacs correspondent à des indépendants critiques et que l'adjacence entre les sommets des sacs n'est qu'entre les sommets internes et leurs feuilles descendantes.

Pour le problème PROPER INTERVAL COMPLETION, Bessy et Perez [14] montrent un noyau avec $O(k^3)$ sommets qui se base également sur une approche « décomposition et sommets affectés ». En particulier, les sommets des graphes d'intervalles propres peuvent être décomposés en un ensemble de chemins de cliques qui partitionne les sommets du graphe de manière à ce que les voisinages des sommets d'une clique dans la clique consécutive soient fortement contraints : qu'ils forment une famille imbriquée. La décomposition étant un ensemble de chemins, les cliques qui ne contiennent pas de sommets affectés forment des sous-chemins qui peuvent être adjacents à zéro, une ou deux cliques affectées. Bessy et Perez montrent que la taille d'une clique qui contient des sommets affectés, ainsi que les sous-chemins de cliques non affectés peuvent être bornés.

Un exemple intéressant d'utilisation de l'approche « décomposition et sommets affectés » est le cas du noyau polynomial pour le problème PTOLEMAIC COMPLETION présenté par Crespelle *et al.* [42]. Rappelons que les graphes ptolémaïques sont cordaux et distances héréditaires, de plus ils sont une super-classe des graphes strictement cordaux. La différence ici est que la décomposition qui est exploitée ne partitionne pas les sommets. Cette décomposition est le graphe des cliques maximales et de leurs intersections, qui, dans le cas des graphes ptolémaïques, est un arbre. Crespelle *et al.* réussissent en particulier à borner le nombre de sacs de la décomposition qui contiennent des sommets affectés. De plus des structures de « type 1 » et de « type 2 » apparaissent également dans les parties de la décomposition qui ne contiennent pas de sommets affectés.

Un dernier exemple est celui du noyau avec $O(k^2 \log k)$ sommets pour le problème COGRAPH EDITION présenté Crespelle *et al.* [43]. Ce noyau exploite la caractérisation des cographes par les *coarbres*. Pour tout cographe $G = (V, E)$, il existe un arbre enraciné $T = (V_T, E_T)$ tel que les feuilles de T correspondent aux sommets de G et que les noeuds internes sont étiquetés avec « \cup »

et « \otimes ». De plus, deux sommets $u, v \in V$ sont adjacents dans G si et seulement si leur plus petit ancêtre commun dans T est un noeud étiqueté par « \otimes ». Un tel arbre est appelé un *coarbre*. Pour borner la taille d'une instance réduite, Crespelle *et al.* utilisent le coarbre d'un graphe édité. Les « parties affectées » du coarbre correspondent aux sommets affectés par l'édition, ainsi que les chemins dans l'arbre entre chaque paire de sommets affectés. C'est en ça que leur approche est semblable à l'approche « décomposition et sommets affectés ».

Pour terminer, mentionnons que d'autres approches sont utilisées dans la littérature. Pour certains problèmes, des règles de tournesol et des arguments combinatoires suffisent pour obtenir des noyaux polynomiaux. Par exemple pour le problème TRIVIALY PERFECT COMPLETION, une règle de tournesol ajoutant les non-arêtes qui sont contenues dans $k + 1$ obstructions, ainsi que la suppression des sommets qui ne sont dans aucune obstruction permet d'obtenir un noyau avec $O(k^2)$ sommets [10]. Cependant, la règle supprimant les sommets qui ne sont dans aucune obstruction n'est pas sûre pour les versions délétion et édition de ce problème. Une autre approche est celle du *modulateur*, qui, informellement, est un sous-ensemble de sommets X tel que le graphe $G \setminus X$ appartient à la classe cible. L'objectif est donc de borner $G \setminus X$ en prenant en compte ses interactions avec les sommets de X . C'est notamment l'approche utilisée pour les noyaux TRIVIALY PERFECT EDITION [54] et THRESHOLD EDITION [52]. Nous estimons toutefois que l'approche par « décomposition et sommets affectés » est très générale et peut permettre d'obtenir des noyaux plus compacts.

Chapitre 4

Graphes trivialement parfaits

Dans ce chapitre nous présentons un noyau quadratique pour le problème TRIVIALY PERFECT EDITION ainsi que ses variantes complétion et délétion. Les résultats présentés ici ont fait l'objet de deux publications, une première avec Anthony Perez et Ioan Todinca [61] où nous décrivons un noyau avec $O(k^3)$ sommets. Plus récemment avec Anthony Perez [57] nous avons amélioré ce noyau à $O(k^2)$ sommets en analysant plus finement les structures introduites dans le premier article.

Les graphes *trivialement parfaits*, aussi connus sous le nom *quasi-threshold* sont les graphes sans C_4 ou P_4 comme sous-graphes induits. Ce sont aussi les graphes où chaque sous-graphe induit connexe contient un sommet universel, c'est-à-dire un sommet adjacent à tous les autres. La NP-complétude des problèmes de modification d'arêtes vers les graphes trivialement parfaits a d'abord été montrée pour les versions complétion et délétion [27, 121], celle de la version édition a été montrée plus tardivement par Nastos et Gao [113]. Dans leur article Nastos et Gao mettent en évidence les liens que les graphes trivialement parfaits ont avec certains types de réseaux sociaux. En particulier ils montrent que ces graphes peuvent modéliser ce qu'ils appellent des « groupes familiaux », c'est-à-dire des communautés présentant une structure hiérarchique (comme dans une famille ou une entreprise). Le problème TRIVIALY PERFECT EDITION revient donc à mesurer la différence d'un réseau social à une structure hiérarchique. Les graphes associés à des réseaux sociaux étant très grands, des heuristiques ont naturellement été étudiées, qu'elles soient approchées [21, 22] ou exactes [73].

Concernant la complexité paramétrée de ces problèmes, observons que le [théorème 1.1](#) implique que le problème TRIVIALY PERFECT EDITION est FPT. Notons que pour TRIVIALY PERFECT DELETION il existe un algorithme qui prend un temps $O(2.42^k(n + m))$ [101]. Concernant les algorithmes de noyaux, pour TRIVIALY PERFECT COMPLETION un noyau avec $O(k^3)$ sommets avait été annoncé par Guo [76] mais la preuve n'a jamais été publiée. Un premier noyau avec $O(k^7)$ sommets a été montré par Drange et Pilipczuk [54] pour les trois versions du problème. Plus récemment un noyau avec $O(k^2)$ sommets pour la version complétion seulement a été montré par Bathie *et al.* [10].

Structure du chapitre Nous présentons dans ce chapitre un noyau avec $O(k^2)$ sommets pour TRIVIALY PERFECT EDITION qui fonctionne aussi pour les variantes complétion et délétion. Nous commençons dans la [section 4.1](#) par donner quelques propriétés structurelles des graphes trivialement parfaits ainsi que d'autres notions qui nous seront utiles. Ensuite dans la [section 4.2](#) nous présentons notre noyau pour TRIVIALY PERFECT EDITION, en donnant les règles et montrant leur sûreté, puis qu'elles peuvent être appliquées en temps polynomial, et finissons par borner la taille des instances réduites. Nous montrons dans la [section 4.3](#) que le noyau présenté pour les graphes trivialement parfaits peut être aisément adapté pour le problème THRESHOLD EDITION et ses variantes complétion et délétion.

4.1 Définitions et propriétés

Graphes trivialement parfaits Diverses caractérisations des graphes trivialement parfaits sont connues, nous donnons dans ce qui suit celles utiles pour notre algorithme de noyau. Nous rappelons que ce sont les graphes $\{C_4, P_4\}$ -free.

Proposition 4.1 ([129]). *La classe des graphes trivialement parfaits peut être définie récursivement comme suit :*

- un sommet seul est un graphe trivialement parfait,
- ajouter un sommet universel à un graphe trivialement parfait résulte en un graphe trivialement parfait,
- l'union disjointe de deux graphes trivialement parfaits résulte en un graphe trivialement parfait.

Cette proposition implique notamment que les graphes trivialement parfaits sont clos par union disjointe et pour l'ajout de vrais jumeaux. Elle implique également que tout sous-graphe induit connexe d'un graphe trivialement parfait contient un sommet universel.

La structure que nous définissons maintenant, à savoir la *décomposition en clique universelle* caractérise les graphes trivialement parfaits et est la structure centrale pour notre noyau. Elle est illustrée dans la [figure 4.1](#). Nous rappelons que la clique universelle d'un graphe est l'ensemble de ses sommets universels.

Définition 4.1 (Décomposition en clique universelle, [53]). Une décomposition en clique universelle d'un graphe connexe $G = (V, E)$ est une paire $\mathcal{T} = (T = (V_T, E_T), \mathcal{B} = \{B_t\}_{t \in V_T})$ où T est un arbre enraciné et \mathcal{B} est une partition de l'ensemble de sommets V en sous-ensembles disjoints non vides telle que :

- si $vw \in E$ et $v \in B_t, w \in B_s$ alors s et t sont sur un même chemin d'une feuille à la racine, avec possiblement $s = t$,
- pour chaque noeud $t \in V_T$, l'ensemble B_t de sommets est la clique universelle du sous-graphe $G[\bigcup_{s \in V(T_t)} B_s]$, où T_t désigne le sous arbre de T enraciné en t .

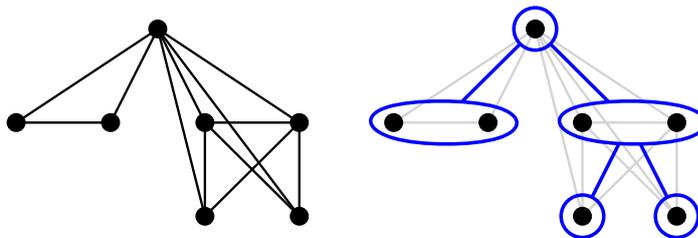


Figure 4.1: Exemple d'un graphe trivialement parfait et sa décomposition en clique universelle.

Les sommets de T sont appelés *noeuds* de la décomposition, les ensembles de \mathcal{B} sont appelés *sacs*. Nous abuserons parfois de la notation et identifierons les noeuds de T avec les sacs correspondants dans \mathcal{B} . Une manière simple de comprendre la [définition 4.1](#) est d'observer qu'une telle décomposition peut être obtenue en supprimant l'ensemble U des sommets universels de G , puis en répétant récursivement ce processus sur chaque composante connexe de $G \setminus U$.

Rappelons que les cliques critiques sont des ensembles maximaux de vrais jumeaux. Nous pouvons donc remarquer que chaque sac de la décomposition en clique universelle correspond à une clique critique de G . De plus, remarquons que dans une décomposition en clique universelle, chaque noeud t de T qui n'est pas une feuille a au moins deux enfants, sinon B_t ne contiendrait pas tous les sommets universels de $G[\bigcup_{s \in V(T_t)} B_s]$.

Lemme 4.1 ([53]). *Un graphe connexe admet une décomposition en clique universelle si et seulement s'il est trivialement parfait. De plus, une telle décomposition est unique à isomorphisme près.*

Nous pouvons observer que construire une décomposition en clique universelle peut être fait en temps polynomial en identifiant de manière itérative les cliques universelles et les composantes connexes. Enfin, la définition 4.1 et le lemme 4.1 peuvent être naturellement étendus aux graphes trivialement parfaits non connexes, en considérant une *forêt enracinée* au lieu d'un arbre enraciné. Plus précisément, la décomposition en clique universelle d'un graphe non connexe $G = (V, E)$ est une forêt enracinée de décompositions en clique universelle de ses composantes connexes. Un tel graphe est donc trivialement parfait si et seulement s'il admet une décomposition en clique universelle ayant la structure d'une forêt enracinée.

Nous définissons une nouvelle caractérisation des graphes trivialement parfaits en terme de cliques maximales et familles imbriquées, illustrée dans la figure 4.2.

Lemme 4.2. *Soit $G = (V, E)$ un graphe, $S \subseteq V$ une clique maximale de G et K_1, \dots, K_r les composantes connexes de $G \setminus S$. Le graphe G est trivialement parfait si et seulement si les conditions suivantes sont vérifiées :*

- (i) $G[S \cup V(K_i)]$ est trivialement parfait pour tout $1 \leq i \leq r$,
- (ii) $\{N_G(V(K_i)) \mid 1 \leq i \leq r\}$ est une famille imbriquée,
- (iii) les sommets de K_i forment un module de G pour tout $1 \leq i \leq r$.

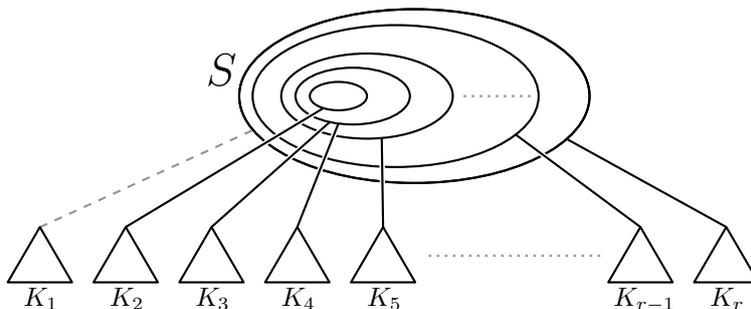


Figure 4.2: Illustration du lemme 4.2, où S est une clique maximale et les K_i sont des composantes connexes trivialement parfaites de $G \setminus S$.

Preuve. Nous supposons que G est connexe, s'il ne l'était pas, nous pourrions utiliser le même raisonnement pour chacune de ses composantes connexes (les sommets d'une composante connexe forment un module avec un voisinage vide). Observons que si G est trivialement parfait, une clique maximale de G correspond aux sommets contenus dans les sacs sur un chemin allant de la racine à l'une des feuilles de la décomposition en clique universelle de G .

Nous commençons par supposer que G est un graphe trivialement parfait et soit $S \subseteq V$ une clique maximale de G . La condition (i) est trivialement vérifiée par hérédité des graphes trivialement parfaits. Soit $\mathcal{T} = (T, \mathcal{B})$ la décomposition en clique universelle de G et $p = (t_1, \dots, t_l)$ le chemin allant de la racine de T à l'une de ses feuilles tel que $S = \bigcup_{1 \leq i \leq l} B_{t_i}$. Observons que pour chaque composante connexe K_i , $1 \leq i \leq r$, nous pouvons trouver un sous-arbre T_x de T tel que $V(K_i) = \bigcup_{s \in V(T_x)} B_s$, où x est un enfant d'un noeud t_j du chemin p . Il découle de la définition d'une décomposition en clique universelle que pour $v \in V(K_i)$, $N_G(v) \setminus V(K_i) = \bigcup_{1 \leq s \leq j} B_{t_s}$, donc la condition (iii) est vérifiée. De plus, cette observation implique que la famille $\{N_G(V(K_i)) \mid 1 \leq i \leq r\}$ est égale à la famille $\{\bigcup_{1 \leq s \leq i} B_{t_s} \mid 1 \leq i \leq l\}$ qui est imbriquée par définition. Ainsi la condition (ii) est vérifiée.

Nous supposons maintenant que G est un graphe qui vérifie les conditions (i), (ii) et (iii). Nous construisons la décomposition en clique universelle de G . Soit N_1, \dots, N_l les ensembles de la famille imbriquée $\{N_G(K_i) \mid 1 \leq i \leq r\}$ avec $N_1 \subseteq \dots \subseteq N_l$. De plus, définissons les cliques $C_1 = N_1$ et $C_i = N_i \setminus N_{i-1}$ pour $i \in \{2, \dots, l\}$. Les cliques C_1, \dots, C_l partitionnent S en cliques critiques de G puisque les sommets de chaque K_i forment un module et que par construction $N_G(C_i) \subseteq \dots \subseteq N_G(C_1)$. Définissons le chemin $p = \{t_1, \dots, t_l\}$ et à chaque noeud t_i de p associons le sac $B_{t_i} = C_i$. Puisque K_i est trivialement parfait, il admet une décomposition en clique

universelle. Pour chaque K_i , $1 \leq i \leq r$, prenons $j \in \{1, \dots, l\}$ tel que $N_G(V(K_i)) = C_1 \cup \dots \cup C_j$ et attachons la racine de la décomposition universelle de K_i au noeud t_j . Ce processus aboutit à la décomposition en clique universelle de G (dont la racine est t_1), nous pouvons conclure que G est trivialement parfait. \square

Bourrage et substitution Nous définissons maintenant des structures et opérateurs utiles pour définir nos règles de réduction et prouver leur sûreté.

La notion de r -bourrage est utilisée dans les règles de réduction pour assurer la présence de sommets non affectés dans des ensembles ordonnés de sous-ensembles de sommets tels que des cliques critiques ou des modules trivialement parfaits.

Définition 4.2 (r -bourrage). Soit $\mathcal{S} = \{C_1, \dots, C_q\}$ une collection ordonnée d'ensembles disjoints. Le sous-ensemble $\mathcal{C} = \{C_1, \dots, C_p\}$ est un r -bourrage de \mathcal{S} si p est le plus petit élément de $\{1, \dots, q\}$ tel que $\sum_{i=1}^p |C_i| \geq r$.

Par un léger abus de notation, nous utilisons \mathcal{C} pour désigner à la fois $\{C_1, \dots, C_p\}$ et l'ensemble $\cup_{i=1}^p C_i$. Faisons maintenant une observation simple. Par définition et par choix de p , $\sum_{i=1}^p |C_i| \geq r$ et $\sum_{i=1}^{p-1} |C_i| \leq r - 1$. De plus si nous avons $|C_p| \leq c$ nous pouvons alors conclure que $\sum_{i=1}^p |C_i| \leq r + (c - 1)$.

Observation 4.1. Soit $G = (V, E)$ un graphe et $\mathcal{S} = \{C_1, \dots, C_q\}$ une collection ordonnée de sous-ensembles disjoints de V telle que $|C_j| \leq c$ pour $1 \leq j \leq q$ et un entier $c > 0$. Soit $\mathcal{C} = \{C_1, \dots, C_p\}$ un r -bourrage de \mathcal{S} , alors $\sum_{i=1}^p |C_i| \leq r + (c - 1)$.

Nous concluons cette section en introduisant une opération de collage sur les graphes trivialement parfaits, à savoir la *substitution*, qui nous permettra de simplifier la rédaction de nos preuves.

Définition 4.3 (Substitution). Soit u un sommet de $G = (V, E)$ et soit un graphe $H = (V_H, E_H)$. La *substitution de u par H dans G* , noté $G(u \rightarrow H)$ est le graphe obtenu en remplaçant u par H dans G . Plus formellement :

$$G(u \rightarrow H) = ((V \setminus \{u\}) \cup V_H, E(G \setminus \{u\}) \cup E_H \cup (V_H \times N_G(u))).$$

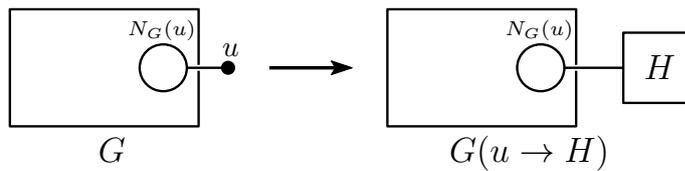


Figure 4.3: Illustration de $G(u \rightarrow H)$, la substitution u par H dans G .

Proposition 4.2. Si $G = (V, E)$ est un graphe trivialement parfait, alors pour tout $u \in V$ tel que $N_G(u)$ est une clique et pour tout graphe trivialement parfait H , le graphe $G(u \rightarrow H)$ est trivialement parfait.

Preuve. Soit $S \subseteq V \setminus \{u\}$ une clique maximale de G contenant $N_G(u)$. Nous appliquons le lemme 4.2 avec S pour obtenir les composantes $\{K_1, \dots, K_r\}$, qui correspondent à des modules tels que $G[S \cup V(K_i)]$ est trivialement parfait pour tout $1 \leq i \leq r$ et $\bigcup_{1 \leq i \leq r} \{N_G(V(K_i))\}$ est une famille imbriquée. Notons que par construction et sans perte de généralité, nous pouvons supposer que $K_1 = \{u\}$. En remplaçant K_1 par H , par le lemme 4.2, il est immédiat que $G(u \rightarrow H)$ est trivialement parfait. \square

4.2 Noyau pour TRIVIAALLY PERFECT EDITION

Nous présentons dans cette section un noyau pour le problème TRIVIAALLY PERFECT EDITION qui contient un nombre quadratique de sommets. Nous commençons en donnant une description informelle du noyau. Pour borner la taille d'une instance réduite, nous utilisons la décomposition en clique universelle qui a été introduite dans la section précédente. Considérons une instance positive $(G = (V, E), k)$ de TRIVIAALLY PERFECT EDITION et F une k -édition de G . Définissons $\mathcal{T} = (T, \mathcal{B})$ la décomposition en clique universelle du graphe trivialement parfait $H = G \Delta F$. Puisque F est de taille au plus k , il existe au plus $2k$ sacs de \mathcal{T} qui contiennent un sommet affecté (rappelons que \mathcal{B} partitionne les sommets de G). Soit A l'ensemble des noeuds associés à ces sacs, et définissons A' comme la clôture par plus petit ancêtre commun de A dans la forêt T (définition 4.6). Ces ensembles sont illustrées dans la figure 4.4. La taille de cet ensemble A' est linéaire en k (lemme 4.11). De plus, les graphes trivialement parfaits sont clos pour l'ajout de vrais jumeaux, nous pouvons donc borner par $O(k)$ la taille de chacun des sacs de \mathcal{B} (qui correspondent à des cliques critiques). Ainsi, les sacs correspondant aux noeuds de A' contiennent au total $O(k^2)$ sommets.

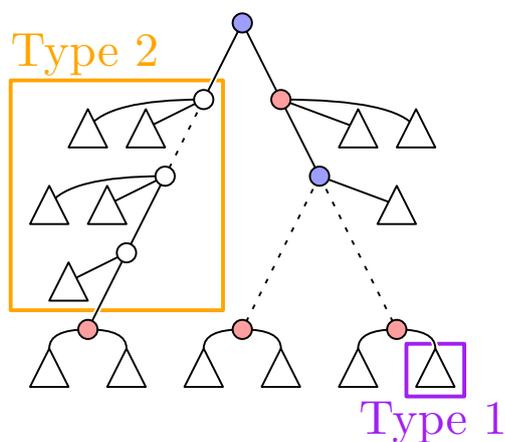


Figure 4.4: Analyse d'une décomposition en clique universelle d'un graphe trivialement parfait connexe. Les noeuds rouges correspondent aux noeuds de A , les noeuds bleus à $A' \setminus A$. Les triangles correspondent à des sous-arbres de la décomposition en clique universelle.

Retirer les sommets de A' de T crée de nouvelles composantes connexes, qui peuvent être adjacentes à zéro, un ou deux sommets de A' dans T . Nous qualifions ces composantes connexes de type 0, 1 ou 2 respectivement. Les composantes connexes de type 0 correspondent à des composantes connexes de H qui ne contiennent pas de sommets affectés, ce sont donc des composantes connexes de G qui sont trivialement parfaites. Elles peuvent être simplement retirées puisque les graphes trivialement parfaits sont clos par union disjointe.

Nous pouvons observer que les composantes connexes de type 1 adjacentes à un même sommet de A' forment un module trivialement parfait dans le graphe G par définition de la décomposition en clique universelle. De plus nous montrerons qu'il est possible de borner la taille des modules trivialement parfaits par $O(k)$. Il y a autant de modules trivialement parfaits constitués de composantes de type 1 qu'il y a de sommets de A' . Nous pouvons conclure que puisque $A' = O(k)$, le nombre de sommets dans toutes les composantes connexes de type 1 est borné par $O(k^2)$.

Il nous reste à borner le nombre de sommets dans les composantes connexes de type 2. Elles sont adjacentes dans T à deux sommets $a_1, a_2 \in A'$, l'un étant l'ancêtre de l'autre, en particulier il en existe $O(k)$. Les sommets de ces composantes connexes de type 2 forment ce que nous appellerons des *peignes*. L'union des sommets d'un peigne induit un graphe trivialement parfait, de plus ils peuvent être partitionnés en un ensemble de cliques critiques et un ensemble de modules trivialement parfaits. Nous montrerons qu'il est possible de borner linéairement en k le nombre de sommets dans chacune des parties d'un peigne.

De cette analyse, nous concluons que le nombre de sommets d'une instance réduite exhaustivement par les règles que nous définissons peut être bornée par $O(k^2)$.

Remarque 4.1. Dans le reste de la section nous supposons que nous avons en entrée une instance $(G = (V, E), k)$ de TRIVIAALLY PERFECT EDITION.

4.2.1 Règles de réduction classiques

Nous commençons par redéfinir les règles portant sur les composantes connexes trivialement parfaites et les cliques critiques. Les graphes trivialement parfaits étant clos par union disjointe et pour l'ajout de jumeaux, ces règles sont donc sûres d'après les [lemmes 3.1](#) et [3.3](#).

Règle 4.1. Soit C une composante connexe de G telle que C est trivialement parfaite. Retirer les sommets de C .

Règle 4.2. Soit $K \subseteq V$ une clique critique de G telle que $|K| > k + 1$. Retirer $|K| - (k + 1)$ sommets de K .

En utilisant une règle de réduction bornant la taille des ensembles indépendants dans les modules trivialement parfaits, Drange et Pilipczuk [54] avaient montré que la taille des modules trivialement parfaits pouvait être bornée par $O(k^2)$. Pour pouvoir borner la taille des modules trivialement parfaits par $O(k)$, nous procédons en deux temps. Nous commençons par définir une règle de réduction qui permet de borner la taille des anti-couplages ([définition 4.4](#)) dans les modules trivialement parfaits. Notons que cette règle généralise celle sur les ensembles indépendants. Néanmoins elle n'est pas suffisante pour borner linéairement en k la taille de tous les modules trivialement parfaits. Afin de borner linéairement en k la taille des modules trivialement parfaits ayant un anti-couplage de taille plus petite que $k + 1$, nous utiliserons une règle de réduction qui opère sur une structure plus complexe, appelée *peigne*, qui induit un graphe trivialement parfait mais pas nécessairement un module.

Définition 4.4 (Anti-couplage). Un *anti-couplage* d'un graphe $G = (V, E)$ est un ensemble de paires $\{u, v\}$ disjointes de G telles que $\{u, v\} \notin E$.

Par un léger abus de notation, nous désignons par $V(D)$ l'ensemble des sommets contenus dans les paires d'un anti-couplage D .

Règle 4.3. Soit $M \subseteq V$ un module trivialement parfait de G . Si $G[M]$ contient un anti-couplage D de taille $(k + 1)$, alors retirer les sommets $M \setminus V(D)$.

Lemme 4.3. La [règle 4.3](#) est sûre.

Preuve. Soit $G' = (V', E')$ le graphe obtenu après l'application de la [règle 4.3](#). Puisque G' est un sous-graphe induit de G et que les graphes trivialement parfaits sont héréditaires, toute k -édition de G est aussi une k -édition de G' .

Supposons que G' admette une k -édition F' et soit $H' = G' \Delta F'$, nous allons construire une k -édition F^* de G à partir de F' . Soit $M' = V(D)$, ce sont les sommets qui n'ont pas été retirés par la [règle 4.3](#). Notons que puisque les paires contenues dans un anti-couplage sont disjointes ([définition 4.4](#)), $|M'| = 2(k + 1)$. De plus, puisque $|F'| \leq k$, au plus $2k$ sommets sont affectés par F' . Il existe donc un sommet u non affecté dans M' .

Affirmation 4.1. $N_{G'}(M')$ est une clique dans H' .

Preuve. Soit $D = \{\{u_i, v_i\} \mid 1 \leq i \leq k + 1\}$ un anti-couplage de taille $(k + 1)$ de M' . Supposons par contradiction que $N_{G'}(M')$ ne soit pas une clique dans H' et prenons $\{x, y\}$ une non-arête de H' avec $x, y \in N_{G'}(M')$. Puisque $|F'| \leq k$ il existe $1 \leq j \leq k + 1$ tel que $\{u_j, v_j\} \notin F'$ et pour tout $z \in V(G') \setminus M'$, $\{u_j, z\}, \{v_j, z\} \notin F'$. Ainsi $\{u_j, x, v_j, y\}$ induit un C_4 dans H' , ce qui est une contradiction. \diamond

Le graphe $H_u = H' \setminus (M' \setminus \{u\})$ est trivialement parfait par hérédité et $N_{H_u}(u) = N_{G'}(M')$. Il en résulte par l'[affirmation 4.1](#) que $N_{H_u}(u)$ est une clique et la [proposition 4.2](#) implique que le graphe $H = H_u(u \rightarrow M)$ est trivialement parfait. Soit F^* l'édition telle que $H = G \Delta F^*$. Puisque u n'est pas affecté par F' et que u est dans M , nous avons que $N_{H_u}(u) = N_G(M)$. Par conséquent, puisque M est un module dans G , nous avons que $N_H(v) = N_G(v)$ pour tout sommet $v \in M$, ce qui implique que $F^* \subseteq F'$ et donc F^* est bien une k -édition de G . \square

4.2.2 Peignes : définition et propriétés structurelles

Nous définissons maintenant la structure principale de notre noyau, à savoir les *peignes*. Notons que cette structure est similaire à celle définie par Drange et Pilipczuk [54] et nous l'avons donc nommée de la même manière. Cependant, les deux structures ne sont pas strictement identiques, en particulier leur voisinage avec le reste du graphe était structuré différemment. En effet, Drange et Pilipczuk [54] définissent les peignes à partir d'un modulateur de sommets, qui est un ensemble de sommets intersectant toutes les obstructions. Nous avons néanmoins choisi d'utiliser le même nom car nous le trouvons adapté pour illustrer la structure.

Définition 4.5 (peigne). Soit $G = (V, E)$ un graphe. Une paire (C, R) de sous-ensembles disjoints de V est un *peigne* de G si :

- $G[C]$ est une clique qui peut être partitionnée en l cliques critiques $\{C_1, \dots, C_l\}$,
- R peut être partitionné en l modules trivialement parfaits non vides $\{R_1, \dots, R_l\}$,
- $N_G(C_i) \cap R = \bigcup_{j=i}^l R_j$ et $N_G(R_i) \cap C = \bigcup_{j=1}^i C_j$ pour $1 \leq i \leq l$,
- il existe deux sous-ensembles (éventuellement vides) de sommets $V_f, V_p \subseteq V(G) \setminus \{C \cup R\}$ tels que:
 - $\forall x \in C, N_G(x) \setminus (C \cup R) = V_p \cup V_f$,
 - $\forall y \in R, N_G(y) \setminus (C \cup R) = V_p$.

Étant donné un peigne (C, R) , C est appelé le *manche* du peigne et R les *dents* du peigne. La [figure 4.5](#) illustre la structure d'un peigne. La *taille* d'un peigne est le nombre de sommets qu'il contient, c'est-à-dire $|C \cup R|$.

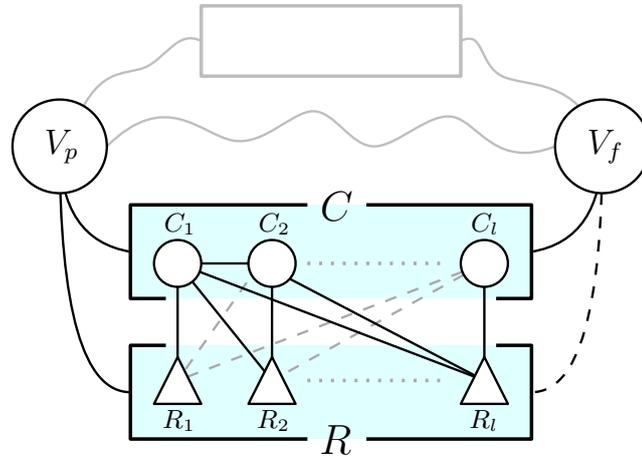


Figure 4.5: Un peigne d'un graphe $G = (V, E)$ avec le manche C et les dents R . Chaque ensemble C_i est une clique critique tandis que chaque ensemble R_i induit un module (éventuellement non connexe) trivialement parfait. Remarquons que les ensembles V_p et V_f peuvent être adjacents à d'autres sommets du graphe G .

Proposition 4.3. *Étant donné un peigne (C, R) d'un graphe $G = (V, E)$, le sous-graphe $G[C \cup R]$ est trivialement parfait. De plus, les ensembles V_p et V_f , et les partitions ordonnées $\{C_1, \dots, C_l\}$ de C et $\{R_1, \dots, R_l\}$ de R sont déterminés de manière unique.*

Preuve. Commençons par observer que $V_p = N_G(R) \setminus C$ et $V_f = N_G(C) \setminus (R \cup V_p)$, donc les deux ensembles sont déterminés de manière unique. Puisque (C, R) est un peigne, C est une union de l cliques critiques qui sont totalement ordonnées par l'inclusion de leurs voisinages fermés. En d'autres termes $N_G[C_i] \subsetneq \dots \subsetneq N_G[C_1]$, en effet pour $1 \leq i \leq l$, $N_G[C_i] = (\bigcup_{j=i}^l R_j) \cup V_p \cup V_f$. Enfin, pour tout $1 \leq i < l$, $R_i = N(C_i) \setminus N(C_{i+1})$ et $R_l = R \setminus \bigcup_{i=1}^{l-1} R_i$ par définition d'un

peigne. Par conséquent, $G[C \cup R]$ admet une décomposition en clique universelle, où chaque $G[R_1], \dots, G[R_{l-1}]$ admet une décomposition en clique universelle, et les noeuds correspondant à C_i , $1 \leq i < l-1$, sont les parents du noeud C_{i+1} et de la racine de la décomposition de $G[R_i]$. Le noeud correspondant à C_{l-1} est connecté à la décomposition de $G[R_{l-1}]$ et $G[R_l \cup C_l]$, qui admet une décomposition en clique universelle puisque $G[R_l]$ est trivialement parfait et C_l est une clique universelle de $G[R_l \cup C_l]$. \square

Remarque 4.2. Dans le reste du chapitre, nous supposons que tout peigne (C, R) est donné avec les partitions ordonnées $\{C_1, \dots, C_l\}$ de C et $\{R_1, \dots, R_l\}$ de R .

Nous faisons maintenant des observations sur les peignes pour faciliter la compréhension de la [définition 4.5](#) et en particulier sa relation avec les modules trivialement parfaits. Étant donné un graphe trivialement parfait $G = (V, E)$ et sa décomposition en clique universelle $\mathcal{T}_G = (T, \mathcal{B})$, nous pouvons construire un peigne (C, R) de G en prenant simplement un chemin P d'un noeud v_1 de T à l'un de ses descendants v_l . Le manche C correspond aux sommets dans les sacs de ce chemin, les dents R correspondent aux sacs des sous-arbres enracinés avec les enfants (qui ne sont pas sur P) de n'importe quel noeud sur le chemin P . Nous pouvons observer que dans ce cas, V_p correspond aux sommets des sacs du chemin allant du parent de v_1 à la racine de T et que V_f est vide.

En particulier, l'ensemble des sommets d'un graphe trivialement parfait connexe peut être partitionné en un peigne (C, R) en prenant un chemin de la racine de sa décomposition en clique universelle à l'une de ses feuilles. Cela signifie que lorsque $V_p = V_f = \emptyset$, le peigne correspond à un graphe trivialement parfait connexe. De même, si seul l'ensemble V_f est vide, la [définition 4.5](#) correspond à un module trivialement parfait connexe puisque $N_G(R) \setminus C = N_G(C) \setminus R = V_p$.

L'observation suivante est immédiate par la définition d'un peigne et est vérifiée que les ensembles V_p et V_f soient vides ou non.

Observation 4.2. L'ensemble de sommets C (resp. R) est un module de $G \setminus R$ (resp. $G \setminus C$).

Nous allons montrer que la taille des peignes peut être réduite à $O(k)$ sommets. Nous nous concentrons d'abord sur les peignes ayant un grand nombre de sommets dans leur manche, ce qui nous permettra de borner la taille des modules trivialement parfaits ayant un petit anti-couplage à $O(k)$ sommets ([lemme 4.6](#)). Nous nous intéresserons ensuite aux peignes ayant un grand nombre de sommets dans leurs dents afin de borner la taille d'un peigne à $O(k)$ sommets ([lemme 4.8](#)).

4.2.3 Réduction des manches

La [règle 4.4](#) qui est présentée dans cette section ne conserve que deux ensembles \mathcal{C}_a et \mathcal{C}_b contenant un nombre linéaire en k de sommets au début et à la fin du manche, ce qui permet de borner la taille de ce dernier linéairement en k . Les deux ensembles \mathcal{C}_a et \mathcal{C}_b sont suffisamment grands pour garantir l'existence de deux sommets qui ne sont pas affectés par une k -édition du graphe. Nous utilisons ces sommets pour prouver qu'il existe une k -édition du graphe qui n'affecte aucun sommet du manche situé entre \mathcal{C}_a et \mathcal{C}_b , et ce qui implique donc la sûreté de la règle.

Règle 4.4. Soit (C, R) un peigne de G tel qu'il existe deux ensembles disjoints \mathcal{C}_a et \mathcal{C}_b où :

- \mathcal{C}_a est un $(2k + 1)$ -bourrage de $\{C_1, \dots, C_l\}$,
- \mathcal{C}_b est un $(2k + 1)$ -bourrage de $\{C_l, C_{l-1}, \dots, C_1\}$.

Retirer les sommets de $C' = C \setminus (\mathcal{C}_a \cup \mathcal{C}_b)$.

Lemme 4.4. La [règle 4.4](#) est sûre.

Preuve. Soit $G' = (V', E')$ le graphe obtenu après l'application de la [règle 4.4](#). Puisque G' est un sous-graphe induit de G et que les graphes trivialement parfaits sont héréditaires, toute k -édition de G est aussi une k -édition de G' .

Supposons que G' admette une k -édition F' et soit $H' = G' \triangle F'$, nous allons construire une k -édition F^* de G à partir de F' . Puisque F' affecte au plus $2k$ sommets et que \mathcal{C}_a et \mathcal{C}_b contiennent respectivement au moins $2k + 1$ sommets, il existe des sommets non affectés c_a et c_b contenus dans

C_a et C_b respectivement. Soit C_a et C_b les cliques critiques de C contenant c_a et c_b , $1 \leq a < b \leq l$. De plus, nous définissons les ensembles suivants qui sont illustrés dans la [figure 4.6](#) :

$$\begin{aligned} C_o &= C_{a+1} \cup \dots \cup C_{b-1} \\ R_o &= R_a \cup \dots \cup R_{b-1} \\ C_< &= C_1 \cup \dots \cup C_a \\ C_> &= C_b \cup \dots \cup C_l \\ R_> &= R_b \cup \dots \cup R_l \end{aligned}$$

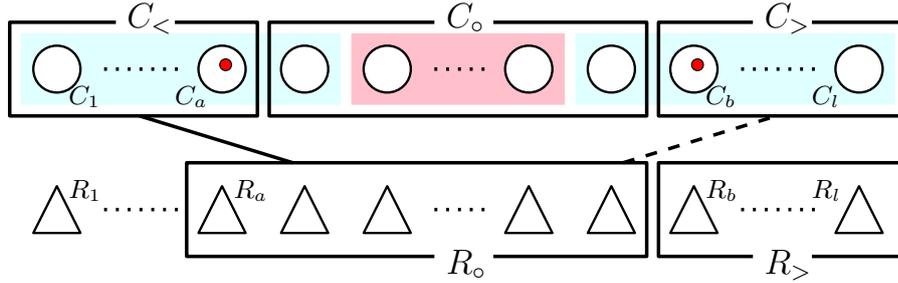


Figure 4.6: Illustration du peigne et des ensembles utilisés dans la preuve du [lemme 4.4](#). Les ronds sont les cliques critiques du manche et les triangles sont les dents. Les sommets rouges correspondent à c_a et c_b , les rectangles bleus clairs correspondent aux ensembles C_a et C_b et le rectangle rose correspond à C' , qui est supprimé par la [règle 4.4](#).

Enfin, définissons $G_o = G \setminus C_o$ et $H_o = H' \setminus C_o$ et observons que H_o est trivialement parfait et que $C' \subseteq C_o$. Soit $F_o \subseteq F'$ la k -édition telle que $H_o = G_o \triangle F_o$ et soit S_o une clique maximale de H_o contenant c_a et c_b . Puisque c_a et c_b ne sont pas affectés, S_o est inclus dans $N_{G_o}(\{c_a, c_b\}) = C \cup V_p \cup V_f \cup R_>$. Nous utilisons le [lemme 4.2](#) avec S_o pour obtenir un ensemble de composantes connexes $\{K_1, \dots, K_r\}$ de $H_o \setminus S_o$ tel que K_1, \dots, K_r sont des modules dans H_o dont les voisinages (éventuellement vides) dans S_o forment une famille imbriquée. Nous commençons par modifier F_o pour obtenir une k -édition de G_o où les sommets de R_o sont affectés uniformément.

Affirmation 4.2. Il existe une k -édition F^* de G_o telle que dans $H^* = G_o \triangle F^*$, R_o est un module et $H^*[R_o] = G[R_o]$.

Preuve. Nous commençons par plusieurs observations qui nous seront utiles. Tout d'abord, R_o est un module dans G_o puisque $R \supset R_o$ est un module dans $G \setminus C$ ([observation 4.2](#)) et que les sommets de R_o sont adjacents à $C_<$ et non-adjacents à $C_>$. Ensuite, puisque les sommets d'une composante K_i , forment un module dans H_o , et que c_a et c_b ne sont pas affectés par F_o , nous avons $N_{H_o}(V(K_i)) \cap \{c_a, c_b\} = N_{G_o}(V(K_i)) \cap \{c_a, c_b\}$. En d'autres termes, les sommets d'une même composante K_i doivent avoir la même adjacence avec $\{c_a, c_b\}$ dans G_o et dans H_o . De même, aucun sommet $v \in R_o$ n'appartient à S_o puisque $N_{G_o}(v) \cap \{c_b\} = \emptyset$. De plus, les seuls sommets de G_o qui sont adjacents à c_a mais pas à c_b sont exactement ceux de R_o . Par conséquent, pour tout sommet $v_o \in R_o$, $N_{H_o}(v_o) \subseteq S_o \cup R_o$.

Supposons maintenant que R_o n'est pas un module dans H_o et que $v_o \in R_o$ est un sommet contenu dans le plus petit nombre de paires de F_o avec l'autre élément de la paire dans S_o . Considérons le graphe $H'_o = H_o \setminus (R_o \setminus \{v_o\})$, qui est trivialement parfait par hérédité. Puisque $N_{H_o}(v_o) \subseteq S_o \cup R_o$, il s'ensuit que $N_{H'_o}(v_o) \subseteq S_o$ est une clique. Par conséquent, la [proposition 4.2](#) implique que le graphe $H^* = H'_o(v_o \rightarrow G[R_o])$ est trivialement parfait. Soit F^* l'édition telle que $H^* = G_o \triangle F^*$. Par le choix de v_o nous avons que $|F^*| \leq |F_o|$. Cela implique que F^* est une k -édition respectant les propriétés voulues, ce qui conclut la preuve de l'[affirmation 4.2](#). \diamond

Nous considérons désormais $H^* = G_o \triangle F^*$ où F^* est la k -édition de l'[affirmation 4.2](#). Notons que les composantes connexes de $H_o \setminus S_o$ et $H^* \setminus S_o$ peuvent être différentes. Par un léger abus de notation, nous définissons toujours ces composantes comme K_1, \dots, K_r . Rappelons que K_1, \dots, K_r sont des modules dans H^* dont les voisinages avec S_o forment une famille imbriquée.

Affirmation 4.3. Le graphe $H = G\Delta F^*$ est trivialement parfait.

Preuve. Le graphe H correspond à H^* où les sommets de C_o ont été ajoutés avec le même voisinage que dans G . Observons d'abord que $S = S_o \cup C_o$ est une clique maximale dans H . En effet, C_o est une clique par définition et $S_o \subseteq (C \cup V_p \cup V_f \cup R_{>}) \subseteq N_H(C_o) = N_G(C_o)$ (rappelons que C est adjacent à $V_p \cup V_f$ par la [définition 4.5](#) et que les sommets de C_o sont adjacents à tous les sommets de $R_{>}$). Par conséquent, les composantes K_1, \dots, K_r définies dans $H^* \setminus S_o$ sont les mêmes dans $H \setminus S$ et leurs voisinages forment une famille imbriquée incluse dans S_o . Nous divisons les composantes K_1, \dots, K_r en trois types de composantes en fonction de leurs adjacences avec c_a et c_b , à savoir :

1. α -composantes qui sont non-adjacentes à c_a et c_b ,
2. β -composantes qui sont adjacentes à c_a mais pas c_b ,
3. δ -composantes qui sont adjacentes à c_a et c_b .

Dans ce qui suit, K_α, K_β et K_δ désignent respectivement une α, β et δ -composante arbitraire. Notons que les inclusions $N_{H^*}(V(K_\alpha)) \subseteq N_{H^*}(V(K_\beta)) \subseteq N_{H^*}(V(K_\delta)) \subseteq S_o$ sont vérifiées par construction. Rappelons que puisque c_a et c_b ne sont pas affectés par F^* , $N_G(V(K_i)) \cap \{c_a, c_b\} = N_H(V(K_i)) \cap \{c_a, c_b\}$ pour tout K_i . Nous affirmons que $\{N_H(V(K_i)) \mid 1 \leq i \leq r\}$ est une famille imbriquée. Notons que le [lemme 4.2](#) impliquera la validité de l'[affirmation 4.3](#) puisque S est une clique maximale dans H . Rappelons que les voisinages des sommets de C_o sont identiques dans G et H . De plus, $N_H[c_b] \subseteq N_H[C_o] \subseteq N_H[c_a]$ est vérifié car ces sommets ne sont pas affectés par F^* . Ainsi les α -composantes (resp. les δ -composantes) sont non-adjacentes (resp. adjacentes) à tous les sommets de C_o dans H . Cela signifie que les voisinages des α et δ -composantes forment une famille imbriquée incluse dans S . De plus, nous pouvons observer que les sommets des β -composantes sont exactement ceux de R_o puisqu'ils sont les seuls à être adjacents à c_a mais pas à c_b dans G . Par conséquent, dans H , nous avons encore :

$$N_H(V(K_\alpha)) \subseteq N_H(V(K_\beta)) \subseteq N_H(V(K_\delta))$$

Il reste à prouver que les voisinages des β -composantes forment une famille imbriquée incluse dans S . Sans perte de généralité soit $\{K_1, \dots, K_p\}$, $1 \leq p \leq r$ les β -composantes. Par définition d'un peigne, les β -composantes (qui sont aussi R_o) peuvent être ordonnées par rapport à l'inclusion de leur voisinage dans $G[C_o]$. Nous pouvons supposer que l'ordre est $N_{G[C_o]}(V(K_1)) \subseteq \dots \subseteq N_{G[C_o]}(V(K_p))$. De plus, nous pouvons observer que pour toute β -composante K_i nous avons $N_{G[C_o]}(V(K_i)) = N_{H[C_o]}(V(K_i))$, $1 \leq i \leq p$. Puisque R_o est un module dans H^* par l'[affirmation 4.2](#) et que les sommets des β -composantes sont exactement ceux de R_o , il s'ensuit que les voisinages des β -composantes forment une famille imbriquée. Par conséquent, $\{N_H(V(K_i)) \mid 1 \leq i \leq r\}$ est une famille imbriquée et H est un graphe trivialement parfait par le [lemme 4.2](#). \diamond

Par l'[affirmation 4.3](#) le graphe $H = G\Delta F^*$ est trivialement parfait et puisque $|F^*| \leq k$, il en résulte que F^* est une k -édition de G , ce qui conclut la preuve du [lemme 4.4](#). \square

La [règle 4.4](#) que nous venons de définir nous permet de borner linéairement en k la taille du manche des peignes.

Observation 4.3. Supposons que l'instance (G, k) soit réduite par les [règles 4.2](#) et [4.4](#). Pour tout peigne (C, R) de G , nous avons $|C| \leq 6k + 2$.

Preuve. Puisque G est réduit par la [règle 4.2](#), chaque clique critique C_i du manche contient au plus $k + 1$ sommets. Par l'[observation 4.1](#), tout $(2k + 1)$ -bourrage de $\{C_1, \dots, C_l\}$ (resp. $\{C_l, \dots, C_1\}$) contient au plus $3k + 1$ sommets. Ainsi $|C| \leq 6k + 2$, sinon on pourrait trouver deux $(2k + 1)$ -bourrages *disjoints* de $\{C_1, \dots, C_l\}$ et de $\{C_l, \dots, C_1\}$ et la [règle 4.4](#) pourrait être appliquée. \square

Nous pouvons maintenant montrer comment réduire la taille de tout module trivialement parfait. Nous avons besoin du résultat combinatoire qui suit pour obtenir la borne désirée.

Lemme 4.5. Soit $G = (V, E)$ un graphe trivialement parfait connexe et α la taille d'un anti-couplage maximum de G . Il existe un peigne (C, R) de G tel que $V = C \cup R$ et $|R| \leq 4\alpha$. De plus, un tel peigne peut être calculé en temps polynomial.

Preuve. Nous faisons une preuve constructive qui impliquera directement la dernière partie du lemme. Rappelons que tout graphe trivialement parfait contient un sommet universel et notons $U_1 \subseteq V(G)$ la clique universelle de G . Soit $R_1^1, \dots, R_{p_1}^1$ les composantes connexes de $G \setminus U_1$ (illustrées dans la figure 4.7). Puisque G ne contient pas d'anti-couplage de taille $(\alpha + 1)$, il existe au plus un ensemble R_i^1 , $1 \leq i \leq p_1$ tel que $|R_i^1| > \alpha$ (car il n'y a pas d'arête entre R_i^1 et R_j^1 , $1 \leq i < j \leq p_1$).

Supposons sans perte de généralité que $|R_1^1| > \alpha$. Nous ajoutons tous les sommets de $\cup_{i=2}^{p_1} R_i^1$ à un ensemble $R_{<}$ et nous répétons ce processus sur $G[R_1^1]$ jusqu'à ce que chaque composante connexe soit plus petite que α . Plus formellement, à l'étape $j > 1$, pour le graphe trivialement parfait $G_j = G[R_1^{j-1}]$, soit U_j sa clique universelle et soit $R_1^j, \dots, R_{p_j}^j$ les composantes connexes de $G_j \setminus U_j$. Soit R_1^j celle dont la taille est supérieure à α si elle existe, sinon nous arrêtons le processus et l est la dernière étape. En particulier, $|R_i^l| \leq \alpha$, $1 \leq i \leq p_l$. Soit $R_{<} = \cup_{j=1}^{l-1} \cup_{i=2}^{p_j} R_i^j$ et $R_{>} = R_1^l \cup \dots \cup R_{p_l}^l$ (illustrés dans la figure 4.7).

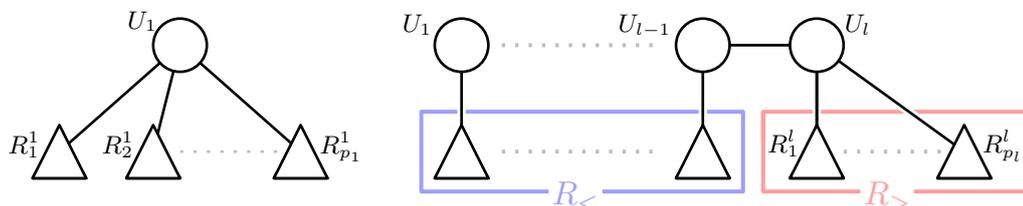


Figure 4.7: Illustration des structures obtenues à la première étape (à gauche) et à la dernière étape (à droite) par le processus décrit dans la preuve du lemme 4.5.

Rappelons que $|R_1^{l-1}| > \alpha$ par construction. Cela implique que $|R_{<}| \leq \alpha$ car sinon $G[R_{<} \cup R_1^{l-1}]$ contiendrait un anti-couplage de taille $(\alpha + 1)$. Nous affirmons que $|R_{>}| \leq 3\alpha$. Pour prouver cette affirmation, considérons le $(\alpha + 1)$ -bourrage $\{R_1^l, \dots, R_q^l\}$ de $\{R_1^l, \dots, R_{p_l}^l\}$ et définissons $R' = \cup_{i=1}^q R_i^l$. Soit $R'' = R_{>} \setminus R'$. Rappelons que l est la dernière étape du processus et que $|R_i^l| \leq \alpha$ pour $1 \leq i \leq p_l$. Par conséquent, l'observation 4.1 implique $|R'| \leq 2\alpha$. Ainsi, nous avons que $|R''| \leq \alpha$ car sinon $G[R' \cup R'']$ contiendrait un anti-couplage de taille $(\alpha + 1)$, ce qui est une contradiction. Par conséquent $|R_{>}| = |R'| + |R''| \leq 3\alpha$.

Pour construire un peigne pour G , nous considérons l'ensemble $C = \{U_1, \dots, U_l\}$ comme le manche (rappelons que U_1 est la clique universelle de G et que U_j désigne la clique universelle de $G[R_1^{j-1}]$ à chaque étape $1 < j \leq l$). De plus, pour chaque $1 \leq j < l$, la dent R_j est égale à $R_2^j \cup \dots \cup R_{p_j}^j$, la dernière dent R_l étant $R_{>}$. Par construction $(C, R = \cup_{j=1}^l R_j)$ est un peigne de G tel que $|R| = |R_{<}| + |R_{>}| \leq 4\alpha$, ce qui conclut la preuve. \square

Lemme 4.6. Supposons que l'instance (G, k) est réduite par les règles 4.1 à 4.4 et soit M un module trivialement parfait de G . Alors M contient au plus $11k + 2$ sommets.

Preuve. Observons que si M contient un anti-couplage de taille plus grande que k , alors il est réduit par la règle 4.3 et contient $2k + 2$ sommets. Supposons donc que M ne contienne pas d'anti-couplage de taille $(k + 1)$. Supposons d'abord que $G[M]$ est connexe. Soit (C, R) un peigne obtenu avec le lemme 4.5, tel que $C \cup R = M$ et $|R| \leq 4k$. Par l'observation 4.3 nous avons que $|C| \leq 6k + 2$, donc $|M| \leq |C| + |R| \leq 10k + 2$.

Pour conclure, il reste à traiter le cas où $G[M]$ est non connexe. Soit $\{M_1, \dots, M_p\}$ les composantes connexes de $G[M]$. Puisque M ne contient pas d'anti-couplage de taille $(k + 1)$, au plus une de ses composantes connexes a une taille supérieure à k , supposons qu'il s'agit de M_1 si elle existe. Soit \mathcal{C} le $(k + 1)$ -bourrage de $\{M_1, \dots, M_p\}$. Comme $|M_1| \leq 10k + 2$ et $|M_i| \leq k$ pour $2 \leq i \leq p$, nous avons que $|\mathcal{C}| \leq 10k + 2$. De plus, puisque M ne contient pas d'anti-couplage de taille $(k + 1)$, cela implique que $|M \setminus \mathcal{C}| \leq k$ et donc $|M| \leq 11k + 2$. \square

4.2.4 Réduction des dents

Nous nous intéressons maintenant au cas où un peigne donné contient beaucoup de sommets dans ses dents. Les arguments sont similaires à ceux utilisés pour la preuve de sûreté de la règle 4.4. La principale différence réside dans le fait que les informations fournies par les sommets non affectés diffèrent lorsqu'ils sont contenus dans les dents plutôt que dans le manche.

Règle 4.5. Soit (C, R) un peigne de G tel qu'il existe trois ensembles disjoints $\mathcal{R}_a, \mathcal{R}_b$ and \mathcal{R}_c où :

- \mathcal{R}_a est un $(2k + 1)$ -bourrage de $\{R_1, \dots, R_l\}$,
- $\mathcal{R}_c = \{R_l, \dots, R_q\}$ est un $(2k + 1)$ -bourrage de $\{R_l, \dots, R_1\}$,
- \mathcal{R}_b est un $(2k + 1)$ -bourrage de $\{R_{q-1}, \dots, R_1\}$.

Retirer les sommets de $R' = R \setminus (\mathcal{R}_a \cup \mathcal{R}_b \cup \mathcal{R}_c)$.

Lemme 4.7. La règle 4.5 est sûre.

Preuve. Soit $G' = (V', E')$ le graphe obtenu après l'application de la règle 4.5. Puisque G' est un sous-graphe induit de G et que les graphes trivialement parfaits sont héréditaires, toute k -édition de G est aussi une k -édition de G' .

Supposons que G' admette une k -édition F' et soit $H' = G' \Delta F'$, nous allons construire une k -édition F^* de G à partir de F' . Soit r_a, r_b et r_c des sommets non affectés dans $\mathcal{R}_a, \mathcal{R}_b$ et \mathcal{R}_c respectivement. Soit R_a, R_b et $R_c, 1 \leq a < b < c \leq l$, les dents de R contenant r_a, r_b et r_b , respectivement (ces ensembles sont bien définis puisque $\mathcal{R}_a, \mathcal{R}_b$ et \mathcal{R}_c sont disjoints). De plus, comme r_a, r_b et r_c ne sont pas affectés par F' , leurs voisinages sont égaux dans G' et H' et donc :

$$(N_{H'}(r_a) \setminus R_a) \subseteq (N_{H'}(r_b) \setminus R_b) \subseteq (N_{H'}(r_c) \setminus R_c)$$

Affirmation 4.4. L'ensemble $N_{H'}(r_b) \setminus R_b$ est une clique dans H' .

Preuve. Supposons par contradiction que $N_{H'}(r_b) \setminus R_b$ contienne des sommets u et v non adjacents. Rappelons qu'il n'y a pas d'arêtes entre R_b et R_c dans G . Ainsi, puisque $(N_{H'}(r_b) \setminus R_b) \subseteq (N_{H'}(r_c) \setminus R_c)$ les sommets $\{r_b, u, v, r_c\}$ induisent un C_4 dans H' , une contradiction. \diamond

Nous définissons les ensembles suivants qui sont illustrés dans la figure 4.8 :

$$\begin{aligned} R_o &= R_{a+1} \cup \dots \cup R_{b-1} \\ C_o &= C_{a+1} \cup \dots \cup C_b \\ C_< &= C_1 \cup \dots \cup C_a \\ R_< &= R_1 \cup \dots \cup R_a \\ R_> &= R_b \cup \dots \cup R_l \end{aligned}$$

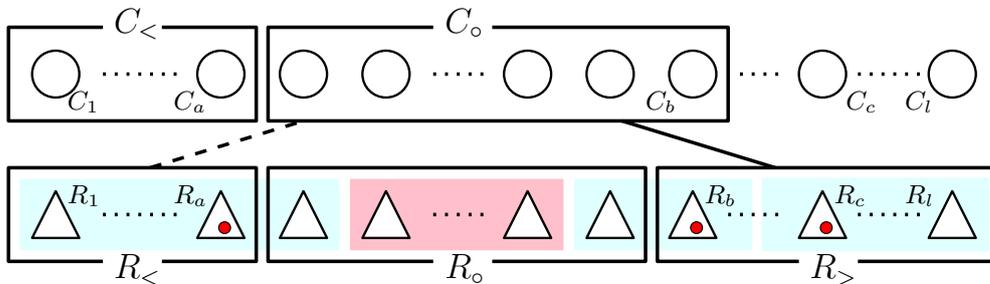


Figure 4.8: Illustration du peigne et des ensembles utilisés dans la preuve du lemme 4.7. Les ronds sont les cliques critiques du manche et les triangles sont les dents. Les sommets rouges correspondent à r_a, r_b et r_c , les rectangles bleus clairs correspondent aux ensembles $\mathcal{R}_a, \mathcal{R}_b$ et \mathcal{R}_c et le rectangle rose correspond à R' , qui est supprimé par la règle 4.5.

De plus nous définissons $G_\circ = G \setminus R_\circ$ et $H_\circ = H' \setminus R_\circ$. Nous pouvons observer que H_\circ est trivialement parfait et que $R' \subseteq R_\circ$. Soit $F_\circ \subseteq F'$ la k -édition telle que $H_\circ = G_\circ \triangle F_\circ$. Nous commençons par modifier F_\circ pour obtenir une k -édition de G_\circ où les sommets de C_\circ sont affectés uniformément.

Affirmation 4.5. Il existe une k -édition F^* de G_\circ telle que dans $H^* = G_\circ \triangle F^*$, C_\circ est un module clique.

Preuve. Commençons par remarquer que C_\circ est une clique critique dans G_\circ puisque $C \supset C_\circ$ est un module dans $G \setminus R$ ([observation 4.2](#)) et les sommets de C_\circ ne sont pas adjacents aux sommets de $R_{<}$ et sont adjacents aux sommets de $R_{>}$.

Supposons maintenant que C_\circ n'est pas un module clique dans H_\circ et soit $v_\circ \in C_\circ$ un sommet contenu dans le plus petit nombre de paires de F_\circ . Considérons le graphe $H'_\circ = H_\circ \setminus (C_\circ \setminus \{v_\circ\})$, qui est trivialement parfait par hérédité, et considérons H^* le graphe obtenu à partir de H'_\circ en ajoutant les sommets de $C_\circ \setminus \{v_\circ\}$ en tant que vrais jumeaux de v_\circ . Soit F^* l'édition telle que $H^* = G_\circ \triangle F^*$. Le graphe H^* est trivialement parfait car la classe des graphes trivialement parfaits est close pour l'ajout de vrais jumeaux. La construction implique que C_\circ est un module clique dans H^* et par le choix de v_\circ , $|F^*| \leq |F_\circ|$, ce qui conclut la preuve. \diamond

Nous considérons désormais $H^* = G_\circ \triangle F^*$ où F^* est la k -édition de l'[affirmation 4.5](#). Nous montrons maintenant que les sommets de R_\circ peuvent être ajoutés à H^* en garantissant qu'il reste trivialement parfait.

Affirmation 4.6. Le graphe $H = G \triangle F^*$ est trivialement parfait.

Preuve. Nous commençons par supprimer les sommets contenus dans $R_b \setminus \{r_b\}$ de H^* , cela nous permet de mieux contrôler le voisinage de r_b et de faciliter la preuve. Soit $H_b^* = H^* \setminus (R_b \setminus \{r_b\})$, qui est trivialement parfait par hérédité. Soit S une clique maximale de H_b^* contenant r_b . Par l'[affirmation 4.4](#), $N_{H_b^*}(r_b)$ est une clique et puisque r_b n'est pas affecté par F^* nous avons que $S = N_{H_b^*}[r_b] = C_{<} \cup C_\circ \cup V_p \cup \{r_b\}$. Nous utilisons le [lemme 4.2](#) avec S pour obtenir un ensemble de composantes connexes $\{K_1, \dots, K_r\}$ de $H_b^* \setminus S$ tel que K_1, \dots, K_r sont des modules dans H_b^* dont les voisinages dans S forment une famille imbriquée. Nous divisons les composantes K_1, \dots, K_r en deux types : K_i est une α -composante si $N_{H_b^*}(V(K_i)) \subseteq (N_{H_b^*}(r_a) \cap S)$ et une β -composante dans le cas contraire. Puisque $N_{H_b^*}(r_a) \cap S = V_p \cup C_{<}$, nous avons que, pour toute α -composante K_α , $N_{H_b^*}(V(K_\alpha)) \subseteq V_p \cup C_{<}$. De plus, puisque $S = N_{H_b^*}[r_b]$ et que C_\circ est un module clique dans H_b^* par l'[affirmation 4.5](#), chaque β -composante K_β vérifie $N_{H_b^*}(V(K_\beta)) = V_p \cup C_{<} \cup C_\circ = S \setminus \{r_b\}$.

Observons maintenant que $(V_p \cup C_{<}) \subseteq N_G(R_\circ) \subseteq S \setminus \{r_b\}$. En d'autres termes, le voisinage de toute dent de R_\circ contient le voisinage de toute α -composante et est contenu dans le voisinage de toute β -composante. De plus, les voisinages des dents de R_\circ forment une famille imbriquée dans G par définition d'un peigne. Nous en concluons que les sommets de R_\circ peuvent être ajoutés à H_b^* avec le même voisinage que celui qu'ils ont dans G et que le graphe H_b qui en résulte est trivialement parfait. Il reste à rajouter les sommets de R_b dans le graphe. Par l'[affirmation 4.4](#) et la [proposition 4.2](#), le graphe $H = H_b(r_b \rightarrow G[R_b])$ est trivialement parfait. \diamond

Par l'[affirmation 4.6](#) le graphe $H = G \triangle F^*$ est trivialement parfait et puisque $|F^*| \leq k$, il en résulte que F^* est une k -édition de G , ce qui conclut la preuve du [lemme 4.7](#). \square

Lemme 4.8. *Supposons que l'instance (G, k) est réduite par les règles 4.1 à 4.5. Soit (C, R) un peigne de G , alors $|C \cup R| \leq 45k + 8$.*

Preuve. Tout d'abord, notons que $|C| \leq 6k + 2$ grâce à l'[observation 4.3](#). Nous procédons de la même manière pour borner la taille de R . Puisque les dents d'un peigne sont des modules trivialement parfaits, le [lemme 4.6](#) garantit que $|R_i| \leq 11k + 2$, $1 \leq i \leq l$. Nous en déduisons, par l'[observation 4.1](#), que tout $(2k + 1)$ -bourrage de $\{R_1, \dots, R_l\}$ contient au plus $13k + 2$ sommets. Par conséquent $|R| \leq 39k + 6$, sinon on pourrait trouver trois $(2k + 1)$ -bourrages disjoints de R qui répondent aux exigences de la [règle 4.5](#). Nous concluons que $|C \cup R| \leq 45k + 8$. \square

4.2.5 Réduction exhaustive du graphe

Nous montrons dans cette section qu'une instance (G, k) peut être réduite en temps polynomial de telle manière qu'aucune des règles 4.1 à 4.5 ne peuvent être appliquées (lemme 4.10). La procédure pour réduire une instance commence par l'application exhaustive des règles 4.1 et 4.2, ensuite les modules sont réduits avec les règles 4.3 et 4.4 et enfin les peignes qui n'ont pas été encore réduits le sont avec les règles 4.4 et 4.5.

La plus grande difficulté réside en l'énumération de tous les peignes qui doivent être réduits. Nous commençons par décrire un algorithme qui énumère tous les *peignes critiques* en temps polynomial. Un peigne (C, R) est dit *critique* si $R \cup C \cup V_f$ n'est pas un module trivialement parfait et si le peigne est maximal pour l'inclusion, c'est-à-dire qu'aucun autre peigne (C', R') ne satisfait $C \subseteq C'$ et $R \subseteq R'$ avec au moins une des inclusions étant stricte. Nous verrons notamment qu'il est suffisant de réduire tous les peignes critiques pour réduire exhaustivement tous les peignes d'un graphe qui ne sont pas inclus dans des modules trivialement parfaits. Dans le cas où $RUCUV_f$ est un module trivialement parfait, le peigne sera réduit en réduisant tous les modules.

Algorithme 2 : Énumération des peignes critiques

Entrée : Un graphe $G = (V, E)$

Sortie : Une liste contenant tous les peignes critiques de G

- 1 **pour chaque** paire de cliques critiques C', C'' telles que $N[C''] \subsetneq N[C']$ **faire**
 - 2 $R(C', C'') = N[C'] \setminus N[C''];$
 - 3 **si** $N(R(C', C'')) \subsetneq N(C'')$ et $R(C', C'')$ est un module trivialement parfait non vide **alors**
 - 4 définir $C' \prec C'';$
 - 5 **pour chaque** clique critique C' ayant au moins deux prédécesseurs avec la relation \prec **faire**
 - 6 supprimer toutes les relations $C'' \prec C';$
 - 7 **pour chaque** chemin $C_1 \prec \dots \prec C_{l-1} \prec C_l$ **faire**
 - 8 définir $R_i = N[C_i] \setminus N[C_{i+1}], \forall 1 \leq i < l;$
 - 9 définir $X = N(C_l) \setminus N(R_{l-1});$
 - 10 définir R_l comme l'union des sommets des composantes connexes de $G[X]$ induisant un module trivialement parfait dont le voisinage est $C_l \cup N(R_{l-1});$
 - 11 **si** (C, R) est un peigne avec les partitions $\{C_1, \dots, C_l\}$ et $\{R_1, \dots, R_l\}$ **alors**
 - 12 ajouter (C, R) à la liste des peignes;
-

Lemme 4.9. *L'algorithme 2 énumère un ensemble de peignes du graphe d'entrée, incluant tous les peignes critiques, en temps polynomial.*

Preuve. Intuitivement, nous visons à construire une relation binaire \prec sur l'ensemble des cliques critiques de G , où $C' \prec C''$ correspond au fait qu'il existe un peigne de G tel que la première clique C' est le parent de la seconde clique C'' dans la décomposition en clique universelle du peigne. Nous rappelons que les sacs d'une décomposition en clique universelle d'un graphe trivialement parfait correspondent aux cliques critiques de ce dernier.

Les conditions nécessaires pour que C' soit le parent de C'' dans la décomposition universelle d'un certain peigne de G sont que $N[C''] \subsetneq N[C']$, que $R(C', C'') = N[C'] \setminus N[C'']$ soit un module trivialement parfait non vide de G , et que $N(R(C', C'')) \subseteq N(C'')$. Désignons cette relation, construite par la première boucle de l'algorithme 2, par $C' \prec C''$. Nous disons que C' précède C'' .

Nous montrons que si C^a, C^b et C^c sont trois cliques critiques distinctes telles que $C^a \prec C^c$ et $C^b \prec C^c$, il n'existe pas de peigne de G dans lequel C^a précède C^c sur le manche. Par contradiction, supposons l'existence d'un tel peigne (C, R) où C est partitionné en cliques critiques $\{C_1, \dots, C_i = C^a, C_{i+1} = C^c, \dots, C_p\}$ et R est partitionné en modules trivialement parfaits non vides $\{R_1, \dots, R_i = R(C^a, C^c), R_{i+1}, \dots, R_p\}$, avec les ensembles V_p et V_f définis comme dans la définition 4.5. Si C^b intersecte une dent R_j du peigne (C, R) pour un $1 \leq j \leq p$, alors R_j est adjacent à $V_f \subseteq N[C^c] \subseteq N[C^b]$, ce qui est une contradiction. Si C^b intersecte V_f , alors V_f est adjacent à $R_{i+1} \subseteq N(C^c) \subseteq N[C^b]$, ce qui est une contradiction.

Il reste les cas où C^b est l'une des cliques critiques du manche de (C, R) ou $C^b \subseteq V_p$. Notons que dans les deux cas $N[C^a] \subsetneq N[C^b]$. En effet, si C^b est sur le manche, il ne peut pas être après C^c car $N[C^c] \subsetneq N[C^b]$, donc C^b est avant C^a et $N[C^a] \subsetneq N[C^b]$. Si C^b est contenu dans V_p , il est adjacent à l'ensemble du peigne (C, R) par définition et aussi à V_p car $V_p \subseteq N[C^c] \subseteq N[C^b]$. Nous ne pouvons pas avoir $N[C^a] = N[C^b]$ car C^a et C^b étant des modules cliques, $C^a \cup C^b$ serait aussi un module clique, ce qui contredirait le fait que C^a et C^b sont des modules cliques maximaux. Par conséquent, $N[C^a]$ est strictement contenu dans $N[C^b]$, et il existe donc un sommet $r^b \in N[C^b] \setminus N[C^a]$. Par le fait que $C^b \prec C^c$, $R(C^b, C^c) = N[C^b] \setminus N[C^c]$ est un module. Néanmoins, ceci est en contradiction avec le fait que $R(C^b, C^c)$ contient r^b , qui ne voit pas C^a , et contient aussi les sommets de $R_i = R(C^a, C^c)$, qui voient C_a . Il ne peut donc pas y avoir de peigne contenant C^a et C^c comme cliques critiques consécutives dans son manche.

Par conséquent, si pour une clique critique C^c il existe deux autres cliques critiques $C^a \prec C^c$ et $C^b \prec C^c$, nous pouvons supprimer ces deux relations (comme dans la boucle [ligne 5](#) de l'[algorithme 2](#)), sans détruire aucune relation de précédence entre des cliques critiques consécutives dans un peigne.

À ce stade, la relation \prec est une forêt orientée, et pour tout peigne (C, R) de G avec les partitions $\{C_1, \dots, C_l\}$ et $\{R_1, \dots, R_l\}$, nous avons que $C_1 \prec C_2 \prec \dots \prec C_l$. De plus, pour tout $1 \leq i < l$, nous avons obligatoirement $R_i = R(C_i, C_{i+1}) = N[C_i] \setminus N[C_{i+1}]$ et R_l est une union des sommets des composantes connexes trivialement parfaites telle que $N(R_l) = C_l \cup N(R_{l-1})$.

L'[algorithme 2](#) vérifie à la [ligne 11](#) qu'il énumère seulement des peignes. Certains de ces peignes peuvent ne pas être critiques, mais nous montrons dans ce qui suit que tous les peignes critiques sont effectivement énumérés.

Soit (C', R') un peigne critique avec les partitions $\{C'_1, \dots, C'_l\}$ et $\{R'_1, \dots, R'_l\}$ et soit les ensembles associés V'_p et V'_f . Deux cliques critiques consécutives C'_i et C'_{i+1} de son manche satisfont $C'_i \prec C'_{i+1}$ comme cela a été vu dans la première partie de la preuve. En particulier, la boucle [ligne 7](#) de l'[algorithme 2](#) va considérer le chemin $C_1 \prec \dots \prec C_l$, avec $C_i = C'_i$ pour tout $1 \leq i \leq l$. Par définition des peignes, nous avons également $R_i = R'_i$ pour tout $i < l$. L'ensemble X construit par l'[algorithme 2](#) (l'[ligne 9](#)) est tel que $X = R'_l \cup V'_f$. Montrons que l'ensemble R_l construit à la [ligne 10](#) est égal à R'_l . L'ensemble R_l est l'union des sommets des composantes connexes de $G[X]$ qui induisent un module trivialement parfait de voisinage $C_l \cup N(R_{l-1})$, en particulier il est maximal pour cette propriété. Ainsi, $R'_l \subseteq R_l$ et puisque (C', R') est un peigne critique, donc maximal pour l'inclusion, nous avons $R'_l = R_l$. Nous concluons que le peigne (C', R') est énuméré par l'[algorithme 2](#), ce qui prouve la partie combinatoire du lemme.

L'[algorithme 2](#) est clairement polynomial. Une analyse plus détaillée montre qu'il peut être implémenté pour fonctionner en $O(n^2m)$. En particulier, la complexité est donnée par la première et la dernière boucle **pour chaque**. Toutes deux font $O(n^2)$ itérations : par définition il y a au plus n cliques critiques, et puisque la relation \prec est une forêt, il y a $O(n^2)$ chemins. Chaque itération peut être exécutée en temps linéaire, en particulier, tester qu'une paire (C, R) est un peigne peut être effectué en temps $O(n + m)$. \square

Lemme 4.10. *Il existe un algorithme en temps polynomial qui produit une instance (G', k) telle qu'aucune des règles [règles 4.1](#) à [4.5](#) ne peut être appliquée.*

Preuve. Premièrement, nous appliquons les [règles 4.1](#) et [4.2](#), ce qui peut être fait en temps polynomial d'après le [lemme 3.5](#). Ensuite nous appliquons les autres règles sur les modules trivialement parfaits puis sur les peignes. Pour les modules, il suffit de réduire les modules forts. Rappelons qu'ils peuvent être énumérés en temps linéaire [124]. Pour chaque module fort M , nous pouvons vérifier en temps polynomial s'il est trivialement parfait [129]. Nous pouvons en outre vérifier si M contient un anti-couplage de taille $(k + 1)$ en trouvant un couplage maximum dans le graphe complémentaire $\overline{G[M]}$, en utilisant par exemple l'algorithme d'Edmonds [62]. Si M a un grand anti-couplage, alors nous pouvons appliquer la [règle 4.3](#). Sinon, si $|M| \geq 11k + 2$ alors il peut être réduit en utilisant la [règle 4.4](#). En effet, $G[M]$ contient dans ce cas au plus une composante connexe C avec plus de k sommets, telle que $|M \setminus V(C)| \leq k$ (sinon M contiendrait un anti-couplage de taille $(k + 1)$). Nous construisons un peigne (C, R) de C avec le [lemme 4.5](#) tel que $|R| \leq 4k$. Ainsi $|C| > 6k + 2$ et l'[observation 4.3](#) implique que la [règle 4.4](#) s'applique.

Il reste à prouver que les peignes qui ne sont pas inclus dans des modules trivialement parfaits peuvent être réduits en temps polynomial. Pour ce faire nous énumérons tous les peignes critiques à l'aide de l'algorithme 2. Il est suffisant de réduire ces peignes car tous les peignes qui ne sont pas inclus dans des modules trivialement parfaits sont inclus dans des peignes critiques par définition. Étant donné un peigne critique les règles 4.4 et 4.5 peuvent y être trivialement appliquées en temps polynomial. \square

4.2.6 Borner la taille d'une instance réduite

Nous montrons maintenant que toute instance positive de TRIVIALY PERFECT EDITION réduite par les règles 4.1 à 4.5 contient $O(k^2)$ sommets. Pour obtenir cette borne, nous avons besoin de la structure et du résultat qui suivent.

Définition 4.6 (PAC-clôture [67]). Soit $T = (V, E)$ un arbre enraciné et $A \subseteq V(T)$. La clôture par plus petit ancêtre commun (PAC-clôture) A' de A est obtenue comme suit. Initialement, $A' = A$. Ensuite, tant qu'il existe $x, y \in A'$ dont le plus petit ancêtre commun w n'est pas dans A' , ajouter w à A' . La PAC-clôture de A est le dernier ensemble A' obtenu par cette procédure.

Lemme 4.11 ([67]). Soit $T = (V, E)$ un arbre enraciné, $A \subseteq V(T)$ et $A' = \text{PAC-clôture}(A)$. Alors $|A'| \leq 2 \cdot |A|$ et pour toute composante connexe C de $T \setminus A'$, $|N_T(C)| \leq 2$.

Théorème 4.1. TRIVIALY PERFECT EDITION admet un noyau avec $O(k^2)$ sommets.

Preuve. Soit $(G = (V, E), k)$ une instance de TRIVIALY PERFECT EDITION réduite par les règles 4.1 à 4.5 et F une k -édition de G . Soit $H = G \Delta F$ et soit $\mathcal{T} = (T, \mathcal{B})$ la décomposition en clique universelle de H . Le graphe G n'est pas nécessairement connexe, donc T est une forêt. Soit A l'ensemble des noeuds $t \in V(T)$ tel que le sac B_t contient un sommet affecté par F . Puisque $|F| \leq k$, nous avons $|A| \leq 2k$. Soit $A' \subseteq V(T)$ l'ensemble contenant les noeuds de PAC-clôture(A) et la racine de chaque composante connexe de T (dans le cas où la clôture A' ne les contiendrait pas). Observons que la règle 4.1 implique qu'il y a au plus $2k$ composantes connexes dans H et donc $2k$ racines. De cette observation et d'après le lemme 4.11 nous déduisons que $|A'| \leq 6k$.

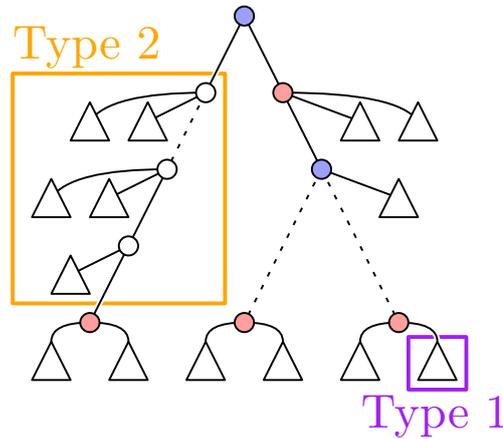


Figure 4.9: La décomposition en clique universelle d'une composante connexe de H . Les noeuds rouges sont ceux dont les sacs contiennent un sommet affecté par F . Les noeuds bleus sont les sommets de $A' \setminus A$.

Soit D une composante connexe de $T \setminus A'$. Par construction de A' (qui, pour chaque paire de noeuds, contient également leur plus petit ancêtre commun dans T), seuls trois cas sont possibles (voir la figure 4.9) :

- $N_T(D) = \emptyset$, D est une composante connexe de T ,
- $N_T(D) = \{a\}$, D est un sous-arbre de T dont le parent $a \in A'$,

- $N_T(D) = \{a_1, a_2\}$, avec un des noeuds $a_1, a_2 \in A'$ étant l'ancêtre de l'autre dans T .

Nous disons que ces composantes connexes sont respectivement de type 0, 1 ou 2. Pour D une composante connexe de $T \setminus A'$, nous définissons $W(D) = \bigcup_{t \in D} B_t$ l'ensemble de sommets de G correspondant aux sacs associés aux noeuds de D .

Observons qu'il n'y a pas de composante connexe de type 0, sinon $W(D)$ induirait une composante connexe trivialement parfaite de G et la règle 4.1 aurait été appliquée sur D , contredisant que G est une instance réduite.

Considérons ensuite l'ensemble $\{D_1, D_2, \dots, D_r\}$ des composantes de type 1 de $T \setminus A'$ attachées dans T à un même noeud $a \in A'$. Nous montrons que $W_a = W(D_1) \cup W(D_2) \cup \dots \cup W(D_r)$ est un module trivialement parfait de G . Dans le graphe H , l'ensemble de sommets W_a est par construction un module. Puisque aucun sommet de W_a n'a été affecté par l'édition F , W_a est aussi un module trivialement parfait de G par hérédité. Par le lemme 4.6, nous avons $|W_a| = O(k)$. Il y a au plus $|A'| \leq 6k$ tels ensembles W_a , par conséquent l'ensemble des sommets de G dans les sacs de composantes de type 1 est de taille $O(k^2)$.

Enfin, considérons les composantes connexes D de type 2 de $T \setminus A'$ qui ont deux voisins dans T . Soit a_1 et a_2 ces voisins tels que a_1 est l'ancêtre de a_2 . Soit $\{a_1, t_1, \dots, t_l, a_2\}$ le chemin de a_1 à a_2 dans T . La composante D peut être vue comme le peigne de manche $\{B_{t_1}, \dots, B_{t_l}\}$. Plus précisément, par construction de la décomposition en clique universelle, $W(D)$ peut être partitionné en un peigne (C, R) de H : la partition en cliques critiques de C est $\{C_1 = B_{t_1}, \dots, C_l = B_{t_l}\}$, et pour chaque $1 \leq i < l$, R_i correspond à l'union des sacs des sous-arbres enracinés avec les descendants de t_i (sauf celui enraciné en t_{i+1}). Enfin R_l correspond à l'union des sacs des sous-arbres enracinés avec les descendants de t_l (sauf celui enraciné en a_2). Puisque (C, R) n'a pas été affecté par F , c'est aussi un peigne de G . Ainsi, pour chaque composante D de type 2, $W(D)$ contient $O(k)$ sommets par le lemme 4.8. Puisque T est une forêt, il y a au plus $|A'| - 1 \leq 6k - 1$ telles composantes dans $T \setminus A'$. Par conséquent, l'ensemble des sacs contenant des composantes connexes de type 2 de $T \setminus A'$ contient $O(k^2)$ sommets.

Il reste à borner l'ensemble des sommets de G qui sont dans des sacs des noeuds de A' . Le sac d'un noeud de $A' \setminus A$ est une clique critique de G , et est donc de taille au plus $k + 1$ par la règle 4.2. Ainsi, l'ensemble des sommets contenus dans les sacs des noeuds de $A' \setminus A$ est de taille $O(k^2)$. Les sacs des noeuds de A sont des cliques critiques de H mais pas nécessairement de G , néanmoins, par le lemme 3.4, l'ensemble des sommets contenus dans les sacs des noeuds de A est de taille $O(k^2)$.

Nous concluons que $|V(G)| = O(k^2)$. Pour finir, le graphe peut être réduit par les règles 4.1 à 4.5 en temps polynomial grâce au lemme 4.10. \square

4.2.7 Noyaux pour la version complétion et délétion

Le noyau pour l'édition vers les graphes trivialement parfaits présenté dans la section 4.2 fonctionne aussi pour les versions délétion et complétion du problème. Les preuves de sûreté des règles de réduction sont aussi correctes en remplaçant la mention de « édition » par « complétion » ou « délétion ».

Plus en détail, pour prouver la sûreté des règles 4.3 à 4.5, la k -édition F^* pour le graphe original qui est construite à partir d'une k -édition F' pour l'instance réduite n'utilise que les opérations qui ont été faites par F' . En particulier, si F' est une délétion, c'est-à-dire qu'elle ne contient que des arêtes, alors F^* ne contient que des arêtes aussi, et donc que F^* est effectivement une délétion. Remarquons que dans la preuve de la règle 4.3, le voisinage du module M doit être une clique dans $H = G \triangle F$ pour toute k -édition F de G . Cela implique que si nous sommes restreints à la suppression d'arêtes et que $N_G(M)$ n'est pas une clique dans G , alors (G, k) est une instance négative du problème délétion.

Les règles 4.1 et 4.2 étant elles aussi sûres pour les versions complétion et délétion, nous obtenons le théorème qui suit.

Théorème 4.2. *TRIVIAALLY PERFECT COMPLETION et TRIVIAALLY PERFECT DELETION admettent un noyau avec $O(k^2)$ sommets.*

Rappelons qu'un noyau quadratique est déjà connu pour le problème TRIVIAALLY PERFECT COMPLETION [10], et bien qu'étant tous deux quadratiques ce noyau est de plus petite taille que celui que nous avons présenté.

4.3 Noyau pour THRESHOLD EDITION

Nous expliquons dans cette section comment adapter les règles introduites dans la section précédente pour définir un noyau avec $O(k^2)$ sommets pour le problème THRESHOLD EDITION. Notons que l'existence d'un noyau avec un nombre quadratique de sommets a déjà été montrée pour ce problème par Drange *et al.* [52]. Ils utilisent une approche basée sur un modulateur. Notre noyau n'améliore pas l'ordre sur le nombre de sommets, néanmoins il améliore le facteur constant et illustre la technique « décomposition et sommets affectés ».

4.3.1 Définition et structures

Les graphes threshold sont une sous classe des graphes trivialement parfaits. Ce sont les graphes $\{P_4, C_4, 2K_2\}$ -free. Ils sont la restriction des graphes trivialement parfaits aux graphes qui admettent une décomposition en clique universelle dont l'arbre est une chenille (chemin avec des sommets pendants) et dont les sacs associés aux pattes (sommets pendants) ne contiennent qu'un seul sommet. Une autre manière de voir cela est les graphes threshold sont les graphes qui admettent une partition threshold (C, I) où C est une clique et I un indépendant.

Définition 4.7 (partition threshold [103]). Soit $G = (V, E)$ un graphe. Une paire (C, I) de sous-ensembles de V disjoints est une partition threshold de G si :

- $G[C]$ est une clique qui peut être partitionnée en l cliques critiques $\{C_1, \dots, C_l\}$,
- $G[I]$ peut être partitionné en l indépendants critiques non-vides $\{I_1, \dots, I_l\}$,
- $N_G(C_i) \cap R = \bigcup_{j=i}^l R_j$ et $N_G(R_i) \cap C = \bigcup_{j=1}^i C_j$ pour $1 \leq i \leq l$,

Notons que la troisième condition implique que les familles $\{N_G(u) \mid u \in C\}$ et $\{N_G(u) \mid u \in I\}$ sont des familles imbriquées.

Les propriétés structurelles que nous avons introduites dans la section 4.1, à savoir la proposition 4.2 et le lemme 4.2, peuvent être adaptées en remplaçant les graphes trivialement parfaits par des indépendants ou des modules threshold.

Proposition 4.4. Si $G = (V, E)$ est un graphe threshold, alors pour tout $u \in V$ tel que $N_G(u)$ est une clique et pour tout indépendant H , le graphe $G(u \rightarrow H)$ est threshold.

Preuve. Observons que pour un graphe threshold, ajouter un faux jumeau v à un sommet u dont le voisinage est une clique résulte en un graphe G' qui est threshold. En effet, toute obstruction de G' doit contenir u et v , autrement le graphe G ne serait pas threshold. Hors, aucune des obstructions P_4 , C_4 et $2K_2$ des graphes threshold ne contient deux sommets qui sont faux jumeaux et ont un voisinage qui est une clique. Nous pouvons donc conclure qu'ajouter un faux jumeaux à tout sommet dont le voisinage est une clique conserve la propriété d'un graphe d'être threshold. Ceci implique que le graphe $G(u \rightarrow H)$ dans l'énoncé de la proposition est threshold. \square

Dans le lemme qui suit, toutes les composantes connexes K_i sont des sommets seuls à l'exception d'une qui peut être un graphe threshold. Notons que les K_i qui correspondent à des sommets seuls sont trivialement des modules threshold, nous retirons en conséquence cette condition dans la version du lemme pour les graphes threshold.

Lemme 4.12. Soit $G = (V, E)$ un graphe, $S \subseteq V$ une clique maximale de G et K_1, \dots, K_r les composantes connexes de $G \setminus S$. Le graphe G est threshold si et seulement si les conditions suivantes sont vérifiées :

- (i) K_i est un sommet seul pour tout $1 \leq i \leq r$ sauf pour éventuellement un K_j , où $G[K_j]$ contient une arête et est threshold (s'il existe supposons $j = r$),

- (ii) $\{N_G(V(K_i)) \mid 1 \leq i \leq r\}$ est une famille imbriquée,
- (iii) Si K_r n'est pas un sommet seul alors $V(K_r)$ est un module de G , $N_G(V(K_i)) \subseteq N_G(V(K_r))$ pour tout $1 \leq i < r$ et $|N_G(V(K_r))| = |S| - 1$.

Preuve. Nous commençons par supposer que G est un graphe threshold et soit $S \subseteq V$ une clique maximale de G . Un graphe threshold n'admet pas de $2K_2$ comme sous-graphe induit, ainsi puisque les composantes connexes K_1, \dots, K_r sont non-adjacentes deux à deux, au plus une peut contenir une arête. De plus si elle existe, supposons sans perte de généralité que ce soit K_r , qui est threshold par hérédité, ainsi la condition (i) est vérifiée. Si K_r est un sommet seul, alors tous les $K_i, 1 \leq i \leq r$ sont des sommets seuls, il existe donc une partition threshold (C, I) de G telle que $C = S$ et $\bigcup_{1 \leq i \leq r} K_i \subseteq I$. Par définition, $\{N_G(u) \mid u \in I\}$ forme une famille imbriquée dans C , ainsi la condition (ii) est vérifiée dans ce cas.

Supposons maintenant que K_r contient plus d'un sommet, et donc au moins une arête $\{u, v\}$ puisque K_r est connexe. Montrons premièrement que $N_G(V(K_i)) \subseteq N_G(V(K_r))$ pour tout $1 \leq i \leq r$. Supposons que ce ne soit pas le cas pour un K_i , il existe alors une arête $\{x, y\} \in E$ avec $x \in V(K_i)$ et $y \in S$ telle que $y \notin N_G(V(K_r))$. Les sommets $\{x, y, u, v\}$ induisent un $2K_2$ dans G , une contradiction. Par les mêmes arguments $|N_G(V(K_r))| \geq |S| - 1$, sinon il existerait une arête $\{x, y\}$ telle que $x, y \in S$ et x et y non adjacents à u et v . De plus puisque $\{u, v\}$ est une arête de G nous avons que $|N_G(V(K_r))| = |S| - 1$ sinon S ne serait pas maximum. Montrons à présent que $V(K_r)$ est un module de G . Supposons qu'il ne soit pas un module, alors il existe deux sommets $u, v \in K_r$ et un sommet $w \in S$ tels que w est adjacent à u mais pas à v . Soit le sommet $z \in S \setminus N_G(V(K_r))$ qui est non adjacent à u et v . Les sommets $\{u, v, w, z\}$ induisent un P_4 , ce qui est une contradiction. Nous avons montré que la condition (iii) est vérifiée si K_r n'est pas un sommet seul. Enfin en utilisant les mêmes arguments que dans le cas précédent nous pouvons montrer que $\{N_G(V(K_i)) \mid 1 \leq i < r\}$ est une famille imbriquée. Puisque $N_G(V(K_i)) \subseteq N_G(V(K_r))$ pour $i \neq r$, nous pouvons conclure que $\{N_G(V(K_i)) \mid 1 \leq i \leq r\}$ est une famille imbriquée. Ainsi la condition (ii) est également vérifiée dans ce cas.

Nous supposons maintenant que G est un graphe qui vérifie les conditions (i), (ii) et (iii). Nous construisons une partition threshold de G . Si tous les K_i sont des sommets seuls alors pour $C = S$ et $I = \bigcup_{1 \leq i \leq r} V(K_i)$, (C, I) est une partition threshold de G puisque $\{N_G(V(K_i)) \mid 1 \leq i \leq r\}$ est une famille imbriquée. Supposons maintenant que K_r contienne plus d'un sommet. Par la condition (i), K_r est threshold, ainsi il existe (C_r, I_r) une partition threshold de K_r . Nous affirmons que (C, I) où $C = C_r \cup (S \cap N_G(V(K_r)))$ et $I = V(G) \setminus C$ est une partition threshold de G . Observons d'abord que puisque $V(K_r)$ est un module de G , C est une clique. De plus puisque $N_G(V(K_i)) \subseteq N_G(V(K_r))$ pour $i \neq r$, le sommet x de $S \setminus N_G(V(K_r))$ n'est adjacent à aucun autre sommets de I et son voisinage est exactement $N_G(V(K_r))$. Puisque les sommets de $K_i, i \neq r$ ne sont pas adjacents à ceux de K_r , l'ensemble $I = I_r \cup \{x\} \cup \bigcup_{1 \leq i < r} V(K_i)$ est bien un indépendant. Enfin, les familles $\{N_G(V(K_i)) \mid 1 \leq i \leq r\}$ et $\{N_G(u) \mid u \in I_r\}$ sont imbriquées par définition. Puisque $N_G(V(K_i)) \subseteq N_G(V(K_r))$ pour $i \neq r$, nous concluons que la famille $\{N_G(u) \mid u \in I\}$ est imbriquée. Nous concluons que (C, I) est une partition threshold et que le graphe G est threshold. \square

4.3.2 Règles de réduction pour THRESHOLD EDITION

Rappelons que les graphes threshold sont les graphes sans P_4 , C_4 ni $2K_2$ comme sous-graphe induit. En particulier la dernière obstruction implique qu'ils sont composés d'une unique composante connexe non triviale et de sommets isolés. En particulier, nous ne pouvons pas retirer du graphe les composantes connexes threshold non triviales, car s'il y en a au moins deux, elles induisent un $2K_2$.

Règle 4.6. Soit $v \in V$ un sommet isolé de G . Retirer le sommet v .

Les graphes threshold ne sont pas clos pour l'ajout de vrais jumeaux ainsi que pour l'ajout de faux jumeaux. Nous ne pouvons donc pas utiliser notre règle habituelle pour borner les cliques critiques. Nous utilisons des règles introduites par Drange *et al.* [52] permettant de borner la taille des cliques et indépendants critiques dans les graphes threshold.

Règle 4.7. Soit $K \subseteq V$ une clique critique de G telle que $|K| > 2k + 2$. Retirer $|K| - (2k + 2)$ sommets de K .

Règle 4.8. Soit $I \subseteq V$ un indépendant critique de G tel que $|I| > 2k + 2$. Retirer $|I| - (2k + 2)$ sommets de I .

Lemme 4.13 ([52]). *Les règles 4.7 et 4.8 sont sûres.*

Nous pouvons définir les *peignes* pour les graphes threshold de manière analogue à la [définition 4.5](#) en remplaçant les modules trivialement parfaits R_i par des indépendants critiques I_i . Notamment, dans le cas des graphes threshold, un peigne correspond exactement à un sous-chemin de la décomposition threshold ([définition 4.7](#)). Nous pouvons définir les règles portant sur les peignes exactement de la même manière que pour le cas des graphes trivialement parfaits.

Règle 4.9. Soit (C, R) un peigne de G tel qu'il existe deux ensembles disjoints \mathcal{C}_a et \mathcal{C}_b où :

- \mathcal{C}_a est un $(2k + 1)$ -bourrage de $\{C_1, \dots, C_l\}$,
- \mathcal{C}_b est un $(2k + 1)$ -bourrage de $\{C_l, C_{l-1}, \dots, C_1\}$.

Retirer les sommets de $C' = C \setminus (\mathcal{C}_a \cup \mathcal{C}_b)$.

Règle 4.10. Soit (C, I) un peigne de G tel qu'il existe trois ensembles disjoints $\mathcal{I}_a, \mathcal{I}_b$ et \mathcal{I}_c où :

- \mathcal{I}_a est un $(2k + 1)$ -bourrage de $\{I_1, \dots, I_l\}$,
- $\mathcal{I}_c = \{I_l, \dots, I_q\}$ est un $(2k + 1)$ -bourrage de $\{I_l, \dots, I_1\}$,
- \mathcal{I}_b est un $(2k + 1)$ -bourrage de $\{I_{q-1}, \dots, I_1\}$,

Retirer les sommets de $I' = I \setminus (\mathcal{I}_a \cup \mathcal{I}_b \cup \mathcal{I}_c)$.

Les [règles 4.9](#) et [4.10](#) peuvent être montrées de manière similaire aux [règles 4.4](#) et [4.5](#). En particulier, il suffit de remplacer les mentions de « module trivialement parfait » par *indépendant critique*, et de remplacer les mentions de la [proposition 4.2](#) et du [lemme 4.2](#) par la [proposition 4.4](#) et le [lemme 4.12](#). Notons que dans le [lemme 4.12](#), les composantes connexes K_i (sauf au plus une) correspondent à des uniques sommets, les dents d'un peigne qui correspondent à des indépendants critiques sont donc des unions des sommets de K_i . De plus dans le cas particulier où la composante connexe K_r est un module threshold et non un unique sommet, cette composante connexe K_r ne pose pas d'inconvénient dans les preuves car son voisinage dans la clique maximale S contient tous les voisinages des autres K_i d'après le [lemme 4.12](#). Plus en détail, elle serait une γ -composante dans la preuve de sûreté de la [règle 4.9](#) et une β -composante dans la preuve de sûreté de la [règle 4.10](#).

Lemme 4.14. *Les règles 4.9 et 4.10 sont sûres.*

Ces règles permettent de borner à un nombre linéaire les sommets contenus dans un peigne.

Lemme 4.15. *Supposons que l'instance (G, k) est réduite par les règles 4.7 à 4.10. Soit (C, I) un peigne de G , alors $|C \cup R| = 20k + 10$.*

Preuve. Par les [règles 4.7](#) et [4.8](#), chaque clique critique C_i et indépendant critique I_i contient au plus $2k + 2$ sommets. Par l'[observation 4.1](#), tout $(2k + 1)$ -bourrage de $\{C_1, \dots, C_l\}$ ou $\{I_1, \dots, I_l\}$ contient au plus $4k + 2$ sommets. Ainsi $|C| \leq 8k + 4$, sinon on pourrait trouver deux $(2k + 1)$ -bourrages *disjoints* de $\{C_1, \dots, C_l\}$ et de $\{C_l, \dots, C_1\}$ et la [règle 4.9](#) pourrait être appliquée. De même, $|I| \leq 12k + 6$, sinon on pourrait trouver trois $(2k + 1)$ -bourrages disjoints de I qui répondent aux exigences de la [règle 4.10](#). Nous concluons que $|C \cup R| \leq 20k + 10$. \square

4.3.3 Borner la taille d'une instance réduite

Puisque les graphes threshold admettent également une décomposition en clique universelle, nous pourrions adapter de manière immédiate la preuve du [théorème 4.1](#) au problème THRESHOLD EDITION. Cependant une analyse plus fine utilisant une partition threshold permet d'obtenir une borne avec un plus petit facteur constant. Notons que le noyau proposé par Drange *et al.* [52] contient au plus $336k^2 + 388k + 92$ sommets.

Théorème 4.3. *THRESHOLD EDITION admet un noyau qui contient au plus $50k^2 + 50k + 10$ sommets.*

Preuve. Soit $(G = (V, E), k)$ une instance positive de THRESHOLD EDITION réduite par les [règles 4.6 à 4.10](#) et F une k -édition de G . Soit $H = G \triangle F$, (C, I) une décomposition threshold de H et soit $\{C_1, \dots, C_l\}$ et $\{I_1, \dots, I_l\}$ les partitions de C et I , respectivement. Soit A l'ensemble des paires (C_i, I_i) telles que C_i ou I_i contient un sommet affecté par F . Puisque $|F| \leq k$, nous avons $|A| \leq 2k$. Le graphe étant réduit d'après les [règles 4.7 et 4.8](#), les paires de A contiennent en tout au plus $2k \cdot 4k + 4$ sommets non affectés et $2k$ sommets affectés, soit $10k^2 + 10k$ sommets.

Observons maintenant qu'il existe au plus $2k + 1$ sous-séquences (maximales) continues de $\{(C_1, I_1), \dots, (C_l, I_l)\}$ qui ne contiennent pas d'éléments de A . Une telle sous-séquence

$$\{(C_x, I_x), \dots, (C_y, I_y)\}, 1 \leq x < y \leq l$$

correspond à un peigne $(C_x \cup \dots \cup C_y, I_x \cup \dots \cup I_y)$ de G puisque aucun C_j ou I_j pour $x \leq j \leq y$ ne contient de sommets affectés. Puisque le graphe est réduit par les [règles 4.9 et 4.10](#), d'après le [lemme 4.15](#), un tel peigne contient au plus $20k + 10$ sommets. Nous concluons donc que $|V(G)| \leq (2k + 1)(20k + 10) + 10k^2 + 10k = 50k^2 + 50k + 10$.

Pour finir, notons que la [règle 4.6](#) peut être trivialement appliquée en temps linéaire, que les [règles 4.7 et 4.8](#) peuvent l'être également d'après le [lemme 3.5](#). Enfin les [règles 4.9 et 4.10](#) peuvent être également appliquées en temps polynomial en énumérant les peignes critiques à l'aide d'un algorithme similaire à celui utilisé pour les graphes trivialement parfaits ([algorithme 2](#)). \square

Nous pouvons terminer cette section en mentionnant le cas des graphes chaînes, définis de manière similaire aux graphes threshold. En particulier ces graphes admettent une partition en deux indépendants où les voisinages des sommets de chaque partie forment une famille imbriquée. Drange *et al.* [52] ont proposé des noyaux avec un nombre quadratique de sommets pour CHAIN EDITION ainsi que ses versions complétion et délétion, en adaptant leur noyau pour THRESHOLD EDITION. Le noyau que nous venons de montrer pour THRESHOLD EDITION peut également être adapté à ces problèmes de modification d'arêtes vers les graphes chaînes.

Chapitre 5

Graphes blocs et strictement cordaux

Dans ce chapitre nous présentons des noyaux polynomiaux pour les problèmes BLOCK GRAPH EDITION, BLOCK GRAPH DELETION et pour le problème STRICTLY CHORDAL EDITION et ses variantes complétion et délétion. Le noyau pour les problèmes de modification d'arêtes pour les graphes strictement cordaux a fait l'objet d'une première publication avec Anthony Perez et Ioan Todinca [60]. Les résultats pour les graphes blocs ont été ajoutés dans la version complète de l'article avec le concours de Mathis Rocton [58].

Les graphes *blocs* sont les graphes dans lesquels toute composante biconnexe est une clique [79]. Cela implique notamment que les séparateurs minimaux des graphes blocs sont de taille 1. Ces graphes peuvent aussi être caractérisés comme les graphes cordaux sans diamants comme sous-graphe induit [9]. Une généralisation naturelle des graphes blocs sont les graphes *strictement cordaux*, aussi définis sous le nom de *block duplicate*. Ils peuvent être obtenus à partir de graphes blocs en choisissant de manière répétée un sommet séparateur u et en ajoutant un vrai jumeau v de u , c'est-à-dire un sommet v adjacent à u et à tous les voisins de u [72]. Ils peuvent aussi être caractérisés comme les graphes cordaux sans gemmes ou dards comme sous-graphe induit (illustrés dans la [figure 5.1](#)). Les graphes strictement cordaux sont une super-classe des graphes 3-leaf power [49] et sont une sous-classe des graphes 4-leaf power [89] ainsi que des graphes ptolémaïques.

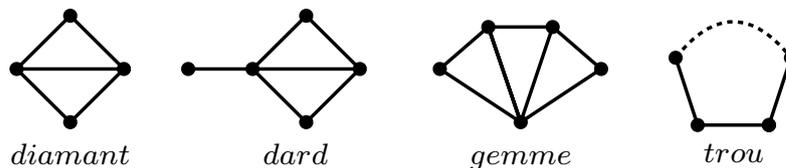


Figure 5.1: Les sous-graphes interdits des graphes blocs et strictement cordaux.

Structure du chapitre Nous présentons dans ce chapitre :

- un noyau avec $O(k^2)$ sommets pour les problèmes BLOCK GRAPH EDITION et BLOCK GRAPH DELETION,
- un noyau avec $O(k^4)$ sommets pour le problème STRICTLY CHORDAL EDITION,
- un noyau avec $O(k^3)$ sommets pour les problèmes STRICTLY CHORDAL COMPLETION et STRICTLY CHORDAL DELETION.

Nous rappelons que le problème BLOCK GRAPH COMPLETION est polynomial à résoudre, il suffit de compléter toutes les composantes biconnexes en une clique. Nous commençons dans la [section 5.1](#) par donner des propriétés structurelles des graphes blocs et strictement cordaux ainsi que d'autres notions utiles pour définir nos algorithmes de noyaux. Nous montrons dans la [section 5.2](#) que tous les problèmes étudiés dans ce chapitre sont NP-complets. Nous présentons ensuite le noyau pour BLOCK GRAPH EDITION et sa variante délétion dans la [section 5.3](#). Enfin, nous présentons le

noyau pour STRICTLY CHORDAL EDITION dans la [section 5.4](#). Nous terminons par montrer dans la [section 5.4.5](#) comment adapter et améliorer ce noyau pour STRICTLY CHORDAL COMPLETION et STRICTLY CHORDAL DELETION.

5.1 Définitions et propriétés

Nous commençons cette section en donnant des propriétés structurelles sur les graphes blocs et strictement cordaux. Notamment, nous donnons des caractérisations des graphes strictement cordaux, nous définissons l'opération de *jointure* qui permet de construire des graphes cordaux, blocs ou strictement cordaux sous certaines conditions. Enfin nous donnons un résultat combinatoire sur les sous-graphes connexes couvrant un ensemble de sommets dans les graphes blocs, ce résultat est essentiel pour borner la taille de notre noyau.

Rappelons que les sommets du graphe des cliques critiques $\mathcal{C}(G)$ correspondent aux cliques critiques de G , ses arêtes aux adjacences entre ces cliques ([définition 3.1](#)). Les graphes strictement cordaux admettent plusieurs caractérisations.

Théorème 5.1 ([108]). *Soit $G = (V, E)$ un graphe. Les propriétés suivantes sont équivalentes :*

1. G est un graphe strictement cordal,
2. Le graphe des cliques critiques $\mathcal{C}(G)$ est un graphe bloc,
3. G ne contient pas de dards, de diamants ou de trous comme sous-graphe induit,
4. G est cordal et les séparateurs minimaux de G sont deux à deux disjoints.

Jointure \otimes Nous nous intéressons maintenant à la notion de *jointure* (opération notée \otimes) entre deux graphes. Cette opération joue un rôle majeur dans les preuves des règles de réduction des noyaux présentés.

Définition 5.1. Soit $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ deux graphes disjoints et soit $S_1 \subseteq V_1, S_2 \subseteq V_2$. La *jointure* de G_1 et G_2 sur S_1 et S_2 , notée $(G_1, S_1) \otimes (G_2, S_2)$, est le graphe :

$$(V_1 \cup V_2, E_1 \cup E_2 \cup (S_1 \times S_2))$$

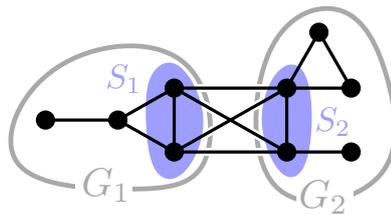


Figure 5.2: Illustration d'un graphe $(G_1, S_1) \otimes (G_2, S_2)$.

Nous présentons maintenant des propriétés structurelles pour l'opération de jointure. Le lemme qui suit concerne les séparateurs minimaux dans un graphe $G = (G_1, S_1) \otimes (G_2, S_2)$ où G_1 et G_2 sont cordaux, et où S_1 et S_2 sont des cliques. En particulier les séparateurs minimaux de G sont ceux de G_1 et G_2 plus éventuellement S_1 ou S_2 sous certaines conditions. C'est par exemple le cas de S_1 dans la [figure 5.2](#), où S_1 est un séparateur minimal de G mais pas de G_1 .

Lemme 5.1. *Soit $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ deux graphes cordaux disjoints. Soit deux cliques non vides $S_1 \subseteq V_1$ et $S_2 \subseteq V_2$ et soit $G = (G_1, S_1) \otimes (G_2, S_2)$. Le graphe G est cordal, de plus tout séparateur minimal S de G est aussi un séparateur minimal de G_1 ou de G_2 , excepté si $S = S_i$ pour $i \in \{1, 2\}$ et S_i est strictement inclus dans une clique maximale de G_i .*

Preuve. Le graphe G est cordal puisque tout cycle de G est soit contenu dans l'un de G_1 et G_2 ou, s'il les intersecte tous les deux, il contient au moins trois sommets de $S_1 \cup S_2$ qui induisent une clique dans G . Dans les deux cas, si le cycle a quatre sommets ou plus, alors il contient une corde.

Tout séparateur minimal S d'un graphe cordal G est l'intersection de deux cliques maximales Ω et Ω' de G (cf. le théorème 7 et le lemme 5 dans [70]). De plus S est un ab -séparateur minimal pour tout $a \in \Omega \setminus S$, $b \in \Omega' \setminus S$. En particulier, S est strictement inclus dans chacune des cliques. Par construction de G , n'importe laquelle de ses cliques maximales est soit une clique maximale de G_1 ou de G_2 , soit exactement $S_1 \cup S_2$.

Si $S = \emptyset$, alors le graphe G n'est pas connexe. Cela n'arrive que si l'un des graphes G_1 ou G_2 n'est pas connexe, puisque S_1 et S_2 ne sont pas vides. Nous concluons que S est un séparateur minimal de G_1 ou G_2 .

Dans ce qui suit nous supposons que S n'est pas vide. Considérons en premier le cas où ni Ω ni Ω' n'est égal à $S_1 \cup S_2$. Alors Ω et Ω' sont toutes les deux contenues dans le même G_i , pour $i \in \{1, 2\}$ (sinon leur intersection S serait vide). Supposons qu'elles soient toutes les deux dans G_1 , nous affirmons que $S = \Omega \cap \Omega'$ est un séparateur minimal de G_1 . Prenons $a \in \Omega \setminus S$ et $b \in \Omega' \setminus S$. Puisque S sépare a et b dans G , il les sépare aussi dans G_1 , qui est un sous-graphe de G . Mais S est dans le voisinage commun de a et b dans G_1 , donc S est minimal parmi les ab -séparateurs de G_1 , ce qui implique la conclusion.

Il nous reste le cas où $\Omega = S_1 \cup S_2$ et Ω' est contenue dans l'un des graphes G_1 et G_2 . Supposons sans perte de généralité que ce soit dans G_1 , par conséquent, $S \subseteq S_1$. Si cette inclusion est stricte, nous affirmons que S est aussi un séparateur minimal de G_1 . En effet, S est un séparateur minimal dans G pour tout $a \in \Omega \setminus S$ et $b \in \Omega' \setminus S$, donc nous pouvons choisir $a \in S_1 \setminus S$. À nouveau S sépare a et b dans le sous-graphe G_1 de G , et comme S est dans le voisinage commun de a et b dans G_1 , il est nécessairement un séparateur minimal de G . Nous en déduisons que, si S n'est pas un séparateur minimal de G_1 , nous devons avoir $S = S_1$. Nous concluons que S est strictement inclus dans la clique maximale Ω' de G_1 , ce qui prouve le lemme. \square

Nous présentons maintenant des conditions suffisantes sur les ensembles S_1 et S_2 pour que si les graphes G_1 et G_2 sont blocs ou strictement cordaux, alors le graphe $(G_1, S_1) \otimes (G_2, S_2)$ le soit aussi.

Lemme 5.2. *Soit $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ deux graphes blocs disjoints et soit deux sous-ensembles de sommets non vides $S_1 \subseteq V_1$ et $S_2 \subseteq V_2$. Le graphe $G = (G_1, S_1) \otimes (G_2, S_2)$ est un graphe bloc si pour $i \in \{1, 2\}$, S_i est un sommet ou une clique maximale de G_i .*

Preuve. Observons que les séparateurs minimaux des graphes blocs sont des sommets puisque les composantes biconnexes des graphes blocs sont des cliques. D'après le lemme 5.1, G est cordal et ses séparateurs minimaux sont ceux de G_1 et G_2 plus éventuellement S_1 (resp. S_2) s'il est strictement inclus dans une clique maximale de G_1 (resp. G_2). Ainsi, si pour $i \in \{1, 2\}$, S_i est un sommet ou une clique maximale de G_i , les séparateurs minimaux de G sont des sommets et donc G est un graphe bloc. \square

Lemme 5.3. *Soit $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ deux graphes strictement cordaux disjoints et soit deux sous-ensembles de sommets non vides $S_1 \subseteq V_1$ et $S_2 \subseteq V_2$. Le graphe $G = (G_1, S_1) \otimes (G_2, S_2)$ est un graphe strictement cordal si pour $i \in \{1, 2\}$, S_i est une clique critique, une clique maximale ou intersecte exactement une clique maximale de G_i .*

Preuve. D'après le lemme 5.1, G est cordal et ses séparateurs minimaux sont ceux de G_1 et G_2 plus éventuellement S_1 (resp. S_2) s'il est strictement inclus dans une clique maximale de G_1 (resp. G_2). Si S_i intersecte exactement une seule clique maximale de G_i , il est immédiat que S_i n'intersecte aucun séparateur minimal de G_i . Si S_i est une clique critique, nous affirmons que S_i est un séparateur minimal de G_i ou qu'il n'intersecte aucun séparateur minimal de G_i . En effet, s'il existe un séparateur (clique) minimal S de G_i tel que $S_i \subset S$, alors puisque S_i est une clique critique, il existe $u \in S_i$, $v \in S \setminus S_i$ et $w \in V(G_i) \setminus S$ tels que w est adjacent soit à u soit à v . Supposons que w soit adjacent à u et non à v , alors u est dans un vw -séparateur minimal qui intersecte S et qui en est différent. C'est une contradiction puisque G_i est strictement cordal et que le théorème 5.1 indique que les séparateurs minimaux des graphes strictement cordaux sont

disjoints deux à deux. Par conséquent les séparateurs minimaux de G sont disjoints deux à deux et par le [théorème 5.1](#) G est strictement cordal. \square

Dans les preuves de certaines règles pour le noyau pour STRICTLY CHORDAL EDITION et ses variantes, nous devons effectuer une jointure de graphes G_2, \dots, G_r avec un graphe G_1 sur le même ensemble de sommets $S_1 \subseteq V(G_1)$. L'observation suivante nous garantit que cette opération peut-être faite en assurant que le graphe obtenu soit strictement cordal.

Observation 5.1. Soit $G_1 = (V_1, E_1), \dots, G_r = (V_r, E_r)$ un ensemble de graphes strictement cordaux disjoints et pour chaque $i \in \{2, \dots, r\}$ soit $S_i \subseteq V_i$ une clique critique, une clique maximale ou un ensemble qui intersecte exactement une clique maximale de G_i . Si S_1 est une clique critique ou intersecte exactement une clique maximale de G_1 , alors S_1 est une clique critique ou intersecte exactement une clique maximale de $(G_1, S_1) \otimes (G_2, S_2)$. Cela implique par le [lemme 5.3](#) que $(G_1, S_1) \otimes (\bigcup_{2 \leq i \leq r} G_i, \bigcup_{2 \leq i \leq r} S_i)$ est strictement cordal. Cette construction est illustrée dans la [figure 5.3](#).

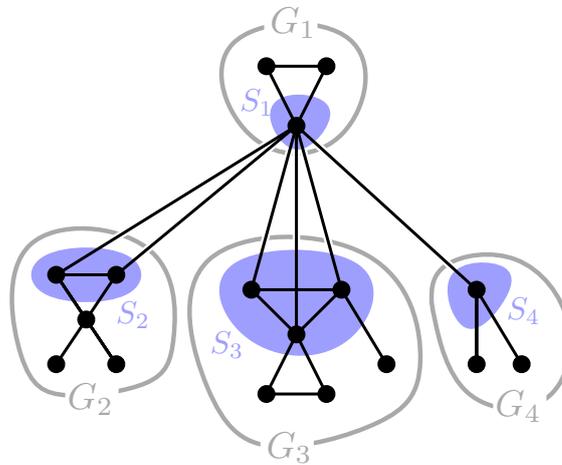


Figure 5.3: Illustration d'un graphe $G = (G_1, S_1) \otimes (\bigcup_{2 \leq i \leq 4} G_i, \bigcup_{2 \leq i \leq 4} S_i)$ construit comme dans l'[observation 5.1](#). Ici, S_1 intersecte exactement une clique maximale de G_1 et est une clique critique de G ; S_2 intersecte exactement une clique maximale de G_2 ; S_3 est une clique maximale de G_3 ; S_4 est une clique critique de G_4 .

Sous-graphe couvrant Nous présentons un résultat combinatoire sur les sous-graphes connexes couvrant un ensemble de sommets A dans les graphes blocs. Ce résultat est crucial pour borner la taille des instances réduites des problèmes étudiés dans ce chapitre. Pour montrer ce résultat nous avons besoin de la notion des *arbres blocs-séparateurs*. Notons qu'un sommet d'un séparateur de taille 1 est appelé un *sommet séparateur*.

Définition 5.2. Soit $G = (V, E)$ un graphe, l'*arbre blocs-séparateurs* $T_{BC}(G)$ de G est un arbre ayant deux types de noeuds :

1. les noeuds séparateurs qui correspondent aux sommets séparateurs de G ,
2. les noeuds blocs qui correspondent aux composantes biconnexes de G .

Il existe une arête entre un noeud séparateur et un noeud bloc dans $T_{BC}(G)$ si le sommet séparateur correspondant est contenu dans la composante biconnexe correspondante.

Dans un graphe bloc G , les noeuds blocs de $T_{BC}(G)$ correspondent à des cliques maximales de G puisque ses composantes biconnexes sont des cliques.

Lemme 5.4. Soit $G = (V, E)$ un graphe bloc connexe et soit un ensemble $A \subseteq V$. Soit T_A un sous-graphe induit connexe minimal de G qui couvre tous les sommets de A . Soit $f(T_A)$ l'ensemble des sommets de degré au moins 3 dans T_A . Nous avons les propriétés suivantes :

- (i) Le sous-graphe T_A est unique,
- (ii) $|f(T_A) \setminus A| \leq 3 \cdot |A|$,
- (iii) Le graphe $T_A \setminus (A \cup f(T_A))$ contient au plus $2 \cdot |A|$ composantes connexes.

Preuve. Observons que les sommets qui ne sont pas des sommets séparateurs de G appartiennent à une unique composante biconnexe de G . Par conséquent, nous pouvons associer à chaque sommet v de $V(G)$ un unique noeud $n(v)$ de $T_{BC}(G)$ comme suit : si v est un sommet séparateur, nous l'associons au noeud séparateur correspondant dans $T_{BC}(G)$, sinon nous l'associons au noeud bloc correspondant à l'unique composante biconnexe de G qui contient v . Observons maintenant que, pour deux sommets $u, v \in A$, tout chemin P de u à v dans G correspond au chemin P_{BC} de $n(u)$ à $n(v)$ dans $T_{BC}(G)$. En particulier, P contient tous les sommets correspondant aux noeuds séparateurs de P_{BC} . Par conséquent le sous-graphe couvrant T_A doit contenir tous les sommets de P . Nous en concluons que les sommets de T_A sont précisément les sommets de A , plus tous les sommets séparateurs de G correspondant aux noeuds séparateurs de tels chemins, ce qui implique l'unicité de T_A .

Nous définissons $T_{BC}(A)$ comme le sous-arbre de $T_{BC}(G)$ couvrant tous les noeuds de $n(A) = \{n(a) \mid a \in A\}$. Nous dénombrons les sommets de $f(T_A) \setminus A$. Si $|A| = 2$ alors $f(T_A) \setminus A = \emptyset$. Supposons maintenant que $|A| \geq 3$ et soit $a' \in f(T_A) \setminus A$. Par l'observation précédente, c'est un sommet séparateur de G , donc $n(a')$ est un noeud séparateur de $T_{BC}(A)$. Soit v un voisin de a' dans T_A . Par construction de T_A et de $T_{BC}(A)$, il existe un noeud bloc b adjacent à $n(a')$ dans $T_{BC}(A)$, tel que v et a' sont dans la clique maximale de G correspondant au noeud b . De plus, v est dans A , ou v est un sommet séparateur de G tel que le noeud séparateur $n(v)$ est adjacent à b dans $T_{BC}(A)$. Nous avons donc :

1. $n(a')$ est de degré au moins 3 dans $T_{BC}(A)$, ou
2. $n(a')$ est le voisin d'un noeud bloc b de degré au moins 3 dans $T_{BC}(A)$, ou si ce n'est aucun de ces deux cas,
3. $n(a')$ a exactement deux voisins b et b' dans $T_{BC}(A)$, les cliques maximales correspondantes à b et b' dans G contiennent au moins $p \geq 1$ sommets de A , où p plus le degré de b et b' dans $T_{BC}(A)$ est au moins 3.

Soit k le nombre de feuilles de $T_{BC}(A)$. Observons que pour toute feuille l de $T_{BC}(A)$, il existe un $a \in A$ tel que $n(a) = l$. Choisissons pour chaque feuille l un unique sommet $a_l \in A$ tel que $n(a_l) = l$, nous appelons a_l un sommet feuille. Notons que le nombre de sommets $a' \in f(T_A)$ correspondant aux deux premiers points ci-dessus est borné par $3k$. En effet, $n(a')$ est incident à une arête de $T_{BC}(A)$, ayant un noeud incident de degré au moins 3. Nous pouvons facilement vérifier que, dans tout arbre de k feuilles, le nombre de telles arêtes est borné par $3k$ (nous pouvons le montrer par induction sur le nombre de feuilles de l'arbre, en ajoutant les noeuds feuilles un par un). Nous dénombrons maintenant les sommets $a' \in f(T_A)$ du troisième type. Par le troisième point, a' a au moins un voisin $a \in A$ dans le graphe T_A , tel que a n'est pas un sommet feuille. Observons que a peut être dans le voisinage d'au plus deux sommets de $f(T_A)$ de ce troisième type, sinon nous serions dans le second cas. Ainsi, il y a au plus $|A| - k$ sommets semblables à a et $2(|A| - k)$ noeuds du troisième type. Il en résulte donc que $|f(T_A) \setminus A| \leq 3k + 2(|A| - k) \leq 3 \cdot |A|$.

Pour prouver le troisième point, le nombre de composantes connexes de $T_A \setminus (A \cup f(T_A))$, nous examinons à nouveau la situation dans $T_{BC}(A)$. Rappelons que tout noeud de degré au moins 3 dans $T_{BC}(A)$ est soit un noeud séparateur, auquel cas il est dans $f(T_A)$ (cas 1), soit un noeud bloc, mais alors tous ses voisins dans $T_{BC}(A)$ correspondent à des sommets de $f(T_A)$ (cas 2). Par conséquent, les composantes connexes de $T_A \setminus f(T_A)$ correspondent aux composantes connexes de $T_{BC}(A)$ après suppression de tous les noeuds de degré au moins 3. Ainsi, le nombre de telles composantes connexes est borné par $2k$. Retirer les sommets feuilles de A n'augmente pas le nombre de composantes connexes, et retirer les autres sommets de A augmente le nombre de composantes connexes d'au plus 1. Ainsi $T_A \setminus (A \cup f(T_A))$ contient au plus $2 \cdot |A|$ composantes connexes, ce qui conclut la preuve du lemme. \square

5.2 NP-complétude

Nous commençons par montrer la NP-complétude de BLOCK GRAPH DELETION et BLOCK GRAPH EDITION en donnant une réduction depuis CLUSTER DELETION et CLUSTER EDITION qui sont connus comme étant NP-complets [95, 114, 120]. Rappelons que les graphes cluster peuvent être caractérisés comme les graphes qui n'admettent pas de P_3 comme sous-graphe induit. De plus rappelons que problème BLOCK GRAPH COMPLETION admet un algorithme en temps polynomial en raison de la nature des graphes blocs : toute composante biconnexe doit être transformée en une clique (ces composantes peuvent être calculées en temps polynomial [83]).

Lemme 5.5. *BLOCK GRAPH DELETION et BLOCK GRAPH EDITION sont NP-complets.*

Preuve. Nous commençons par prouver que BLOCK GRAPH DELETION est NP-complet en faisant une réduction à partir de CLUSTER DELETION. Étant donné une instance $(G = (V, E), k)$ de CLUSTER DELETION, nous construisons une instance de BLOCK GRAPH DELETION en ajoutant un sommet universel u adjacent à tous les sommets de V . Soit $(G' = (V', E'), k)$ l'instance construite.

Nous montrons que le graphe G admet une k -délétion en un graphe cluster si et seulement si G' admet une k -délétion dans un graphe bloc. Supposons d'abord qu'il existe une k -délétion F de G en un graphe cluster. Le graphe $G - F$ est un graphe sans aucun P_3 comme sous-graphe induit. Considérons maintenant le graphe $H' = G' - F$. Par construction, $H'[V]$ ne contient pas de P_3 , donc H' est cordal et ne contient pas de diamant, donc H' est un graphe bloc.

Supposons maintenant qu'il existe une k -délétion F' de G' dans un graphe bloc. Nous affirmons que soit F' n'affecte pas le sommet universel u , soit nous pouvons construire une k -délétion F^* à partir de F' qui n'affecte pas u . Supposons que F' contient au moins une arête incidente à u .

Nous considérons tout d'abord le cas où $H' = G' - F'$ n'est pas connexe. Soit C la composante connexe qui contient u . Soit C' une autre composante connexe de H' et un sommet $v \in V(C')$. Définissons la délétion $F'' = F' \setminus \{\{u, v\}\}$. Par le lemme 5.2, $(C, \{u\}) \otimes (C', \{v\})$ est un graphe bloc, et les autres composantes connexes de H' sont inchangées. Par conséquent, $G' - F''$ est un graphe bloc et puisque $|F''| < |F'|$, F'' est une k -délétion. On itère cette construction jusqu'à obtenir une solution intermédiaire F_1^* telle que $H_1^* = G' - F_1^*$ soit connexe.

Si F_1^* n'affecte pas u , nous fixons $F^* = F_1^*$ et nous avons terminé. Sinon, il existe au moins un sommet v non adjacent à u dans H_1^* . Puisque H_1^* est connexe, nous pouvons choisir v à une distance exactement 2 de u . Puisque H_1^* est un graphe bloc, nous avons $N(u) \cap N(v) = \{w\}$. Considérons la composante biconnexe C contenant v et w , et soit $C' = C \setminus \{w\}$. Puisque H_1^* est un graphe bloc, C (et donc C') est une clique. Soit K_u et K_v les deux composantes connexes de $H_1^* - (\{w\} \times C')$ qui contiennent respectivement les sommets u et v . Ces composantes sont des graphes blocs par hérédité. Par le lemme 5.2 le graphe $H_2^* = (K_u, \{u\}) \otimes (K_v, \{C'\})$ est un graphe bloc. Observons que $(\{u\} \times C') \subseteq F_1^*$ puisque u est un sommet universel. Soit F_2^* la délétion telle que $H_2^* = G' - F_2^*$. Par construction, observons que $|F_2^*| = |F_1^*|$, donc F_2^* est une k -délétion de G' en un graphe bloc. Nous itérons cette construction jusqu'à ce que nous obtenions une k -délétion F^* de G' qui n'affecte pas u .

Pour conclure, observons que $H = G - F^*$ ne contient aucun P_3 sinon il y aurait un diamant dans $G' - F^*$ avec le sommet universel u , par conséquent H est un graphe cluster.

La même réduction peut être faite pour BLOCK GRAPH EDITION, il suffit d'observer qu'il n'y a pas d'arêtes ajoutées incidentes à u , et nous pouvons utiliser le même argument que ci-dessus sur les arêtes supprimées : s'il n'y a pas d'arête supprimée incidente à u nous avons terminé, sinon nous construisons un autre ensemble d'arêtes supprimées de même taille, mais contenant moins d'arêtes incidentes à u . Si, lors de la construction, nous devons supprimer une arête qui a été ajoutée par l'édition, alors la suppression de cette arête de l'ensemble d'édition se traduit par un ensemble d'édition strictement plus petit. Ceci conclut la preuve de ce lemme. \square

La NP-complétude de STRICTLY CHORDAL COMPLETION résulte directement de la preuve de NP-complétude pour 3-LEAF POWER COMPLETION de [49, Théorème 3]. Notamment, ils montrent la NP-complétude de la complétion vers des classes de graphes caractérisées par une famille d'obstructions qui contient les trous, les gemmes et d'autres obstructions contenant trois sommets indépendant (ce qui est le cas du dard).

Lemme 5.6. *STRICTLY CHORDAL COMPLETION est NP-complet.*

Nous montrons la NP-complétude de STRICTLY CHORDAL EDITION et STRICTLY CHORDAL DELETION en donnant une réduction depuis CLUSTER EDITION et CLUSTER DELETION respectivement.

Lemme 5.7. *STRICTLY CHORDAL EDITION et STRICTLY CHORDAL DELETION sont NP-complets.*

Preuve. Étant donné une instance $(G = (V, E), k)$ de CLUSTER EDITION, nous construisons une instance de STRICTLY CHORDAL EDITION en ajoutant une clique $U = \{u_1, \dots, u_{k+1}\}$ de taille $k+1$ adjacente à tous les sommets de V , et pour chaque sommet x dans V , $k+1$ sommets $\{v_1^x, \dots, v_{k+1}^x\}$ adjacents seulement à x . Soit $(G' = (V', E'), k)$ l'instance construite. Nous montrons que le graphe G admet une k -édition en graphe cluster si et seulement si G' admet une k -édition en graphe strictement cordal. Supposons d'abord qu'il existe une k -édition F de G en un graphe cluster. Le graphe $G \triangle F$ est un graphe sans aucun P_3 comme sous-graphe induit. Considérons le graphe $H' = G' \triangle F$. Par construction, $H'[V]$ ne contient pas de P_3 , donc H' est cordal et ne contient ni gemmes ni dards puisque ces obstructions contiennent un P_3 induit et un sommet adjacent à tous les sommets de ce P_3 . Il en résulte que H' est strictement cordal.

Supposons maintenant qu'il existe une k -édition F' de G' en un graphe strictement cordal $H' = G' \triangle F'$. Nous affirmons que $H = G \triangle F'$ est un graphe cluster. Par contradiction, supposons que $H = H'[V]$ contienne un $P_3 \{x, y, z\}$ où x, z sont les extrémités du chemin. Il existe alors $i, j \in \{1, \dots, k+1\}$ tels que $\{x, y, z, u_i, v_j^y\}$ forme un dard dans H' , ce qui contredit le fait que H' est strictement cordal, et donc H est un graphe cluster.

La même réduction peut être faite de CLUSTER DELETION à STRICTLY CHORDAL DELETION, montrant que ce dernier est NP-complet. \square

5.3 Noyau pour BLOCK GRAPH EDITION

Nous présentons dans cette section un noyau pour le problème BLOCK GRAPH EDITION qui contient un nombre quadratique de sommets. Nous commençons en donnant une description informelle du noyau. Considérons une instance positive $(G = (V, E), k)$ de BLOCK GRAPH EDITION, F une k -édition de G et soit $H = G \triangle F$. Puisque F est de taille au plus k , il existe au plus $2k$ sommets de H qui sont affectés. Soit A ces sommets et T le sous-graphe induit connexe minimal de H qui couvre les sommets de A . Définissons A' comme l'ensemble des sommets de degré au moins 3 dans T . D'après le [lemme 5.4](#), nous avons $|A' \setminus A| \leq 3 \cdot |A|$.

Nous définirons la notion de BG-branche, qui correspond aux sous-graphes de G qui induisent des graphes blocs. Nous nous intéresserons à deux types de BG-branches : celles dont un seul sommet est adjacent au reste du graphe, appelées *1-BG-branches*, et celles dont exactement deux sommets (non adjacents) sont adjacents au reste du graphe, appelées *2-BG-branches*. Les sommets de ces BG-branches adjacents au reste du graphe sont appelés des points d'attache. Nous montrerons que nous pouvons conserver uniquement les points d'attache des 1-BG-branches. Nous montrerons également que les 2-BG-branches peuvent être réduites à un nombre de sommets linéaire en k .

Nous montrerons qu'après réduction des BG-branches, il reste deux types de composantes connexes dans le graphe $H \setminus (A \cup A')$: celles qui sont adjacentes aux cliques maximales des sommets de $A \cup A'$, correspondant à des cliques critiques (qui peuvent être réduites à un nombre linéaire de sommets) et celles qui sont adjacentes à deux sommets non adjacents de $A \cup A'$ (qui correspondent à une 2-BG-branche dans H). Puisque H est un graphe bloc, il contient au plus $|A \cup A'| = O(k)$ cliques maximales de sommets de $A \cup A'$. D'après le [lemme 5.4](#), il y a au plus $4k$ composantes connexes dans $T \setminus (A \cup A')$, et chacune correspond à une 2-BG-branche dans le graphe H . Cela implique que $H \setminus (A \cup A')$ contient un nombre linéaire de composantes connexes, chacune contenant $O(k)$ sommets. Nous en concluons que le graphe H contient $O(k^2)$ sommets.

Remarque 5.1. Dans le reste de la section nous supposons que nous avons en entrée une instance $(G = (V, E), k)$ de BLOCK GRAPH EDITION.

5.3.1 Règles de réduction classiques

Les graphes blocs sont clos par union disjointe, nous commençons donc par redéfinir la règle portant sur les composantes connexes blocs (sûre par le [lemme 3.1](#)).

Règle 5.1. Soit C une composante connexe de G telle que C est bloc. Retirer les sommets de C .

Cependant remarquons que les graphes blocs ne sont pas clos pour l'ajout de vrais jumeaux. En effet, il n'est pas possible d'ajouter un jumeau à un sommet séparateur car cela créerait un séparateur de taille 2. Ceci implique que le [lemme 3.2](#) ne peut pas être appliqué pour BLOCK GRAPH EDITION et BLOCK GRAPH DELETION et donc nous ne pouvons pas réduire directement la taille des cliques critiques à $k + 1$. Pour remédier à ce problème, nous formulons une règle de réduction légèrement plus faible.

Règle 5.2. Soit $K \subseteq V$ une clique critique de G telle que $|K| > 2k + 2$. Retirer $|K| - (2k + 2)$ sommets de K .

Lemme 5.8. La [règle 5.2](#) est sûre.

Preuve. Soit G' le graphe obtenu en enlevant $|K| - (2k + 2)$ sommets de K dans G et $K' = V(G) \setminus V(G')$. Supposons d'abord que G admette une k -édition F . Observons que F est une k -édition de G' par hérédité. Inversement, supposons que G' admet une k -édition F et définissons $H' = G' \triangle F$. Remarquons que puisque F affecte au plus $2k$ sommets et que $|K \setminus K'| = 2k + 2$, il existe deux sommets non affectés u et v dans $K \setminus K'$. Nous affirmons que $H = G \triangle F'$ est un graphe bloc. Supposons par contradiction que H contient une obstruction W . Par construction, W doit intersecter K' car sinon H' contiendrait une obstruction. De plus, remarquons que $|W \cap K'| \leq 2$, sinon W contiendrait un module clique avec 3 ou 4 sommets, ce qui est impossible dans un cycle ou un diamant. Supposons donc d'abord que $|W \cap K'| = \{a\}$: dans ce cas, puisque u n'est pas affecté, nous avons $N_H[a] \setminus K' = N_{H'}[u]$ et donc l'ensemble $(W \setminus \{a\}) \cup \{u\}$ induit une obstruction dans H' , ce qui est une contradiction. De même, si $|W \cap K'| = \{a, b\}$ l'ensemble $(W \setminus \{a, b\}) \cup \{u, v\}$ induit une obstruction dans H' , ce qui implique à nouveau à une contradiction. \square

5.3.2 Réduction des BG-branches

Nous définissons la structure principale que va réduire notre noyau, à savoir les *BG-branches*. Elles correspondent à des parties déjà blocs du graphe. Notons que nous reprenons la notion de « branche », ainsi que les notations introduites par Bessy *et al.* [13] pour les graphes 3-leaf power.

Définition 5.3 (BG-branche). Soit $G = (V, E)$ un graphe, une *BG-branche* de G est un sous-graphe induit connexe B de G tel que B est un graphe bloc.

Un sommet $v \in V(B)$ est un *point d'attache* d'une BG-branche B si $N_G(v) \setminus V(B) \neq \emptyset$. Une BG-branche est une p -BG-branche si elle a exactement p points d'attache. Nous désignons par B^R le sous-graphe d'une BG-branche B dans lequel tous les sommets des points d'attache ont été retirés. Nous définissons une première règle simple sur les 1-BG-branches qui permet de retirer tous leurs sommets à l'exception des points d'attaches.

Règle 5.3. Soit B une 1-BG-branche de G telle que B^R est connexe et non vide. Retirer $V(B^R)$.

Lemme 5.9. La [règle 5.3](#) est sûre.

Preuve. Soit B une telle 1-BG-branche et $G' = G \setminus V(B^R)$. Soit F une k -édition de G en un graphe bloc. Puisque la classe des graphes blocs est héréditaire F est une k -édition de G' .

Réciproquement, supposons que F' soit une k -édition de G' . Soit u le point d'attache de B dans G . Observons que puisque B^R est connexe et non vide, u n'est pas un sommet séparateur du graphe bloc B et son voisinage dans B est soit une clique maximale, soit un seul sommet. Puisque $H' = G' \triangle F'$ et B sont des graphes blocs et que $N_B(u)$ est un sommet unique ou une clique maximale de B , le [lemme 5.2](#) garantit que $H = (H', \{u\}) \otimes (B^R, N_B(u))$ est un graphe bloc. Observons que $H = G \triangle F'$, donc F' est une k -édition de G . \square

Nous traitons maintenant les 2-BG-branches qui demandent plus de technicité. Une 2-BG-branche B est *propre* si B^R est connexe et si les deux points d'attache ne sont pas adjacents. Pour une 2-BG-branche propre B , une *coupe minimum* de B est un ensemble d'arêtes $M \subseteq E(B)$ de taille minimum tel que les points d'attache p_1 et p_2 ne sont pas dans la même composante connexe de $B - M$. Nous notons par $\text{mc}(B)$ la taille d'une coupe minimum de B .

Lemme 5.10. *Soit (G, k) une instance positive de BLOCK GRAPH EDITION réduite par la règle 5.3, et B une 2-BG-branche propre de G . Si $|V(B)| \geq k + 3$ alors il existe une k -édition F de G telle que l'ensemble des paires de F contenant un sommet de $V(B^R)$ est soit vide soit exactement une coupe minimum de B .*

Preuve. Soit B une 2-BG-branche propre de G avec les points d'attache p_1 et p_2 telle que $|V(B)| \geq k + 3$. Observons que $N_B(p_1)$ et $N_B(p_2)$ sont des cliques maximales de B^R ou des sommets car B^R est connexe et que p_1 et p_2 ne sont pas adjacents dans B puisqu'elle est propre. Puisque B est réduite par la règle 5.3, observons que les sommets séparateurs de B sont contenus dans tout chemin π de p_1 à p_2 . En effet, s'il existe un sommet séparateur x qui n'est pas dans π , alors p_1 et p_2 sont dans la même composante connexe de $B \setminus \{x\}$, ce qui implique qu'il existe une composante connexe K de $B \setminus \{x\}$ ne contenant ni p_1 ni p_2 . Puisque B est une 2-BG-branche, $N_G(K) = \{x\}$, donc $G[V(K) \cup \{x\}]$ induit une 1-BG-branche de G avec le point d'attache x , ce qui est une contradiction. Remarquons en particulier qu'il existe un unique plus court chemin π_s entre p_1 et p_2 qui contient tous les sommets séparateurs de B (illustré dans la figure 5.4). De plus, puisque les seuls sommets séparateurs de B sont ceux de π_s , un sommet de B qui n'est pas dans π_s se trouve dans exactement une clique maximale de B contenant deux sommets consécutifs de π_s .

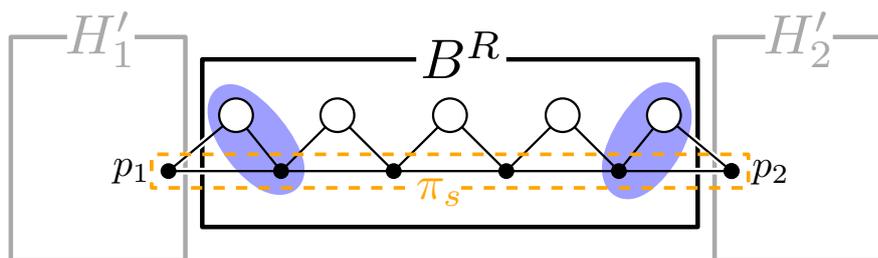


Figure 5.4: Illustration du cas où p_1 et p_2 ne sont pas dans la même composante connexe de H' dans le lemme 5.10. Les ronds blancs correspondent à des cliques critiques de B . Les ensembles bleus correspondent à $N_B(p_1)$ et $N_B(p_2)$.

Soit F une k -édition de G , et soit $H = G \Delta F$ et $H' = H \setminus V(B^R)$. Considérons d'abord le cas où p_1 et p_2 sont dans deux composantes connexes différentes de H' . Soit H'_1, H'_2 les deux composantes connexes de H' contenant p_1 et p_2 , respectivement, et H'_3 le reste de H' . Le lemme 5.2 implique que le graphe H^* résultant de l'union disjointe du graphe

$$(((B^R, N_B(p_1)) \otimes (H'_1, \{p_1\})), N_B(p_2)) \otimes (H'_2, \{p_2\})$$

et de H'_3 est un graphe bloc (illustré dans la figure 5.4). Soit F^* l'édition telle que $H^* = G \Delta F^*$. Par construction, F^* est un sous-ensemble de F dont aucune paire ne contient un sommet de $V(B^R)$.

Supposons maintenant que p_1 et p_2 se trouvent dans la même composante connexe de H' et définissons $\pi_{H'}$ le plus court chemin entre eux. Notons que ce chemin peut être de longueur 1 si p_1 et p_2 sont adjacents dans H' . Nous considérons d'abord le cas où il existe un chemin $\pi_B = \{p_1 = u_1, u_2, \dots, u_r = p_2\}$ (pas nécessairement induit) de p_1 à p_2 dans B qui existe encore dans $B \Delta F$. Puisque B est une 2-BG-branche propre de G réduite par la règle 5.3, nous savons par les arguments précédents que chaque sommet séparateur de B est dans π_B . Les chemins $\pi_{H'}$ et π_B ne s'intersectent que en leurs extrémités p_1 et p_2 et forment donc un cycle dans H . Nous en déduisons que tous les sommets dans π_B appartiennent à la même composante biconnexe de H , qui est une clique. Nous pouvons donc considérer que π_B est le chemin contenant exactement p_1, p_2 et les sommets séparateurs de B , qui est le plus court chemin dans B entre p_1 et p_2 . Puisque

les sommets de π_B induisent un chemin dans B (sur au moins 3 sommets puisque p_1 et p_2 ne sont pas adjacents) et une clique dans $B \triangle F$, alors pour tout $i \in \{1, \dots, r-2\}$, nous avons $\{u_i, u_r\} \in F$. Tout sommet $v \in B \setminus V(\pi_B)$ est adjacent dans B à deux sommets $w_1, w_2 \in V(\pi_B)$ et si v reste dans la même composante biconnexe que w_1 et w_2 dans H , alors $\{v, p_1\} \in F$ ou $\{v, p_2\} \in F$. Autrement F doit contenir $\{v, w_1\}$ ou $\{v, w_2\}$. Dans tous les cas, F contient au moins une paire pour chaque sommet $v \in B \setminus \{u_{r-1}, u_r\}$. Il résulte des arguments précédents que $|F| \geq |B| - 2$. Cependant, puisque $|B| \geq k + 3$ nous avons que $|F| \geq k + 1$, ce qui contredit le fait que F est une k -édition, il n'existe donc pas de tel chemin π_B .

Nous pouvons supposer que si p_1 et p_2 sont dans la même composante connexe de H' , alors F contient une coupe de B . En effet si ce n'était pas le cas, il y aurait un chemin π_B de p_1 à p_2 dans B qui existerait encore dans $B \triangle F$. Soit M une coupe minimum de B et considérons B_1 (resp. B_2) la composante connexe contenant p_1 (resp. p_2) dans $B \triangle M$. Les sous-graphes $B_1^R = B_1 \setminus \{p_1\}$ et $B_2^R = B_2 \setminus \{p_2\}$ sont des graphes blocs par hérédité (l'un d'entre eux peut être vide). Par le [lemme 5.2](#), le graphe suivant est bloc (illustré dans la [figure 5.5](#)) :

$$H^* = ((H', \{p_1\}) \otimes (B_1^R, N_{B_1}(p_1)), \{p_2\}) \otimes (B_2^R, N_{B_2}(p_2))$$

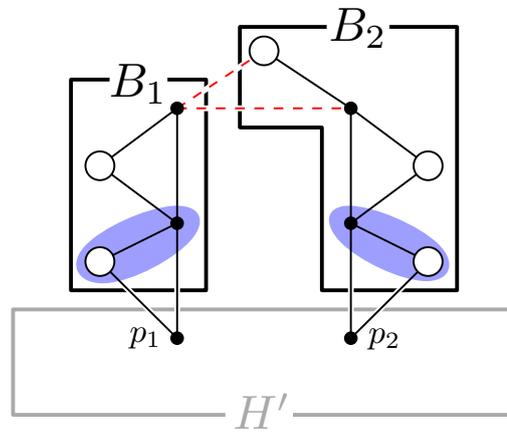


Figure 5.5: Illustration du cas où p_1 et p_2 sont dans la même composante connexe de H' et F contient une coupe de B dans le [lemme 5.10](#). Les ronds blancs correspondent à des cliques critiques de B . Les ensembles bleus correspondent à $N_B(p_1)$ et $N_B(p_2)$. Les arêtes en pointillés rouges entre B_1 et B_2 sont des arêtes d'une coupe de B supprimée par l'édition.

Soit F^* l'édition telle que $H^* = G \triangle F^*$, nous avons $F^* = (F \setminus (V(B^R) \times V(G))) \cup M$, puisque M est une coupe minimum de B , $|F^*| \leq |F|$ et donc F^* est une k -édition de G .

Dans tous les cas, il existe une k -édition F^* de G en un graphe bloc telle que l'ensemble des paires de F^* contenant un sommet de $V(B^R)$ est soit vide, soit exactement une coupe minimum de B . \square

À l'aide du lemme que nous venons de montrer, nous pouvons montrer aisément la sûreté de la règle de réduction suivante.

Règle 5.4. Soit B une 2-BG-branche propre de G contenant au moins $k + 3$ sommets avec les points d'attache p_1 et p_2 . Retirer $V(B^R)$ de G et ajouter un sommet x adjacent à p_1 et p_2 , ajouter une clique de taille $\min\{\text{mc}(B) - 1, k\}$ adjacente à p_1 et x , et ajouter une clique de taille k adjacente à p_2 et x .

Lemme 5.11. La [règle 5.4](#) est sûre.

Preuve. Soit G' le graphe obtenu par application de la [règle 5.4](#) sur la 2-BG-branche propre B du graphe G avec les points d'attache p_1 et p_2 . Soit S les sommets introduits par cette règle. Observons que $B' = G'[S \cup \{p_1\} \cup \{p_2\}]$ est une 2-BG-branche propre de G' , $|V(B')| \geq k + 3$ et $\text{mc}(B') = \min\{\text{mc}(B), k + 1\}$.

Supposons qu'il existe une k -édition F de G , choisissons-en une qui satisfait le [lemme 5.10](#) et définissons $H = G \triangle F$. Si F contient une coupe minimum de B , observons que $\text{mc}(B') = \text{mc}(B) \leq k$. Considérons B'_i , $i \in \{1, 2\}$ la composante connexe de $B' \triangle F_m$ contenant p_i , où F_m est une coupe minimum de B' (remarquons qu'une des deux composantes B'_i peut être vide). Par le [lemme 5.2](#) le graphe

$$H' = ((H \setminus V(B^R), p_1) \otimes (B'_1, N_{B'}(p_1)), p_2) \otimes (B'_2, N_{B'}(p_2))$$

est un graphe bloc. Soit F' l'édition telle que $H' = G' \triangle F'$. Puisque $\text{mc}(B') = \text{mc}(B) \leq k$, par construction $|F'| = |F|$, ainsi F' est une k -édition de G' . Si F ne contient pas de coupe minimum de B , alors par le [lemme 5.10](#) aucun sommet de B^R n'est affecté. De plus la preuve de ce lemme montre également que dans ce cas p_1, p_2 ne sont pas dans la même composante connexe de $H \setminus V(B^R)$. Soit H_1 et H_2 les composantes connexes de $H \setminus V(B^R)$ qui contiennent respectivement p_1 et p_2 et H_3 les composantes connexes restantes de $H \setminus V(B^R)$. Par le [lemme 5.2](#) le graphe H' correspondant à l'union disjointe du graphe

$$(((B'^R, N_{B'}(p_1)) \otimes (H_1, \{p_1\}), N_{B'}(p_2)) \otimes (H_2, \{p_2\}))$$

et de H_3 est un graphe bloc. Observons que $H' = G' \triangle F$, donc F est une k -édition de G' .

Des arguments symétriques permettent de montrer que toute k -édition de G' permet de construire une k -édition de G . \square

5.3.3 Borner la taille d'une instance réduite

Nous montrons dans cette section que le problème BLOCK GRAPH EDITION admet un noyau avec $O(k^2)$ sommets. Nous commençons par montrer que les règles impliquant des BG-branches peuvent être appliquées en temps polynomial.

Lemme 5.12. *Les règles 5.3 et 5.4 peuvent être appliquées exhaustivement en temps polynomial.*

Preuve. Nous montrons que nous pouvons énumérer toutes les 1-BG-branches et 2-BG-branches en temps polynomial. La reconnaissance des graphes blocs peut être faite en temps linéaire, en effet, les composantes biconnexes peuvent être énumérées en temps linéaire [123]. Toutes les 1-BG-branches peuvent être énumérées en supprimant un sommet $v \in V(G)$ et en recherchant parmi les composantes connexes restantes celles qui induisent avec v un graphe bloc connexe. Nous procédons de la même manière pour détecter les 2-BG-branches propres en supprimant une paire de sommets non adjacents $u, v \in V(G)$ et en recherchant parmi les composantes connexes restantes celles qui induisent avec u et v un graphe bloc connexe. \square

Nous montrons maintenant que toute instance positive de BLOCK GRAPH EDITION réduite par les [règles 5.1 à 5.4](#) contient $O(k^2)$ sommets.

Théorème 5.2. *BLOCK GRAPH EDITION admet un noyau avec $O(k^2)$ sommets.*

Preuve. Soit $(G = (V, E), k)$ une instance positive de BLOCK GRAPH EDITION réduite par les [règles 5.1 à 5.4](#). Soit F une k -édition de G et $H = G \triangle F$. Nous supposons que G est connexe, les arguments qui suivent peuvent être facilement adaptés si ce n'est pas le cas en sommant sur toutes les composantes connexes de G . Soit A l'ensemble des sommets affectés de H . Puisque $|F| \leq k$, nous avons $|A| \leq 2k$. Soit T le sous-graphe induit minimal connexe de H qui couvre tous les sommets de A , et soit A' l'ensemble des sommets de degré au moins 3 dans T . Par le [lemme 5.4](#), $|A' \setminus A| \leq 3 \cdot |A| \leq 6k$.

Tout d'abord, nous pouvons observer que s'il existe un sommet séparateur $x \in V(H)$ qui n'est pas dans $V(T)$, puisque T est le sous-graphe induit minimal connexe de H qui couvre tous les sommets affectés, il existe une composante connexe K de $H \setminus \{x\}$ qui ne contient aucun sommet affecté. Puisque $B = H[V(K) \cup \{x\}]$ est un graphe bloc par hérédité, il en résulte que B induit une 1-BG-branche avec un point d'attache x et peut être réduit par la [règle 5.3](#), ce qui est une contradiction. Ainsi chaque sommet séparateur de H est contenu dans $V(T)$ et chaque sommet de $V(H) \setminus V(T)$ est adjacent à une clique maximale ou à un sommet de $V(T)$. Puisque le graphe est

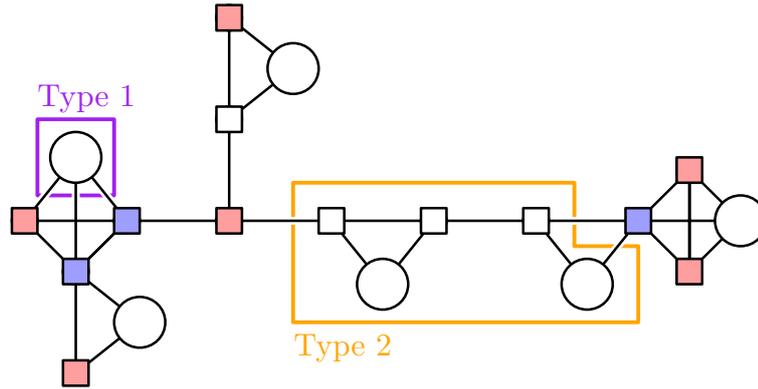


Figure 5.6: Illustration d'une instance réduite dans la preuve du [théorème 5.2](#). Les nœuds carrés correspondent aux sommets de T , ceux en rouge sont des sommets de A , ceux en bleu sont des sommets de $A' \setminus A$. Les nœuds circulaires correspondent aux cliques critiques de G .

réduit par la [règle 5.3](#), le deuxième cas ne peut pas se produire, sinon G contiendrait une 1-BG-branche. Par conséquent, il existe deux types de composantes connexes dans $H \setminus (A \cup A')$, à savoir celles qui sont adjacentes à une clique maximale de sommets de $A \cup A'$ et celles qui sont adjacentes à deux sommets non adjacents de $A \cup A'$. Ces dernières sont celles qui contiennent des composantes connexes de $T \setminus (A \cup A')$. Nous disons que ces composantes connexes sont respectivement de type 1 ou de type 2 (illustrées dans la [figure 5.6](#)).

Les composantes connexes de type 1 sont des cliques puisque H est un graphe bloc. De plus elles ont le même voisinage dans H , ce sont donc des cliques critiques. Elles sont réduites par la [règle 5.2](#) et contiennent au plus $2k + 2$ sommets. Puisque T est un graphe bloc, il contient au plus $|A \cup A'| \leq 8k$ cliques maximales de sommets de $A \cup A'$, et contient donc au plus $8k \cdot (2k + 2)$ sommets dans des composantes connexes de type 1.

Considérons maintenant les composantes connexes de type 2. Nous pouvons observer qu'une telle composante K avec ses sommets adjacents $x, y \in A \cup A'$ forment une 2-BG-branche propre, et contient donc au plus $2k + 3$ sommets par la [règle 5.4](#). D'après le [lemme 5.4](#), $T \setminus (A \cup A')$ contient au plus $2 \cdot |A| = 4k$ composantes connexes ce qui est exactement le nombre de composantes connexes de type 2. Par conséquent ces dernières contiennent au plus $4k \cdot (2k + 1)$ sommets.

Nous concluons que $|V(G)| = O(k^2)$. Pour finir, une instance peut être réduite exhaustivement en temps polynomial. En effet, toutes les règles de réduction peuvent être appliquées exhaustivement en temps polynomial d'après les [lemmes 3.5](#) et [5.12](#). \square

Nous avons présenté un noyau pour BLOCK GRAPH DELETION. Pour le cas de BLOCK GRAPH DELETION, toutes les règles présentées restent sûres. En particulier, la [règle 5.4](#) reste sûre en remplaçant simplement dans la preuve « édition » par « délétion ».

Théorème 5.3. *BLOCK GRAPH DELETION admet un noyau avec $O(k^2)$ sommets.*

5.4 Noyau pour STRICTLY CHORDAL EDITION

Nous présentons dans cette section un noyau pour le problème STRICTLY CHORDAL EDITION qui contient $O(k^4)$ sommets. Nous montrerons ensuite que pour les versions complétion et délétion du problème, un noyau avec $O(k^3)$ sommets peut être obtenu en renforçant une des règles de réduction.

Nous commençons en donnant une description informelle du noyau pour STRICTLY CHORDAL EDITION. Considérons une instance positive $(G = (V, E), k)$ de STRICTLY CHORDAL EDITION, F une k -édition de G et soit $H = G \triangle F$. Nous utilisons le graphe des cliques critiques $\mathcal{C}(H)$ de H pour borner le nombre de sommets d'une instance réduite. Rappelons que le graphe $\mathcal{C}(H)$ est

un graphe bloc d'après le [théorème 5.1](#). L'approche pour borner la taille du noyau est similaire à celle utilisée pour les graphes blocs, les règles dans le cas des graphes strictement cordaux étant cependant plus complexes.

Puisque F est de taille au plus k , il existe au plus $2k$ cliques critiques de $\mathcal{C}(H)$ qui contiennent des sommets affectés. Soit A ces cliques critiques et T le sous-graphe induit connexe minimal de $\mathcal{C}(H)$ qui couvre les cliques critiques de A . Définissons A' comme l'ensemble des cliques critiques de degré au moins 3 dans T . D'après le [lemme 5.4](#), nous avons $|A' \setminus A| \leq 3 \cdot |A|$.

Nous définirons la notion de SC-branche, correspondant à des sous-graphes strictement cordaux G , avec des contraintes supplémentaires sur leur structure. Nous nous intéresserons à deux types de SC-branches : celles qui sont connectées au reste du graphe par une seule clique critique, appelées 1-SC-branches, et celles qui sont connectées au reste du graphe par exactement deux cliques critiques, appelées 2-SC-branches. Cette fois-ci, les 1-SC-branches ne pourront être réduites qu'à une paire de cliques critiques. Les composantes connexes du graphe $T \setminus (A \cup A')$ correspondent à des sous-graphes de 2-SC-branches et la longueur de ces 2-SC-branches sera bornée linéairement en k . Il y a au plus $4k$ telles composantes connexes, et donc T contient $O(k^2)$ cliques critiques.

Enfin, les composantes connexes du graphe $\mathcal{C}(H) \setminus V(T)$ correspondent à des 1-SC-branches ou à des ensembles de 1-SC-branches. Ce dernier cas nécessitera une règle de réduction spécifique qui bornera linéairement le nombre de 1-SC-branches ayant le même voisinage. Ainsi, chaque clique critique ou clique maximale de T peut avoir un nombre linéaire de 1-SC-branches de $\mathcal{C}(H) \setminus V(T)$ qui lui sont adjacentes. Nous en déduisons que le graphe $\mathcal{C}(H)$ contient $O(k^3)$ cliques critiques. De plus, puisque les graphes strictement cordaux sont clos pour l'ajout de vrais jumeaux, la taille de chaque clique critique peut être bornée par $k + 1$, nous en concluons que le graphe G contient $O(k^4)$ sommets.

Remarque 5.2. Dans le reste de la section nous supposons que nous avons en entrée une instance $(G = (V, E), k)$ de STRICTLY CHORDAL EDITION.

Les graphes strictement cordaux sont clos par union disjointe et contrairement aux graphes blocs, les graphes strictement cordaux sont clos pour l'ajout de vrais jumeaux, nous pouvons donc définir les [règles 5.5](#) et [5.6](#) qui sont sûres d'après les [lemmes 3.1](#) et [3.3](#).

Règle 5.5. Soit C une composante connexe de G telle que C est strictement cordal. Retirer les sommets de C .

Règle 5.6. Soit $K \subseteq V$ une clique critique de G telle que $|K| > k + 1$. Retirer $|K| - (k + 1)$ sommets de K .

5.4.1 SC-branches : définition et propriétés structurelles

Nous définissons les *SC-branches* de manière similaire aux BG-branches, mais cette fois ci leur structure sera plus contrainte.

Définition 5.4 (SC-branche). Soit $G = (V, E)$ un graphe. Une *SC-branche* de G est un sous-graphe induit connexe B de G où $V(B)$ est une union de cliques critiques $K_1, \dots, K_r \in \mathcal{K}(G)$ telle que le sous-graphe de $\mathcal{C}(G)$ induit par K_1, \dots, K_r est un graphe bloc connexe.

Nous insistons sur le fait que notre définition d'une SC-branche B est plus forte que simplement demander que B soit un sous-graphe induit strictement cordal. Par exemple, si G est un dard, le sous-graphe obtenu en supprimant le sommet pendant est strictement cordal, mais ce n'est pas une SC-branche car les cliques critiques correspondantes ne forment pas un graphe bloc dans $\mathcal{C}(G)$. Soit B une SC-branche du graphe G et soit K_1, \dots, K_r les cliques critiques de G contenues dans $V(B)$. Nous disons que K_i est un *point d'attache* de B si $N_G(K_i) \setminus V(B) \neq \emptyset$. Une SC-branche B est une p -SC-branche si elle a exactement p points d'attache. Nous notons B^R le sous-graphe de B dans lequel tous les points d'attache ont été supprimés.

Nous montrons maintenant deux lemmes qui seront essentiels pour prouver la sûreté des règles de réduction portant sur les SC-branches. Nous commençons par donner des propriétés structurelles sur les voisinages des points d'attaches dans les SC-branches. Ce lemme nous permettra notamment de réaliser des opérations de jointure entre des SC-branches et des graphes édités. Pour un exemple, la [figure 5.7](#) illustre la structure d'un voisinage d'une 1-SC-branche.

Lemme 5.13. *Soit $G = (V, E)$ un graphe et B une SC-branche de G . Pour un point d'attache P de B , notons $B' = B \setminus P$. Considérons les composantes connexes G_1, G_2, \dots, G_r de B' et notons $Q_i = N_B(P) \cap V(G_i)$. Pour chaque $1 \leq i \leq r$, Q_i est une clique critique, une clique maximale ou intersecte exactement une clique maximale de G_i .*

Preuve. Tout d'abord, nous montrons que tous les ensembles Q_i sont des cliques. Supposons que Q_i n'est pas une clique pour $1 \leq i \leq r$. Soit x et y des sommets non adjacents de Q_i et $z \in P$. Puisque G_i est connexe, prenons le plus court chemin π entre x et y dans G_i . Le sous-graphe induit par les sommets $\{x, y, z\}$ et ceux de π contient soit un cycle de longueur au moins égale à 4 si z n'est adjacent à aucun sommet interne de π , ce qui est une contradiction puisque B est un graphe bloc, soit un diamant dont z est l'un des sommets de degré 3. Dans ce dernier cas, puisque z n'est pas dans la même clique critique de $\mathcal{C}(G)$ que son vrai jumeau dans le diamant, les cliques critiques de $\mathcal{C}(G)$ qui contiennent les sommets de ce diamant forment aussi un diamant dans $\mathcal{C}(G)$. Un tel diamant est formé par les cliques critiques de G contenues dans B , ce qui contredit la définition d'une SC-branche. Dans tous les cas nous avons une contradiction, ainsi Q_i est une clique dans G .

Maintenant supposons par contradiction que Q_i , $1 \leq i \leq r$ n'est ni une clique critique ni une clique maximale de G_i et qu'elle intersecte au moins deux cliques maximales de G_i . Supposons d'abord que Q_i n'est pas un module, dans ce cas il existe des sommets $u, v \in Q_i$ et un sommet $x \in V(G_i) \setminus Q_i$ adjacent à seulement u ou v . Puisque Q_i n'est pas une clique maximale de G_i , il existe $y \in V(G_i) \setminus Q_i$ adjacent à u et v . Avec un sommet $w \in P$, les sommets $\{u, v, w, x, y\}$ induisent dans B une gemme si x et y sont adjacents et un dard dans le cas contraire, ce qui est une contradiction. Nous pouvons donc supposer que Q_i est un module. Puisque Q_i n'est pas une clique critique (c'est-à-dire un module clique maximal), il existe des sommets $u \in Q_i$ et $v \in V(G_i) \setminus Q_i$ qui sont dans la même clique critique de G_i . Puisque Q_i intersecte deux cliques maximales de G_i , il existe deux sommets non adjacents $x, y \in V(G_i) \setminus Q_i$ qui sont adjacents à la fois à u et v . Avec un sommet $w \in P$, les sommets $\{u, v, w, x, y\}$ induisent un dard dans B , ce qui est une contradiction. Nous pouvons donc conclure que Q_i est une clique critique, une clique maximale ou intersecte exactement une clique maximale de G_i . \square

Nous montrons à présent qu'un point d'attache d'une SC-branche reste un module clique (pas nécessairement maximal) dans une édition optimale d'un graphe. Nous rappelons que toutes les éditions que nous considérons dans cette section sont homogènes au sens du [lemme 3.2](#), c'est à dire qu'elles affectent de manière uniforme les sommets d'une même clique critique.

Lemme 5.14. *Soit $G = (V, E)$ un graphe et B une SC-branche de G . Soit F une édition optimale de G , et $H = G \Delta F$. Pour un point d'attache P de B , soit C la clique critique de H qui contient P et $C' = C \setminus V(B^R)$. Si $N_H(C') \cap V(B^R) \neq \emptyset$, alors C' est une clique critique ou intersecte exactement une clique maximale de $H' = H \setminus V(B^R)$.*

Preuve. Supposons que C' ne soit pas une clique critique de H' et intersecte au moins deux cliques maximales de H' . Puisque $C' \subseteq C$ est un module mais pas une clique critique de H' , ce n'est pas un module clique maximal et donc il existe $u \in C'$ et $v \notin C'$ qui se trouvent dans la même clique critique dans H' . Remarquons qu'il doit exister un sommet $w \in N_H(C') \cap V(B^R)$ adjacent à exactement un des sommets u et v dans H , sinon nous aurions $N_H(u) \cap V(B) = N_H(v) \cap V(B)$ et $N_{H'}(u) = N_{H'}(v)$, ce qui signifierait que C n'est pas une clique critique de H , ce qui est une contradiction. Supposons sans perte de généralité que w soit adjacent à u et non adjacent à v . Puisque C' intersecte deux cliques maximales de H' , il existe deux sommets non adjacents $x, y \in V(H')$ qui sont adjacents à la fois à u et à v . Les sommets $\{u, v, w, x, y\}$ induisent un dard dans H si w n'est adjacent ni à x ni à y , une gemme s'il est adjacent à l'un d'eux, et si w est adjacent à la fois à x et à y , $\{w, x, v, y\}$ induit un C_4 dans H . Dans tous les cas, nous sommes dans une contradiction. Nous concluons que C' est une clique critique de H' ou que C' est inclus dans exactement une clique maximale de H' si $N_H(C') \cap V(B^R) \neq \emptyset$. \square

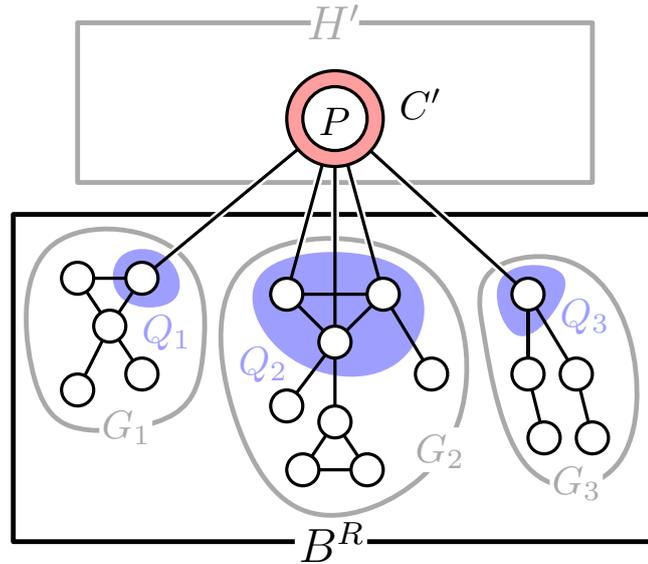


Figure 5.7: Illustration du graphe $H^* = (\bigcup_{1 \leq i \leq r} G_i, \bigcup_{1 \leq i \leq r} Q_i) \otimes (H', C')$ dans la preuve du lemme 5.15 pour $r = 3$.

5.4.2 Réduction des 1-SC-branches

Nous montrons deux règles de réduction sur les 1-SC-branches, la première impliquera que toutes les 1-SC-branches sont uniquement constituées d'un point d'attache et d'une clique adjacente à ce dernier. La seconde règle permet de borner le nombre de 1-SC-branches qui partagent un même voisinage. Nous commençons par montrer qu'il existe toujours une édition optimale qui n'affecte que le point d'attache d'une 1-SC-branche et son voisinage.

Lemme 5.15. *Soit $G = (V, E)$ un graphe et B une 1-SC-branche de G avec le point d'attache P . Il existe une édition optimale F de G telle que :*

- l'ensemble de sommets de B affectés par F est inclus dans $P \cup N_B(P)$,
- dans $H = G \triangle F$ les sommets de $N_B(P)$ sont tous adjacents aux mêmes sommets de $V(G) \setminus V(B^R)$.

Preuve. Soit F une édition optimale de G et définissons $H = G \triangle F$. Soit G_1, G_2, \dots, G_r les composantes connexes de B^R et notons $Q_i = N_B(P) \cap V(G_i)$. Les graphes $H' = H \setminus V(B^R)$ et G_1, \dots, G_r sont strictement cordaux par hérédité. Ils sont illustrés dans la figure 5.7.

Par le lemme 5.13, Q_i est une clique critique, une clique maximale ou intersecte exactement une clique maximale de G_i . Nous définissons x comme un sommet de $Q_1 \cup \dots \cup Q_r$ contenu dans un nombre minimum de paires $\{x, b\} \in F$ avec $b \in V(G) \setminus V(B^R)$, et notons $N = N_H(x) \setminus V(B^R)$. Observons que puisque P est une clique critique et que F est homogène, $P \cap N = \emptyset$ ou $P \subseteq N$. Si $P \subseteq N$, supposons que C est la clique critique de H qui contient P , prenons $C' = C \setminus V(B^R)$ et observons que $P \subseteq C' \subseteq N$. Par le lemme 5.14, C' est une clique critique ou intersecte exactement une clique maximale de H' . Par le lemme 5.3 et l'observation 5.1, le graphe

$$H^* = \left(\bigcup_{1 \leq i \leq r} G_i, \bigcup_{1 \leq i \leq r} Q_i \right) \otimes (H', C')$$

est strictement cordal. Si $P \cap N = \emptyset$, notons H^* le graphe strictement cordal correspondant à l'union disjointe de H' et des graphes G_1, \dots, G_r . Soit F^* l'édition telle que $H^* = G \triangle F^*$, dans les deux cas par construction $|F^*| \leq |F|$ et les propriétés souhaitées sont vérifiées. \square

Le lemme 5.15 nous permet de déduire la règle de réduction qui suit.

Règle 5.7. Soit B une 1-SC-branche de G avec le point d'attache P . Retirer les sommets de B^R puis ajouter une clique de taille $\min\{|N_B(P)|, k + 1\}$ adjacente à P .

Lemme 5.16. *La règle 5.7 est sûre.*

Preuve. Soit G' le graphe obtenu par l'application de la règle 5.7 sur la 1-SC-branche B . Soit K la clique qui remplace B^R dans G' . Commençons par supposer qu'il existe une k -édition F de G et soit $H = G \triangle F$. Par le lemme 5.15 nous pouvons supposer que F n'affecte que les sommets de $P \cup N_B(P)$ et que tous les sommets de $N_B(P)$ sont adjacents aux mêmes sommets de $V(G) \setminus V(B^R)$. Ceci implique que si $|N_B(P)| > k$ aucune k -édition ne peut affecter ces sommets. Soit C la clique critique de H qui contient P et soit $C' = C \setminus V(B^R)$. Par le lemme 5.14, C' est une clique critique ou intersecte exactement une clique maximale de $H_a = H \setminus V(B^R)$. Si $N_B(P)$ est adjacent à C' dans H , considérons le graphe $H' = (H_a, C') \otimes (K, K)$. Puisque K est une clique, par le lemme 5.3, H' est strictement cordal. Sinon, considérons le graphe strictement cordal H' correspondant à l'union disjointe de H_a et K . Soit F' l'édition telle que $H' = G' \triangle F'$. Puisque $|K| = \min\{|N_B(P)|, k+1\}$, par construction nous avons $|F| = |F'|$, donc F' est une k -édition de G' .

L'autre sens est similaire, mais ici B^R peut contenir plusieurs composantes connexes et donc appliquer l'opération de jointure avec B^R demande plus de détails. Supposons qu'il existe une k -édition F' de G' et soit $H' = G' \triangle F'$. Par le lemme 5.15, nous pouvons supposer que F' n'affecte que les sommets de $P \cup K$ et que tous les sommets de K sont adjacents aux mêmes sommets de $V(G') \setminus K$. Ceci implique que si $|K| > k$, aucune k -édition ne peut affecter ces sommets. Soit C la clique critique de H' qui contient P et soit $C' = C \setminus K$. Par le lemme 5.14, C' est une clique critique ou intersecte exactement une clique maximale de $H'_a = H' \setminus K$. Soit G_1, G_2, \dots, G_r les composantes connexes de B^R et notons $Q_i = N_B(P) \cap V(G_i)$. Par le lemme 5.13, Q_i est une clique critique, une clique maximale ou intersecte exactement une clique maximale de G_i .

Si la clique K est adjacente à C' dans H' , considérons le graphe :

$$H = (H'_a, C') \otimes \left(\bigcup_{1 \leq i \leq r} G_i, \bigcup_{1 \leq i \leq r} Q_i \right)$$

Par le lemme 5.3 et l'observation 5.1, H' est strictement cordal. Sinon, considérons le graphe strictement cordal H correspondant à l'union disjointe de H'_a et G_i pour $1 \leq i \leq r$. Soit F l'édition telle que $H = G \triangle F$, puisque $|K| = \min\{|N_B(P)|, k+1\}$, par construction nous avons $|F| = |F'|$, donc F est une k -édition de G . \square

Nous montrons maintenant le lemme qui permet de définir la règle bornant le nombre de 1-SC-branches partageant un même voisinage. Notamment, si suffisamment de 1-SC-branches partagent un voisinage N , elles forcent que toute k -édition transforme N en une clique.

Lemme 5.17. *Soit $(G = (V, E), k)$ une instance positive de STRICTLY CHORDAL EDITION réduite par la règle 5.6. Soit B_1, \dots, B_l des 1-SC-branches disjointes de G avec des points d'attache P_1, \dots, P_l qui partagent le même voisinage N dans $G \setminus \bigcup_{i=1}^l V(B_i)$ et forment une union disjointe de cliques Q_1, \dots, Q_r dans $G[P_1 \cup \dots \cup P_l]$. Pour toute k -édition F de G , définissons $H = G \triangle F$ et $H' = H \setminus \bigcup_{i=1}^l V(B_i)$. Si $\sum_{i=1}^l |P_i| > 2k+1$ alors :*

- (i) *Si $r = 1$ alors N est une clique de H et N est une clique critique, une clique maximale ou intersecte exactement une clique maximale de H' ,*
- (ii) *Si $r > 1$ et $(\sum_{i=1}^l |P_i|) - \max_{1 \leq j \leq r} \{|Q_j|\} > k$ alors N est une clique critique de H et N est une clique critique ou intersecte exactement une clique maximale de H' .*

Preuve. Soit F une k -édition de G et soit les graphes $H = G \triangle F$ et $H' = H \setminus \bigcup_{i=1}^l V(B_i)$. Nous commençons par montrer le cas (i) puis nous montrerons le cas (ii).

Cas (i) : $r = 1$. Nous commençons par prouver que N doit être une clique dans H . Supposons que N n'est pas une clique dans H , il existe alors deux sommets $x, y \in N$ tels que $\{x, y\} \notin E(H)$. Les points d'attache sont des cliques critiques, donc par la règle 5.6, $|P_i| \leq k+1$, $1 \leq i \leq l$. Nous pouvons observer qu'il existe au plus une clique critique P_j telle que $N_{B_j}(P_j) = \emptyset$ (rappelons que $r = 1$ et donc que tous les points d'attache sont contenus dans la même clique de $G[P_1 \cup \dots \cup P_l]$). Ainsi $|Q_1 \setminus P_j| > k$, donc il existe un sommet u dans un point d'attache P_t , $t \neq j$ tel que dans H ,

le sommet u est adjacent à un sommet $z \in N_{B_i}(P_t)$ et aux sommets x et y . Puisque $|Q_1 \setminus P_t| > k$, il existe un sommet $v \in Q_1 \setminus P_t$ tel qu'il n'existe aucune paire de F contenant à la fois v et un sommet de $\{x, y, z, u\}$. L'ensemble $\{x, y, u, v, z\}$ induit un dard si x et y ne sont pas adjacents à z , une gemme si l'un d'eux est adjacent à z , ou si les deux sont adjacents à z , $\{x, y, z, v\}$ induit un C_4 dans H , ce qui est une contradiction. Il en résulte que N doit être une clique.

Nous montrons maintenant que N est une clique critique, une clique maximale ou intersecte exactement une clique maximale de H' . Supposons que N ne soit ni une clique critique ni une clique maximale de H' et intersecte au moins deux cliques maximales de H' . Supposons d'abord que N n'est pas un module, donc il existe $u, v \in N$ et $x \in V(H') \setminus N$ adjacent à seulement u ou v . Puisque N n'est pas une clique maximale de H' , il existe $y \in V(H') \setminus N$ adjacent à u et v . Avec un sommet $w \in \bigcup_{i=1}^l P_i$ non affecté par F , les sommets $\{u, v, w, x, y\}$ induisent dans H une gemme si x et y sont adjacents et un dard sinon, ce qui implique une contradiction. Si N est un module et n'est pas une clique critique, alors N n'est pas un module clique maximal, donc il existe $u \in N$, un sommet v dans le même module que u dans H' et un sommet $w \in \bigcup_{i=1}^l P_i$ non affecté par F tel que $vw \notin E(H)$. Puisque N intersecte deux cliques maximales de H' , il existe deux sommets non adjacents $x, y \in V(H')$ qui sont adjacents à la fois à u et v . Les sommets $\{u, v, w, x, y\}$ induisent un dard dans H puisque w n'est adjacent qu'à u . Nous concluons que si $r = 1$, alors N est une clique critique, une clique maximale ou est incluse dans exactement une clique maximale de H' .

Cas (ii): $r > 1$ et $(\sum_{i=1}^l |P_i|) - \max_{1 \leq j \leq r} \{|Q_j|\} > k$. Nous montrons que N est une clique dans H . Supposons à nouveau, par contradiction, que ce ne soit pas le cas, alors il existe deux sommets $x, y \in N$ tels que $xy \notin E(H)$. Pour toute paire de sommets $u_i \in Q_i, u_j \in Q_j, i \neq j$, l'ensemble $\{x, u_i, y, u_j\}$ induit un C_4 dans H si ni u_j ni u_i ne sont affectés par F . Cela implique que les sommets de Q_i ou Q_j doivent être affectés par F et donc qu'au plus une clique Q_i n'est pas affecté par F . Puisque $(\sum_{i=1}^l |P_i|) - \max_{1 \leq j \leq r} \{|Q_j|\} > k$, nous avons $|F| > k$, ce qui est une contradiction. Par conséquent, N doit être une clique dans H .

Nous montrons ensuite que N est une clique critique de H . Supposons que N n'est pas un module de H , alors il existe $x, y \in N$ et $z \in V(H) \setminus N$ adjacent à un seul des sommets x et y dans H , supposons que ce soit x . Pour toute paire de sommets $u_i \in Q_i, u_j \in Q_j, i \neq j$, si ni u_i ni u_j ne sont affectés par F , l'ensemble $\{x, y, u_i, u_j, z\}$ induit un dard si u_i et u_j ne sont pas adjacents à z , ou une gemme si l'un d'entre eux est adjacent à z . Cela implique qu'au plus un Q_i n'est pas affecté par F et donc comme précédemment nous avons $|F| > k$, une contradiction. Par conséquent, N doit être un module dans H . Si N n'est pas une clique critique de H , alors N est strictement contenu dans une clique critique N' . Un sommet $x \in N' \setminus N$ doit avoir le même voisinage fermé que les sommets de N dans H . Si x n'est pas contenu dans P_i ou $N_{B_i}(P_i)$ pour un $1 \leq i \leq l$, alors F doit affecter tous les sommets de $P_1 \cup \dots \cup P_l$, ce qui implique que $|F| > k$, une contradiction. Si x est contenu dans P_i ou $N_{B_i}(P_i)$ pour un $1 \leq i \leq l$, alors F doit affecter tous les sommets contenus dans tous les Q_j tels que $P_i \not\subseteq Q_j$. Puisque $(\sum_{i=1}^l |P_i|) - \max_{1 \leq j \leq r} \{|Q_j|\} > k$, cela implique $|F| > k$, une contradiction. Nous concluons que N est une clique critique dans H .

Il reste à montrer que N est une clique critique ou intersecte exactement une clique maximale dans $H' = H \setminus \bigcup_{i=1}^l V(B_i)$. Puisque N est un module dans H' , si N n'est pas une clique critique de H' , il existe $u \in N$, un sommet v dans la même clique critique que u dans H' et un sommet $w \in \bigcup_{i=1}^l P_i$ non affecté par F tel que $\{v, w\} \notin E(H)$. Si N intersecte au moins deux cliques maximales de H' , il existe deux sommets non adjacents $x, y \in V(H')$ qui sont adjacents à la fois à u et v . Les sommets $\{u, v, w, x, y\}$ induisent un dard dans H puisque w n'est adjacent qu'à u . Nous concluons que N est une clique critique ou intersecte exactement une clique maximale de H' . \square

Nous pouvons déduire du [lemme 5.17](#) la règle qui suit, qui permet de borner le nombre de 1-SC-branches ayant le même voisinage.

Règle 5.8. Soit B_1, \dots, B_l des 1-SC-branches disjointes de G avec des points d'attache P_1, \dots, P_l qui partagent le même voisinage N dans $G \setminus \bigcup_{i=1}^l V(B_i)$ et forment une union disjointe de cliques Q_1, \dots, Q_r dans $G[P_1 \cup \dots \cup P_l]$. Si $\sum_{i=1}^l |P_i| > 2k + 1$ alors :

- Si $r = 1$, retirer les sommets $\bigcup_{i=1}^l V(B_i)$, ajouter deux cliques adjacentes K et K' de taille $k + 1$ avec le voisinage N et un sommet u_K adjacent aux sommets de K .

- Si $r > 1$ et $(\sum_{i=1}^l |P_i|) - \max_{1 \leq j \leq r} \{|Q_j|\} > k$, retirer les sommets $\bigcup_{i=1}^l V(B_i)$, ajouter deux cliques non adjacentes de taille $k + 1$ avec le voisinage N .

Lemme 5.18. *La règle 5.8 est sûre.*

Preuve. Soit G' le graphe obtenu après l'application de la règle 5.8 sur les 1-SC-branches B_1, \dots, B_l . Soit S les sommets ajoutés par la règle 5.8. Remarquons que le sous-graphe $G'_S = G'[S]$ est strictement cordal et correspond à deux 1-SC-branches de G' qui satisfont les conditions du lemme 5.17. Dans un premier temps, supposons qu'il existe une k -édition F de G , soit $H = G \triangle F$ et $H_a = H \setminus \bigcup_{i=1}^l V(B_i)$. Par le lemme 5.17, N est une clique critique, une clique maximale (sauf si $r > 1$) ou intersecte exactement une clique maximale de H_a . Par conséquent, par le lemme 5.3 et l'observation 5.1 le graphe $H' = (H_a, N) \otimes (G'_S, K \cup K')$ est strictement cordal. L'édition F' telle que $H' = G' \triangle F'$ est de taille au plus $|F| \leq k$, donc F' est une k -édition de G' .

Inversement, supposons qu'il existe une k -édition F' de G' , soit $H' = G' \triangle F'$ et $H'_a = H' \setminus S$. Remarquons que le sous-graphe $G_b = G[\bigcup_{i=1}^l V(B_i)]$ est strictement cordal. Par le lemme 5.17, N est une clique critique, une clique maximale (sauf si $r > 1$) ou intersecte exactement une clique maximale de H'_a . Soit Q_1, \dots, Q_r les cliques de G_b adjacentes à N , en utilisant des arguments similaires à ceux du lemme 5.13, il est facile de voir que chaque Q_i est une clique critique, une clique maximale ou intersecte exactement une clique maximale de G_b . Par conséquent, par le lemme 5.3 et l'observation 5.1 le graphe $H = (H'_a, N) \otimes (G_b, Q_1 \cup \dots \cup Q_r)$ est strictement cordal. L'édition F telle que $H = G \triangle F$ est de taille au plus $|F'| \leq k$, donc F est une k -édition de G . \square

5.4.3 Réduction des 2-SC-branches

Soit B une 2-SC-branche d'un graphe G réduit par la règle 5.7, avec les points d'attache P_1 et P_2 . Nous disons que B est *propre* si B^R est connexe et définissons la *longueur* d'une 2-SC-branche propre comme la longueur du plus court chemin entre les deux points d'attache dans $\mathcal{C}(B)$. Une coupe minimum de B est un ensemble d'arêtes $M \subseteq E(B)$ de taille minimum tel que P_1 et P_2 ne sont pas dans la même composante connexe de $B - M$. Nous notons par $\text{mc}(B)$ la taille d'une coupe minimum de B . Nous pouvons observer qu'une coupe minimum de B contient les arêtes entre une paire de cliques critiques consécutives d'un plus court chemin entre P_1 et P_2 et les arêtes entre l'une de ces cliques critiques et les sommets qu'elles ont dans leur voisinage commun. Nous commençons par l'observation suivante qui se révélera utile pour réduire les 2-SC-branches.

Observation 5.2. Soit F une édition optimale de G et $F_1 \subseteq F$. Si F_2 est une édition optimale du graphe $G \triangle F_1$, alors $F_1 \cup F_2$ est une édition optimale de G .

Nous montrons à présent que si une 2-SC-branche est suffisamment longue, alors une k -édition peut éventuellement contenir une coupe minimum et n'affecte autrement que les sommets des points d'attache et leurs voisinages. En particulier une 2-SC-branche B de longueur au moins $k + 3$ permet de faire apparaître un cycle de longueur au moins $k + 3$ si les deux points d'attache de B sont dans la même composante connexe de $G \setminus V(B)$. Une édition de taille k ne peut pas compléter un tel cycle pour qu'il soit cordal, et donc une édition devra soit déconnecter B soit déconnecter les points d'attache à l'extérieur de B .

Lemme 5.19. Soit $(G = (V, E), k)$ une instance positive de STRICTLY CHORDAL EDITION et B une 2-SC-branche propre de G de longueur au moins $k + 3$ avec les points d'attache P_1, P_2 . Alors il existe une k -édition optimale F de G telle que :

- si P_1 et P_2 sont dans la même composante connexe de $(G \triangle F) \setminus V(B^R)$, alors F contient une coupe de B , qui est $(P_1 \times N_B(P_1)), (P_2 \times N_B(P_2))$ ou une coupe minimum de B
- les autres sommets de B affectés par F sont inclus dans $P_1 \cup N_B(P_1) \cup P_2 \cup N_B(P_2)$,
- dans $G \triangle F$ les sommets de $N_B(P_1)$ (resp. $N_B(P_2)$) sont tous adjacents aux mêmes sommets de $V(G) \setminus V(B^R)$.

Preuve. Soit F une k -édition optimale de G et $H = G\Delta F$. Soit C_1 (resp. C_2) la clique critique de H qui contient P_1 (resp. P_2). Définissons $H' = H \setminus V(B^R)$, $C'_1 = C_1 \setminus (V(B^R))$ et $C'_2 = C_2 \setminus (V(B^R))$.

Nous considérons d'abord le cas où P_1 et P_2 sont dans des composantes connexes différentes de H' . Soit H'_1 et H'_2 les deux composantes connexes de H' contenant respectivement P_1 et P_2 , et H'_3 le reste de H' . Pour $i \in \{1, 2\}$, puisque B^R est connexe, le [lemme 5.13](#) implique que $N_B(P_i)$ est une clique critique, une clique maximale ou intersecte exactement une clique maximale de B^R . Par le [lemme 5.14](#), C'_i est une clique critique ou intersecte exactement une clique maximale de la composante connexe H'_i de H' qui contient C'_i . Par le [lemme 5.3](#), le graphe H^* correspondant à l'union disjointe de

$$((B^R, N_B(P_1)) \otimes (H_1, C'_1), N_B(P_2)) \otimes (H_2, C'_2)$$

et H_3 est strictement cordal (illustré dans la [figure 5.8](#)) et l'édition F^* telle que $H^* = G\Delta F^*$ vérifie les propriétés souhaitées. Puisque $|F^*| \leq |F|$ par construction, nous concluons que F^* est une k -édition optimale.

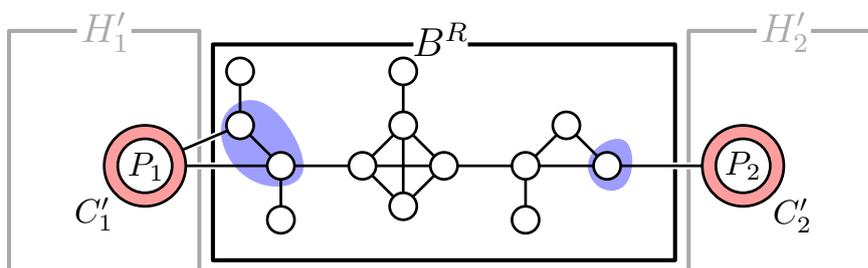


Figure 5.8: Illustration du cas où $B\Delta F$ est connexe dans le [lemme 5.19](#).

Supposons maintenant que P_1 et P_2 sont dans la même composante connexe de H' et définissons $\pi_{H'}$ un plus court chemin entre $c_1 \in P_1$ et $c_2 \in P_2$. Notons que ce chemin peut être de longueur 1 si c_1 et c_2 sont adjacents dans H' . Nous considérons d'abord le cas où il existe un chemin $\pi_B = \{c_1 = u_1, u_2, \dots, u_r = c_2\}$ (pas nécessairement induit) de c_1 à c_2 dans B qui existe encore dans $B\Delta F$. Observons que ce chemin π_B est de longueur au moins $k+3$ puisque B est de longueur au moins $k+3$. Il n'y a pas d'arête entre les deux chemins dans G , donc $H[V(\pi_B) \cup V(\pi_{H'})]$ admet au moins un cycle induit de longueur au moins égale à 4, ce qui est une contradiction. Nous en concluons que si P_1 et P_2 sont dans la même composante connexe de H' , alors F contient une coupe de B . Sinon, il y aurait un chemin π_B de c_1 à c_2 dans B qui existe encore dans $B\Delta F$.

Par conséquent, nous pouvons supposer que F contient une coupe de B . Premièrement, si $F_1 = (P_1 \times N_B(P_1)) \subseteq F$, considérons le graphe $G_1 = G\Delta F_1$. Observons que $B_1 = B \setminus P_1$ est une 1-SC-branche de G_1 avec le point d'attache P_2 . Par le [lemme 5.15](#), il existe une k -édition optimale F_2 de G_1 où les sommets de B_1 affectés par F_2 sont dans $P_2 \cup N_{B_1}(P_2)$. Par l'[observation 5.2](#), $F_1 \cup F_2$ est une k -édition optimale et elle vérifie les propriétés souhaitées. Les mêmes arguments peuvent être utilisés si $(P_2 \times N_B(P_2)) \subseteq F$.

Maintenant, si F ne contient ni $(P_1 \times N_B(P_1))$ ni $(P_2 \times N_B(P_2))$, définissons $F_1 = F \cap [V(B^R)]^2$. Il existe deux composantes connexes dans $B\Delta F_1$, définissons B_1 celle qui contient P_1 et B_2 celle qui contient P_2 . Ces composantes connexes sont des 1-SC-branches de $G_1 = G\Delta F_1$ avec les points d'attache P_1 et P_2 . Comme précédemment, le [lemme 5.15](#) s'applique aux 1-SC-branches B_1 et B_2 , il existe donc une édition optimale F_2 de G_1 où les seuls sommets de B_1 et B_2 affectés sont inclus dans $P_1 \cup N_B(P_1)$ et $P_2 \cup N_B(P_2)$. Si F_1 est une coupe minimum de B , l'édition $F_1 \cup F_2$ est une édition k -optimale de G qui vérifie les propriétés souhaitées, sinon, en remplaçant F_1 par une coupe minimum de B , nous obtenons une édition plus petite qui vérifie les propriétés. \square

Nous pouvons à présent définir la dernière règle de notre noyau. Pour une 2-SC-branche propre B de G de longueur au moins $k+3$, elle remplace B^R par un chemin de $k+2$ cliques ayant une coupe minimum de taille $\min\{\text{mc}(B), k+1\}$.

Règle 5.9. Soit B une 2-SC-branche propre de G de longueur au moins $k+3$ avec les points d'attache P_1, P_2 . Retirer les sommets de $V(B^R)$, ajouter le chemin de $k+2$ cliques suivant :

$$K_{\min\{|N_B(P_1)|, k+1\}} - K_{k+1} - K_1 - K_{\text{mc}(B)} - K_{k+1}^1 - \dots - K_{k+1}^{k-3} - K_{\min\{|N_B(P_2)|, k+1\}}$$

où K_n est une clique de taille n et $K_{\min\{|N_B(P_1)|, k+1\}}$ (resp. $K_{\min\{|N_B(P_2)|, k+1\}}$) est adjacente à P_1 (resp. P_2).

Lemme 5.20. *La règle 5.9 est sûre.*

Preuve. Soit G' le graphe obtenu après l'application de la règle 5.9 sur la 2-SC-branche propre B avec les points d'attache P_1, P_2 . Soit S les sommets ajoutés par la règle 5.9. Observons que $B' = G'[S \cup P_1 \cup P_2]$ est une 2-SC-branche propre de G' de longueur $k+3$ et $\text{mc}(B') = \min\{k+1, \text{mc}(B)\}$.

Supposons qu'il existe une k -édition F de G et soit $H = G \triangle F$. Nous en choisissons une qui satisfait le lemme 5.19. Observons que si $\text{mc}(B) \geq k+1$, alors F n'affecte pas les sommets de $V(B)$ autres que $P_1 \cup N_B(P_1) \cup P_2 \cup N_B(P_2)$ et si $|N_B(P_i)| > k$, $i \in \{1, 2\}$, alors F n'affecte pas les sommets de $N_B(P_i)$. Soit C_i la clique critique de H qui contient P_i et $C'_i = C_i \setminus B^R$. Par le lemme 5.14, C'_i est une clique critique ou intersecte exactement une clique maximale de $H_a = H \setminus V(B^R)$. De plus, observons que $N_{B'}(P_i)$ est une clique critique de B'^R .

Nous construisons maintenant une k -édition F' de G' . Nous considérons d'abord les trois cas où F contient une coupe de B . Supposons sans perte de généralité que $(P_1 \times N_B(P_1)) \subseteq F$ et $(P_2 \times N_B(P_2)) \cap F = \emptyset$. Observons que $(C'_1 \times N_{B'}(P_1)) \subseteq F$ puisque C_1 est une clique critique de H . Par le lemme 5.3 le graphe $H' = ((H_a, C'_1) \otimes (B'^R, N_{B'}(P_1)))$ est strictement cordal. Si $(P_1 \times N_B(P_1)) \subseteq F$ et $(P_2 \times N_B(P_2)) \subseteq F$ alors H' le graphe correspondant à l'union disjointe de H_a et B'^R est strictement cordal. Si F contient une coupe minimum de B , considérons B'_i , $i \in \{1, 2\}$ la composante connexe de $B' \triangle F'_1$ contenant P_i , où F'_1 est une coupe minimum de B' . Par le lemme 5.3 le graphe $H' = ((H_a, C'_1) \otimes (B'_1, N_{B'}(P_1)), C'_2) \otimes (B'_2, N_{B'}(P_2))$ est strictement cordal.

Supposons maintenant que F ne contient pas de coupe de B . La preuve du lemme 5.19 montre que dans ce cas, P_1 et P_2 ne sont pas dans la même composante connexe de H_a . Soit H_1 et H_2 ces composantes connexes et H_3 le reste de H_a . Par le lemme 5.3, le graphe H' correspondant à l'union disjointe de $((B'^R, N_{B'}(P_1)) \otimes (H_1, C'_1), N_{B'}(P_2)) \otimes (H_2, C'_2)$ et H_3 est strictement cordal.

Pour chacun des cas considérés, définissons F' l'édition telle que $H' = G' \triangle F'$, par construction $|F'| \leq |F|$, par conséquent, F' est une k -édition de G' .

Des arguments symétriques permettent de montrer que toute k -édition de G' permet de construire une k -édition de G . \square

5.4.4 Borner la taille d'une instance réduite

Nous montrons dans cette section que le problème STRICTLY CHORDAL EDITION admet un noyau avec $O(k^4)$ sommets. Nous commençons par montrer que les règles portant sur les SC-branches peuvent être appliquées en temps polynomial.

Lemme 5.21. *Les règles 5.7 à 5.9 peuvent être appliquées exhaustivement en temps polynomial.*

Preuve. Commençons par rappeler que le graphe des cliques critiques peut être calculé en temps linéaire, en particulier les cliques critiques sont des modules forts qui peuvent être énumérés en temps linéaire [124]. De plus les graphes blocs peuvent être reconnus en temps linéaire [123]. Nous montrons que nous pouvons énumérer toutes les 1-SC-branches et 2-SC-branches en temps polynomial. Puisqu'un point d'attache est par définition une clique critique de G , nous pouvons détecter les 1-SC-branches en supprimant une clique critique de $\mathcal{C}(G)$ et en recherchant parmi les composantes connexes restantes celles qui induisent un graphe bloc connexe avec P (dans $\mathcal{C}(G)$). En considérant un ensemble maximal de telles composantes connexes plus la clique critique P , nous obtenons une 1-SC-branche B . Nous procédons de la même manière pour détecter les 2-SC-branches propres en supprimant une paire de cliques critiques non adjacentes P_1, P_2 de $\mathcal{C}(G)$, et en recherchant parmi les composantes connexes restantes celles qui induisent un graphe bloc connexe (dans $\mathcal{C}(G)$) avec P_1 et P_2 . De telles composantes connexes avec P_1 et P_2 donnent une 2-SC-branche B propre. Puisqu'il y a $O(|V(G)|)$ cliques critiques, cette énumération peut être faite en temps polynomial. \square

Nous montrons maintenant que toute instance positive de STRICTLY CHORDAL EDITION réduite par les règles 5.5 à 5.9 contient $O(k^4)$ sommets.

Théorème 5.4. *STRICTLY CHORDAL EDITION* admet un noyau avec $O(k^4)$ sommets.

Preuve. Soit $(G = (V, E), k)$ une instance positive de STRICTLY CHORDAL EDITION réduite par les règles 5.5 à 5.9. Soit F une k -édition de G et $H = G \Delta F$. Nous supposons que G est connexe, les arguments qui suivent peuvent être facilement adaptés si ce n'est pas le cas en sommant sur toutes les composantes connexes de G . Le graphe H est strictement cordal, donc le graphe $\mathcal{C}(H)$ est un graphe bloc. Nous montrons d'abord que $\mathcal{C}(H)$ contient $O(k^3)$ cliques critiques. Nous disons qu'une clique critique de $\mathcal{C}(H)$ est affectée si elle contient un sommet affecté par F . Soit A l'ensemble des cliques critiques affectées de $\mathcal{C}(H)$. Puisque $|F| \leq k$, nous avons $|A| \leq 2k$. Soit T le sous-graphe induit connexe minimal de H qui couvre toutes les cliques critiques de A , et A' l'ensemble des sommets de degré au moins égal à 3 dans T (illustré dans la figure 5.9). Rappelons qu'un tel sous-graphe est unique d'après le lemme 5.4.

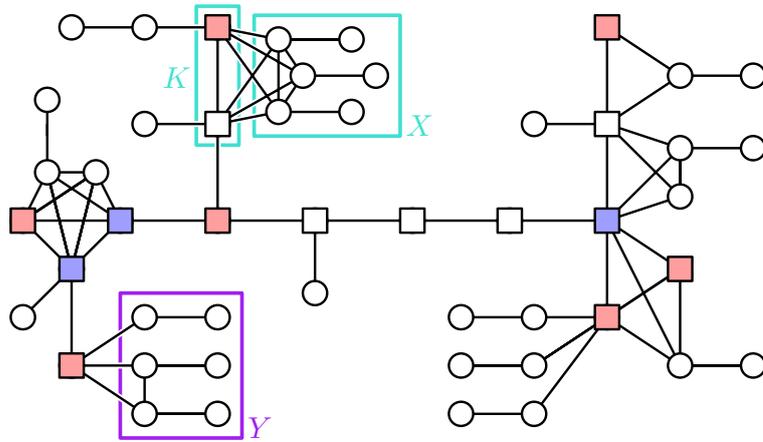


Figure 5.9: Illustration du graphe des cliques critiques d'une instance réduite dans la preuve du théorème 5.4. Les noeuds carrés correspondent aux cliques critiques de T , ceux en rouge sont les cliques critiques de A , ceux en bleu sont les cliques critiques de $A' \setminus A$.

Nous montrons d'abord que $|V(T)|$ contient $O(k^2)$ cliques critiques. Par le lemme 5.4, $|A' \setminus A| \leq 3 \cdot |A| \leq 6k$. Les composantes connexes du graphe $T \setminus (A \cup A')$ sont des chemins puisque chaque sommet est de degré au plus 2 et par le lemme 5.4 il y a au plus $4k$ tels chemins. Soit R l'un de ces chemins, il est composé de cliques critiques non affectées, il existe donc B une 2-SC-branche propre de G qui contient R . Les extrémités de R sont les points d'attache de B . Les 2-SC-branches propres ont été réduites par la règle 5.9. Par conséquent R est de longueur au plus $k + 3$ et donc $|V(T)| \leq 2k + 6k + (k + 3) \cdot 4k = O(k^2)$.

Montrons maintenant que $\mathcal{C}(H) \setminus V(T)$ contient $O(k^3)$ cliques critiques. Observons d'abord que chaque composante connexe de $\mathcal{C}(H) \setminus V(T)$ est adjacente à des sommets de T puisque le graphe est réduit par la règle 5.5. Puisque $\mathcal{C}(H)$ est un graphe bloc, les composantes connexes de $\mathcal{C}(H) \setminus V(T)$ sont adjacentes soit à une clique critique de T soit à exactement une clique maximale de T (sinon, il y aurait un diamant dans $\mathcal{C}(H)$).

Nous affirmons qu'il y a $O(k^3)$ cliques critiques dans les composantes connexes de $\mathcal{C}(H) \setminus V(T)$ adjacentes à des cliques maximales de T . Puisque T est un graphe bloc et que $|V(T)| = O(k^2)$, il y a $O(k^2)$ cliques maximales dans T . De plus, il n'y a qu'une seule composante connexe de $\mathcal{C}(H) \setminus V(T)$ adjacente à chaque clique maximale, sinon il y aurait un diamant dans $\mathcal{C}(H)$. Prenons K , une clique maximale de T , et soit X la composante connexe de $\mathcal{C}(H) \setminus V(T)$ adjacente à K dans $\mathcal{C}(H)$ (illustrée dans la figure 5.9). Observons que X doit être une union de 1-SC-branches et que leurs points d'attache forment une clique. Par la règle 5.8, il y a au plus $2k + 1$ 1-SC-branches dans X et chacune a été réduite par la règle 5.7, donc X contient au plus $4k + 2$ cliques critiques. Nous en concluons qu'il y a $|V(T)| \cdot O(k) = O(k^3)$ cliques critiques dans les composantes connexes de $\mathcal{C}(H) \setminus V(T)$ adjacentes à des cliques maximales de T .

Enfin, nous affirmons qu'il y a $O(k^2)$ cliques critiques dans les composantes connexes de $\mathcal{C}(H) \setminus V(T)$ adjacentes aux cliques critiques de T . Prenons d'abord une clique critique de $T \setminus A$ et ses composantes connexes adjacentes de $\mathcal{C}(H) \setminus V(T)$. Observons qu'elles forment une 1-SC-branche, qui est réduite par la [règle 5.7](#), donc la composante connexe adjacente consiste en une unique clique critique. Rappelons que $|T \setminus A| = O(k^2)$, il existe donc $O(k^2)$ telles 1-SC-branches.

Prenons ensuite une clique critique a de A , et soit C_1, \dots, C_r les composantes connexes de $\mathcal{C}(H) \setminus V(T)$ adjacentes à a . Nous notons Y l'union de ces composantes connexes (illustrée dans la [figure 5.9](#)). Observons que chaque C_i doit être une union de 1-SC-branches dont les points d'attache forment une clique Q_i . Soit B_1, \dots, B_l les 1-SC-branches de Y avec les points d'attache P_1, \dots, P_l . Par la [règle 5.8](#), il y a au plus $2k + 1$ 1-SC-branches par composante connexe et si $\sum_{i=1}^l |P_i| > 2k + 1$, alors $(\sum_{i=1}^l |P_i|) - \max_{1 \leq j \leq r} \{|Q_j|\} \leq k$, impliquant qu'au plus $3k + 1$ 1-SC-branches sont contenues dans Y . Chacune de ces 1-SC-branches est réduite par la [règle 5.7](#), par conséquent Y contient au plus $6k + 2$ cliques critiques au total. Puisque $|A| = O(k)$, il y a $O(k)$ tels ensembles de 1-SC-branches dans $\mathcal{C}(H) \setminus V(T)$. Au total, $\mathcal{C}(H) \setminus V(T)$ contient $O(k^2)$ cliques critiques adjacentes à des sommets de T .

Nous concluons donc que $|V(\mathcal{C}(H))| = O(k^3)$. Puisque toutes les cliques critiques non affectées de H contiennent au plus $k + 1$ sommets par la [règle 5.6](#) et par le [lemme 3.4](#) celles de A qui sont affectées contiennent en tout $O(k^2)$ sommets, nous pouvons conclure que $|V(G)| = O(k^4)$. Pour finir, une instance peut être réduite exhaustivement en temps polynomial d'après les [lemmes 3.5](#) et [5.21](#). \square

5.4.5 Noyaux pour les versions complétion et délétion

Dans cette section, nous présentons des noyaux pour STRICTLY CHORDAL COMPLETION et STRICTLY CHORDAL DELETION. Les [règles 5.5](#) et [5.6](#) sont immédiatement sûres pour ces variantes du problème. Nous pouvons également observer que les [règles 5.7](#) à [5.8](#) sont sûres pour les variantes complétion et délétion puisqu'elles sont un cas particulier de l'édition. Cependant, nous pouvons utiliser des règles plus fortes pour traiter les 2-SC-branches propres et borner leur longueur à un nombre constant de cliques critiques. Cela est dû au fait que nous n'autorisons que l'ajout ou la suppression d'arêtes.

Dans le cas de STRICTLY CHORDAL COMPLETION, nous pouvons observer que pour un graphe G , il suffit de considérer les 2-SC-branches B propres de longueur au moins 3 qui ont des points d'attache qui appartiennent à différentes composantes connexes de $G \setminus V(B^R)$ puisqu'une SC-branche reste connexe dans n'importe quelle complétion de G . Le lemme qui suit résulte de cette observation et sa preuve est contenue dans la preuve du [lemme 5.19](#).

Lemme 5.22. *Soit $G = (V, E)$ un graphe et B une 2-SC-branche propre de G de longueur au moins 3 avec des points d'attache P_1, P_2 qui sont dans des composantes connexes différentes de $G \setminus V(B^R)$. Alors il existe une complétion optimale F de G telle que :*

- les sommets de B affectés par F sont inclus dans $P_1 \cup N_B(P_1) \cup P_2 \cup N_B(P_2)$,
- dans $G + F$ les sommets de $N_B(P_1)$ (resp. $N_B(P_2)$) sont tous adjacents aux mêmes sommets de $V(G) \setminus V(B^R)$.

À partir de ce lemme, nous pouvons définir une règle plus forte pour la variante complétion et dont la sûreté est immédiate par le [lemme 5.20](#).

Règle 5.10. Soit $(G = (V, E), k)$ une instance de STRICTLY CHORDAL COMPLETION et B une 2-SC-branche propre de G de longueur au moins 3 avec des points d'attache P_1, P_2 qui sont dans des composantes connexes différentes de $G \setminus V(B^R)$. Retirer les sommets de $V(B^R)$ et ajouter deux nouvelles cliques K_1 et K_2 de taille respectivement $\min\{|N_B(P_1)|, k + 1\}$ et $\min\{|N_B(P_2)|, k + 1\}$ avec les arêtes $(P_1 \times K_1), (K_1 \times K_2), (K_2 \times P_2)$.

Dans le cas de STRICTLY CHORDAL DELETION, nous pouvons observer que si les deux points d'attache de B , une 2-SC-branche propre d'une longueur d'au moins 3, sont dans la même composante connexe de $G \setminus V(B^R)$, alors soit B est déconnectée, soit les points d'attache ne sont pas

dans la même composante connexe de $(G - F) \setminus V(B^R)$ pour n'importe quelle suppression F de G . En effet, si ce n'était pas le cas, nous pourrions trouver un cycle induit de longueur au moins égale à 4 puisque les ajouts d'arêtes ne sont pas autorisés. Le lemme suivant résulte de cette observation et sa preuve est contenue dans la preuve du [lemme 5.19](#).

Lemme 5.23. *Soit $G = (V, E)$ un graphe et B une 2-SC-branche propre de G de longueur au moins 3 avec les points d'attache P_1, P_2 . Alors il existe une déletion optimale F de G telle que :*

- si P_1 et P_2 sont dans la même composante connexe de $(G - F) \setminus V(B^R)$, alors F contient une coupe de B , qui est $(P_1 \times N_B(P_1))$ ou $(P_2 \times N_B(P_2))$ ou une coupe minimum de B ,
- les autres sommets de B affectés par F sont inclus dans $P_1 \cup N_B(P_1) \cup P_2 \cup N_B(P_2)$,
- dans $G \triangle F$ les sommets de $N_B(P_1)$ (resp. $N_B(P_2)$) sont tous adjacents aux mêmes sommets de $V(G) \setminus V(B^R)$.

Comme précédemment, nous pouvons définir une règle plus forte à partir de ce lemme dont la sûreté découle du [lemme 5.20](#).

Règle 5.11. Soit $(G = (V, E), k)$ une instance de STRICTLY CHORDAL DELETION et soit B une 2-SC-branche propre de G de longueur au moins 3 avec les points d'attache P_1, P_2 . Retirer les sommets de $V(B^R)$, ajouter le chemin de 5 cliques suivant :

$$K_{\min\{|N_B(P_1)|, k+1\}} - K_{k+1} - K_1 - K_{\text{mc}(B)} - K_{\min\{|N_B(P_2)|, k+1\}}$$

où K_n est une clique de taille n et $K_{\min\{|N_B(P_1)|, k+1\}}$ (resp. $K_{\min\{|N_B(P_2)|, k+1\}}$) est adjacente à P_1 (resp. P_2).

Dans la preuve du [théorème 5.4](#), le nombre de cliques critiques contenues dans $\mathcal{C}(H)$ dépend de la taille de T . En particulier nous montrons que $|\mathcal{C}(H) \setminus V(T)| = |V(T)| \cdot O(k)$. La [règle 5.9](#) borne linéairement en k la longueur des 2-SC-branches, c'est pour cela que dans le cas édition, $|V(T)| = O(k^2)$. Cependant, avec les [règles 5.10](#) et [5.11](#), les 2-SC-branches ont une taille constante, ce qui implique que dans le cas de la complétion et de la déletion, T contient $O(k)$ cliques critiques. Nous en déduisons que dans le cas de la complétion et la déletion, un graphe réduit contient $O(k^2)$ cliques critiques et a donc $O(k^3)$ sommets.

Théorème 5.5. *STRICTLY CHORDAL COMPLETION et STRICTLY CHORDAL DELETION admettent un noyau avec $O(k^3)$ sommets.*

Partie I : conclusion et perspectives

Nous avons présenté dans ce chapitre des noyaux polynomiaux pour divers problèmes de modification d'arêtes :

- TRIVIALY PERFECT EDITION ainsi que ses versions complétion et délétion admettent un noyau qui contient $O(k^2)$ sommets,
- THRESHOLD EDITION ainsi que ses versions complétion et délétion admettent un noyau qui contient $O(k^2)$ sommets,
- BLOCK GRAPH EDITION et BLOCK GRAPH DELETION admettent un noyau qui contient $O(k^2)$ sommets,
- STRICTLY CHORDAL EDITION admet un noyau qui contient $O(k^4)$ sommets,
- STRICTLY CHORDAL COMPLETION et STRICTLY CHORDAL DELETION admettent un noyau qui contient $O(k^3)$ sommets.

Tous ces noyaux ont été élaborés avec l'approche que nous appelons « décomposition et sommets affectés ». Cette approche s'avère être idéale pour la mise au point de noyaux pour des classes de graphes qui ont une structure arborescente, ce qui est notamment le cas pour des sous-classes des graphes cordaux. Rappelons, comme cela a été vu dans la [section 3.3](#), que cette approche est utilisée pour d'autres classes de graphes telles que les graphes 3-leaf power, les graphes ptolémaïques, les graphes d'intervalles propres ou les cographes. Cependant, comme nous avons pu le voir dans les [chapitres 4 et 5](#), bien que les idées employées soient similaires, il est nécessaire de les adapter à la structure spécifique des classes considérées.

L'exemple où les noyaux sont les plus proches est dans le [chapitre 4](#), pour les graphes trivialement parfaits et les graphes threshold. En particulier, nous y avons défini la structure de peigne, qui est presque identique dans les deux classes, en effet, « module trivialement parfait » est remplacé par « indépendant » pour les graphes threshold. Les règles portant sur cette structure de peigne ont des énoncés identiques. De plus leur sûreté peut être prouvée de la même manière en choisissant les définitions et lemmes structurels adaptés à la classe considérée. La preuve bornant la taille d'une instance réduite de TRIVIALY PERFECT EDITION peut être également adaptée de manière immédiate pour le problème THRESHOLD EDITION, bien qu'une analyse plus fine permette d'avoir une borne avec un facteur constant plus petit.

Dans le [chapitre 5](#), pour les graphes blocs et les graphes strictement cordaux nous avons utilisé la notion de 1-branche et 2-branche. Ces structures correspondent à des sous-graphes appartenant à la classe considérée, connectés au reste du graphe en un ou deux « points » respectivement. Notons que cette notion avait été introduite par Bessy *et al.* [13] dans le noyau pour le problème 3-LEAF POWER EDITION. Bien que les règles portant sur ces structures utilisent les mêmes idées, la preuve de leur sûreté nécessite l'utilisation de la structure propre à chacune des classes.

Perspectives Une des motivations à l'étude de l'existence de noyaux polynomiaux pour des problèmes de modification d'arêtes vers des sous-classes de graphes cordaux était la conjecture de Bessy et Perez [14].

Conjecture. Soit \mathcal{G} une classe de graphes caractérisée par une famille d'obstructions contenant les trous plus un ensemble fini de graphes. Alors le problème \mathcal{G} -COMPLETION admet un noyau polynomial.

Montrer cette conjecture apparaît être un défi de taille. Cela supposerait d'élaborer un noyau polynomial générique, un « méta-noyau », qui pourrait s'adapter à toute cette famille de problèmes. Pour ce faire, il serait intéressant d'avoir une décomposition des graphes appartenant à notre famille de classes. Il est également possible de considérer une famille de classe de graphes plus petits en ajoutant des restrictions sur l'ensemble d'obstructions. Notons qu'il existe des « méta-noyaux », c'est à dire un noyau générique pour des problèmes exprimables en logique MSO_2 , avec certaines contraintes de couverture, restreints aux graphes planaires ou de genre borné [18]. Cependant, les problèmes de modifications d'arêtes que nous considérons ne sont pas couverts par de tels résultats.

Inversement, il serait intéressant de chercher à infirmer cette conjecture en montrant l'inexistence d'un noyau polynomial pour un problème \mathcal{G} -COMPLETION où la classe graphes \mathcal{G} est caractérisée par une famille d'obstructions contenant les trous plus un ensemble fini de graphes. Rappelons que de tels résultats existent pour des problèmes de modification d'arêtes, mais principalement vers des classes de graphes caractérisées par un seul sous-graphe interdit [29, 75, 93, 110].

Pour de nombreux autres problèmes de modifications d'arêtes vers des sous-classes des graphes cordaux, l'existence d'un noyau polynomial reste une question ouverte (cf. table 3.1). Remarquons que c'est particulièrement le cas pour l'édition et la déletion, cela est en partie dû au fait que l'ensemble des trous est une famille infinie d'obstruction dans le cas de l'édition et la déletion, mais se comporte comme une famille finie dans le cas de la complétion. En conséquence, la complétion vers des sous-classes des graphes cordaux semble être en général un problème plus facile à aborder que l'édition et la déletion.

Nous listons à présent divers problèmes de modifications d'arêtes pour lesquels l'existence d'un noyau polynomial est une question ouverte. Certains de ces problèmes sont ouverts depuis longtemps, les autres ont suscité moins d'attention mais semblent être de bons candidats pour poursuivre les travaux entamés dans cette thèse.

Le problème CHORDAL COMPLETION, plus connu sous le nom MINIMUM FILL-IN, fut un des premiers problèmes pour lequel l'existence d'un noyau polynomial a été montrée [84], cependant la question reste encore ouverte pour CHORDAL EDITION et CHORDAL DELETION. Remarquons que l'existence d'un algorithme FPT pour ces problèmes n'a été montrée que relativement récemment [33].

Problème ouvert. CHORDAL EDITION et CHORDAL DELETION admettent-ils un noyau polynomial ?

Une grande question ouverte dans le domaine des noyaux est l'existence d'un noyau polynomial pour le problème INTERVAL COMPLETION, question qui a été posée à plusieurs reprises [14, 15, 41, 127]. Une idée pour aborder ce problème peut être de considérer d'abord des variantes plus restreintes, par exemple en imposant que le graphe en entrée soit un graphe split ou un graphe cordal. Exiger de telles contraintes structurelles sur le graphe d'entrée pourrait aider à mettre au point des outils pour résoudre le problème plus général. Mentionnons que pour le problème INTERVAL EDITION aucun algorithme FPT n'est connu, il en est donc de même pour l'existence d'un noyau, même exponentiel.

Problème ouvert. INTERVAL COMPLETION et ses versions édition et déletion admettent-ils un noyau polynomial ?

Malgré le noyau polynomial pour PROPER INTERVAL COMPLETION [14], la question reste ouverte pour les versions édition et déletion. Notons que ces deux problèmes sont FPT [31].

Problème ouvert. PROPER INTERVAL EDITION et PROPER INTERVAL DELETION admettent-ils un noyau polynomial ?

Des problèmes pour lesquels l'approche « décomposition et sommets affectés » semble être une bonne candidate pour montrer l'existence d'un noyau polynomial sont les problèmes r -LEAF POWER COMPLETION et ses versions édition et déletion pour $r \geq 4$. Notamment, pour $r = 4$ le problème est FPT [50] et une caractérisation par une décomposition proche du graphe des cliques critiques est connue [25]. Néanmoins la structure est bien plus complexe que pour les graphes 3-leaf power ou strictement cordaux et les techniques que nous avons utilisées ne peuvent pas être adaptées de manière immédiate.

Problème ouvert. r -LEAF POWER COMPLETION et ses versions édition et déletion admettent-ils un noyau polynomial pour $r \geq 4$?

En s'éloignant légèrement des sous-classes des graphes cordaux, nous pouvons considérer les graphes distances héréditaires, qui n'admettent pas de cycle de longueur 5 sans cordes qui se croisent. De par cette proximité aux graphes cordaux, il nous semble intéressant de montrer l'existence ou non d'un noyau polynomial pour le problème DISTANCE HEREDITARY COMPLETION ainsi que ses versions édition et déletion. Le problème est FPT comme l'observent Crespelle *et al.* [41, Table 5] en se basant sur un résultat de Courcelle *et al.* [40].

Problème ouvert. DISTANCE HEREDITARY COMPLETION et ses versions édition et délétion admettent-ils un noyau polynomial ?

Pour terminer cette section, nous demandons s'il est possible de trouver des noyaux de plus petite taille pour les problèmes que nous avons étudiés dans le [chapitres 4 et 5](#). En particulier nous posons le problème ouvert suivant :

Problème ouvert. TRIVIALY PERFECT EDITION et THRESHOLD EDITION ainsi que leurs versions complétion et délétion admettent-ils un noyau avec $O(k^{2-\epsilon})$ sommets pour $\epsilon > 0$?

Notons que l'approche « décomposition et sommets affectés » ne semble pas permettre d'obtenir un tel noyau. En particulier avec cette approche, le [lemme 3.4](#) implique une borne quadratique sur le nombre de sommets affectés dans les cliques critiques. Obtenir une borne sous-quadratique nécessiterait de borner de manière sous-linéaire en k la taille des cliques critiques. Il serait également intéressant de trouver une borne inférieure sur la taille d'un noyau pour ces problèmes. Il existe de telles bornes inférieures pour des problèmes de suppression de sommets, par exemple les problèmes VERTEX COVER et FEEDBACK VERTEX SET n'admettent pas de noyau avec $O(k^{2-\epsilon})$ sommets pour tout $\epsilon > 0$ sous l'hypothèse $\text{coNP} \not\subseteq \text{NP}/\text{poly}$ [47]. À notre connaissance, aucune borne inférieure n'est connue sur la taille des noyaux pour les problèmes que nous avons considérés.

Partie II

Couverture de graphes

Chapitre 6

Couverture par plus courts chemins

Dans ce chapitre nous nous intéressons à divers problèmes de couverture de graphes par des plus courts chemins. Les résultats présentés dans ce chapitre sont issus de travaux réalisés avec Florent Foucaud, Anthony Perez et Ioan Todinca [55, 56]. Nous commençons par définir formellement les problèmes ISOMETRIC PATH COVER et STRONG GEODETIC SET, qui ont été mentionnés dans l'introduction de cette thèse et qui sont illustrés dans la [figure 6.1](#).

ISOMETRIC PATH COVER

Entrée : Un graphe G et un entier k .

Question : Existe-t-il un ensemble de k plus courts chemins tel que chaque sommet de G appartient à au moins un des chemins ?

STRONG GEODETIC SET

Entrée : Un graphe G et un entier k .

Question : Existe-t-il un ensemble de k terminaux et un ensemble de $\binom{k}{2}$ plus courts chemins, chaque chemin reliant une paire distincte de terminaux et tel que chaque sommet de G appartient à au moins un des chemins ?

Nous définissons également les versions de ces problèmes avec une donnée supplémentaire en entrée : les extrémités des chemins pour le premier, les terminaux pour le second.

ISOMETRIC PATH COVER WITH TERMINALS

Entrée : Un graphe G et k paires de terminaux $(s_1, t_1), \dots, (s_k, t_k)$.

Question : Existe-t-il un ensemble de k plus courts chemins tel que le i -ième chemin est un s_i - t_i plus court chemin et chaque sommet de G appartient à au moins un des chemins ?

STRONG GEODETIC SET WITH TERMINALS

Entrée : Un graphe G et k terminaux.

Question : Existe-t-il un ensemble de $\binom{k}{2}$ plus courts chemins, chaque chemin reliant une paire distincte de terminaux et tel que chaque sommet de G appartient à au moins un des chemins ?

Tous les problèmes que nous avons définis peuvent être déclinés en différentes versions, en demandant de couvrir toutes les arêtes du graphe et non uniquement les sommets ou bien en demandant que les plus courts chemins soient disjoints (dans ce cas, c'est un problème de partition).

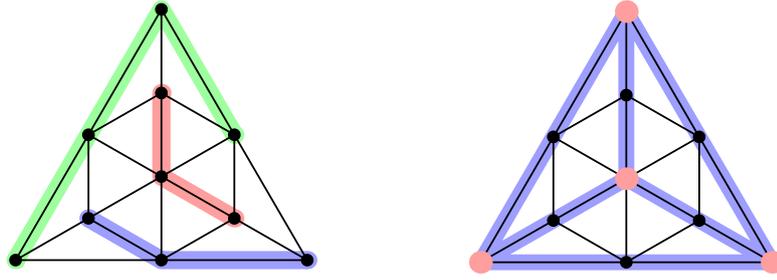


Figure 6.1: Exemple d'un graphe qui peut être couvert par 3 plus courts chemins (illustré à gauche), et qui a une solution avec 4 terminaux pour STRONG GEODETIC SET (illustré à droite).

Nous présentons maintenant les résultats de complexité connus pour ces problèmes. Il a été montré que ISOMETRIC PATH COVER est NP-complet, même pour les graphes cordaux [34]. Pour les graphes quelconques, il a été montré que le problème peut être approximé en temps polynomial avec un facteur de $O(\log d)$, où d est le diamètre du graphe d'entrée [125]. Il a été montré que le problème peut être résolu en temps polynomial sur les graphes blocs [116].

Le problème STRONG GEODETIC SET a été montré NP-complet [107], même pour les graphes (co)bipartis, les graphes cordaux et les graphes de diamètre 2 [46] ainsi que pour les graphes sous-cubiques de maille arbitraire [45]. Cependant il peut être résolu en temps polynomial sur les graphes planaires extérieurs [112], les graphes cactus, les graphes blocs et les graphes threshold [46]. La version couverture des arêtes du problème STRONG GEODETIC SET a elle aussi été montrée NP-complète [106].

Le problème STRONG GEODETIC SET WITH TERMINALS a été montré NP-complet [45] et ce même pour les graphes bipartis de degré maximum 4 ou diamètre 6, mais il peut être résolu en temps polynomial sur les graphes split, les graphes de diamètre 2, les graphes blocs et les graphes cactus [46]. Nous montrons dans le chapitre 6 que ISOMETRIC PATH COVER WITH TERMINALS est lui aussi NP-complet.

La version de ISOMETRIC PATH COVER où les chemins de la solution doivent être disjoints a aussi été étudiée sous les noms de ISOMETRIC PATH PARTITION [105] et SHORTEST PATH PARTITION [64]. Le problème est NP-complet [105], et il l'est également restreint aux graphes bipartis [64]. Fernau *et al.* [64] montrent que le problème est XP paramétré par le nombre de chemins dans les graphes non orientés ainsi que les graphes orientés acycliques (DAG). De plus, ils montrent que ISOMETRIC PATH COVER et SHORTEST PATH PARTITION sont FPT paramétrés par le paramètre *neighborhood diversity* (nombre de classes d'équivalences de voisinages) dans les graphes non orientés et orientés.

Contributions Nous montrons les théorèmes combinatoires suivants qui portent sur les graphes couvrables par k plus courts chemins (voir la section 2.1.4 pour la définition de largeur linéaire) :

Théorème 6.1. *Soit G un graphe dont l'ensemble d'arêtes peut être couvert par au plus k plus courts chemins. Alors la largeur linéaire de G est $O(3^k)$.*

Théorème 6.2. *Soit G un graphe dont l'ensemble de sommets peut être couvert par au plus k plus courts chemins. Alors la largeur linéaire de G est $O(k \cdot 3^k)$.*

Nous montrons en particulier le résultat suivant, qui permet d'obtenir simplement les bornes mentionnées ci-dessus.

Lemme 6.1. *Soit G un graphe dont l'ensemble d'arêtes (resp. de sommets) peut être couvert par k plus courts chemins. Pour tout sommet a de G et tout entier D , le nombre de sommets à distance exactement D du sommet a est $O(3^k)$ (resp. $O(k \cdot 3^k)$).*

De ce lemme découle un algorithme très simple en temps linéaire basé sur un parcours en largeur qui fournit une décomposition linéaire. La largeur de cette décomposition est bornée par le double de la borne sur le nombre de sommets dans un niveau du parcours en largeur. Notons

que la complexité de cet algorithme ne dépend pas de k ([lemme 6.2](#)). Nous pouvons remarquer que ces bornes sont aussi vérifiées pour le cas où les plus courts chemins doivent être disjoints sur les arêtes ou les sommets.

Outre les bornes combinatoires, nous montrons que le problème ISOMETRIC PATH COVER WITH TERMINALS peut se réduire à un problème d'optimisation EMSO₂. Rappelons qu'un problème EMSO₂ est un problème d'optimisation sur une formule de logique monadique du second ordre paramétrée par des ensembles de sommets et d'arêtes (cf. [section 2.2.3](#)).

Nous montrons notamment qu'il est possible d'exprimer en EMSO₂ un ensemble de k sous-graphes connexes couvrant le graphe tel que chaque paire de terminaux est contenue dans un de ces sous-graphes. De plus, la somme du nombre des arêtes de ces sous-graphes (que nous minimisons) est égale à la somme des distances entre chaque paire de terminaux si et seulement si ces sous-graphes sont des plus courts chemins entre les paires de terminaux. Cela nous permet avec l'aide du [théorème 2.4](#), qui indique que les problèmes EMSO₂ peuvent être résolus en temps linéaire pour les graphes de largeur arborescente bornée (et donc, de largeur linéaire bornée), en combinaison avec le [théorème 6.2](#), d'obtenir les résultats qui suivent.

Théorème 6.3. *ISOMETRIC PATH COVER WITH TERMINALS et STRONG GEODETIC SET WITH TERMINALS paramétrés par le nombre de terminaux sont FPT.*

Corollaire 6.4. *ISOMETRIC PATH COVER et STRONG GEODETIC SET paramétrés par le nombre de chemins ou le nombre de terminaux respectivement sont dans XP.*

Le [corollaire 6.4](#) est obtenu très simplement à partir du [théorème 6.3](#). Pour ISOMETRIC PATH COVER, on énumère en temps $n^{O(k)}$ les ensembles de k paires de terminaux possibles, et on applique l'algorithme FPT pour le problème ISOMETRIC PATH COVER WITH TERMINALS sur chaque combinaison. La technique pour STRONG GEODETIC SET est similaire, en énumérant tous les ensembles possibles de k terminaux.

Les algorithmes FPT du [théorème 6.3](#) pourraient également être obtenus par programmation dynamique, sans passer par le théorème de Courcelle. Toutefois ces algorithmes seraient fastidieux et pas particulièrement efficaces, en raison de la borne exponentielle de la largeur linéaire, c'est pourquoi nous préférons le cadre plus général proposé par la logique par souci de simplicité. En effet, grâce à la flexibilité de MSO₂ et au [théorème 6.1](#), les résultats algorithmiques peuvent facilement être étendus aux versions de couvertures des arêtes de nos problèmes. Ces résultats s'étendent aussi aux variantes où les chemins doivent être disjoints sur les sommets ou les arêtes, qui ont été étudiés dans [\[64, 105\]](#). La seconde partie du [théorème 6.3](#) répond positivement à une question ouverte posée dans [\[100\]](#).

Structure du chapitre Après quelques définitions et notions préliminaires dans la [section 6.1](#), nous montrons le [théorème 6.1](#) dans la [section 6.2](#). Plus précisément, la [section 6.2](#) fournit une borne supérieure sur la largeur linéaire des graphes dont les arêtes peuvent être couvertes par k plus courts chemins, puis dans la section suivante les outils sont étendus à la couverture des sommets. Les conséquences algorithmiques ([théorème 6.3](#)) sont détaillés dans la [section 6.4](#). Enfin nous concluons avec quelques questions ouvertes.

6.1 Préliminaires et notations

Pour plus de simplicité, dans ce chapitre, nous supposons les graphes étudiés connexes, cependant tous les résultats combinatoires et algorithmiques s'étendent pour les graphes non-connexes, en traitant séparément chaque composante connexe. Un graphe dont les sommets (resp. les arêtes) peuvent être couverts par k plus courts chemins est dit *sommet-couvrable* (resp. *arête-couvrable*) par k plus courts chemins.

Chemins et opérateurs de concaténation \oplus et \odot Un chemin (x_1, \dots, x_l) est appelé un x_1 - x_l chemin. Les chemins considérés ici sont simples dans le sens où ils n'utilisent pas le même sommet deux fois. Les sommets du chemin P sont notés par $V(P)$ et ses arêtes par $E(P)$. Étant

donné deux sommets $x, y \in V(P)$, $P[x, y]$ désigne le sous-chemin de P entre x et y . Dans ce chapitre les chemins sont implicitement orientés : un a - b chemin est pensé comme un chemin allant de a à b .

Nous utilisons deux types d'opérateurs sur les chemins, le premier concaténant un chemin après l'autre, le second fusionnant une des extrémités des deux chemins. Étant donné deux chemins disjoints sur les sommets, $\nu = (x_1, \dots, x_l)$ et $\eta = (y_1, \dots, y_t)$ de G tels que $\{x_l, y_1\}$ est une arête de G , l'opérateur de *concaténation* \oplus est défini comme suit :

$$\nu \oplus \eta = (x_1, \dots, x_l, y_1, \dots, y_t)$$

En particulier, $|\nu \oplus \eta| = |\nu| + |\eta| + 1$. De façon similaire, l'opérateur de *collage* entre deux chemins $\nu = (x_1, \dots, x_l)$ et $\eta = (x_l, y_1, \dots, y_t)$, avec $V(\nu) \cap V(\eta) = \{x_l\}$, est défini comme suit :

$$\nu \odot \eta = (x_1, \dots, x_l, y_1, \dots, y_t)$$

Dans ce cas là $|\nu \odot \eta| = |\nu| + |\eta|$. Le lemme suivant décrit comment obtenir une décomposition linéaire de largeur bornée à partir d'un parcours en largeur, nécessaire à la preuve des [théorèmes 6.1](#) et [6.2](#).

Lemme 6.2. *Soit $G = (V, E)$ un graphe, a un sommet de G et K une borne supérieure sur le nombre de sommets de G à distance exactement D de a , pour tout entier D . Alors, $\text{pw}(G) \leq 2K - 1$. De plus, une décomposition linéaire de largeur $2K - 1$ peut être calculée en temps linéaire par un parcours en largeur de G .*

Preuve. Soit $\text{ecc}(a) = \max_{x \in V} \text{dist}(a, x)$ l'excentricité du sommet a . Pour tout D avec $0 \leq D \leq \text{ecc}(a)$ l'ensemble de sommets à distance exactement D de a est noté Niveau(D), ce sont les niveaux d'un parcours en largeur de G à partir de a . En prenant en tant que sacs l'union de deux niveaux consécutifs Niveau(D) \cup Niveau($D+1$), pour $0 \leq D < \text{ecc}(a)$, et en les ordonnant en fonction de D on obtient une décomposition linéaire de G (voir la [figure 6.2](#)). En effet, pour chaque arête $\{x, y\}$, ses deux extrémités sont soit dans le même niveau, soit dans deux niveaux consécutifs, et donc apparaîtront dans le même sac. Pour chaque sommet x , il apparaît dans au plus deux sacs : si $d = \text{dist}(a, x)$ alors x est dans les sacs Niveau($d-1$) \cup Niveau(d) et Niveau(d) \cup Niveau($d+1$) (ou un seul niveau si $d = 0$ ou $d = \text{ecc}(a)$), et ces sacs apparaissent consécutivement dans la décomposition. Vu que chaque niveau contient au plus K sommets, la largeur de cette décomposition est au plus $2K - 1$. \square

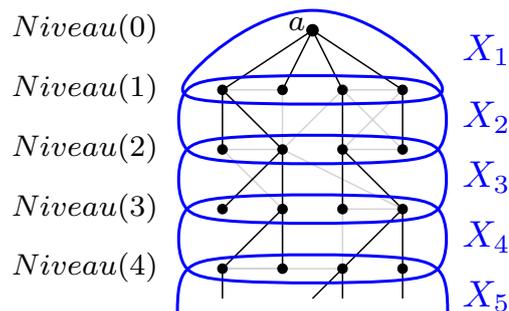


Figure 6.2: Illustration de la construction d'une décomposition linéaire d'un graphe à partir d'un parcours en largeur. Les ensembles X_i correspondent aux sacs de la décomposition.

NP-complétude Nous terminons cette section en montrant que le problème ISOMETRIC PATH COVER WITH TERMINALS est NP-complet. En particulier, nous montrons que le problème STRONG GEODETIC SET WITH TERMINALS s'y réduit, ce dernier étant connu NP-complet [45].

Proposition 6.1. *ISOMETRIC PATH COVER WITH TERMINALS est NP-complet.*

Preuve. Notons que le problème ISOMETRIC PATH COVER WITH TERMINALS est dans NP. Nous donnons une réduction depuis STRONG GEODETIC SET WITH TERMINALS. Soit $(G = (V, E), k)$ une instance de STRONG GEODETIC SET WITH TERMINALS avec les terminaux v_1, \dots, v_k . Nous construisons une instance de ISOMETRIC PATH COVER WITH TERMINALS en considérant toutes les $\binom{k}{2}$ paires possibles de terminaux, soit $(G = (V, E), \bigcup_{1 \leq i < j \leq k} \{(v_i, v_j)\})$. Par définition, (G, k) est une instance positive de STRONG GEODETIC SET WITH TERMINALS, c'est-à-dire il existe un ensemble de $\binom{k}{2}$ plus courts chemins couvrant $V(G)$ si et seulement si $(G, \bigcup_{1 \leq i < j \leq k} \{(v_i, v_j)\})$ est une instance positive de ISOMETRIC PATH COVER WITH TERMINALS, c'est-à-dire un ensemble de plus courts v_i - v_j chemins, $1 \leq i < j \leq k$ couvrant $V(G)$. Nous concluons que le problème ISOMETRIC PATH COVER WITH TERMINALS est NP-complet. \square

6.2 Couverture des arêtes avec k plus courts chemins

Dans cette section nous montrons le [théorème 6.1](#), bornant supérieurement la largeur linéaire des graphes arête-couvrables par k plus courts chemins. Un encodage simple des plus courts chemins permet d'obtenir une borne supérieure factorielle en k , cette preuve sera donnée dans la [section 6.2.1](#). Une borne simplement exponentielle, obtenue grâce à une analyse plus fine de l'encodage des plus courts chemins sera donnée dans la [section 6.2.2](#).

Dans cette section, $G = (V, E)$ désigne un graphe arête-couvrable par k plus courts chemins. Nous fixons un tel ensemble de chemins μ_1, \dots, μ_k , et les appelons les *chemins de base* de G . Toutes les constructions de cette section s'appuient sur cet ensemble de chemins de base particulier (sans le redéfinir pour chaque lemme afin de simplifier les notations). Nous dotons chaque chemin de base μ_c , $1 \leq c \leq k$ d'une *direction* arbitraire. Par exemple, en supposant que les sommets de G sont numérotés de 1 à n , la direction de P est du plus petit au plus grand sommet de ses extrémités. Un sous-chemin (orienté) $\mu_c[x, y]$ de μ_c est associé à un sens $+$ s'il suit la direction de μ_c , sinon il lui est associé un sens $-$. Pour chaque arête e de G , $\text{couleurs}(e)$ est l'ensemble des valeurs $c \in \{1, \dots, k\}$ tel que e est une arête de μ_c .

Bonne coloration Soit P un a - b chemin de G . Une *coloration* du chemin P est une fonction $\text{col}: E(P) \rightarrow \{1, \dots, k\}$ attribuant à chaque arête e du chemin une de ses couleurs $\text{col}(e) \in \text{couleurs}(e)$. La coloration col de P est dite *bonne* si, pour chaque couleur c , l'ensemble des arêtes de P utilisant cette couleur forme un sous-chemin connexe $P[x, y]$ de P (vu que nos chemins sont simples, cette condition implique que $P[x, y] = \mu_c[x, y]$). Un *chemin bien coloré* est une paire (P, col) formée par un chemin P et une bonne coloration de ce chemin.

L'opérateur \odot défini dans la [section 6.1](#) s'étend naturellement aux chemins colorés. Étant donné (P, col) , sa restriction au sous-chemin $P[x, y]$ de P est notée $(P[x, y], \text{col})$. Enfin, nous définissons pour un chemin P et une couleur $1 \leq c \leq k$ la fonction $\text{monochr}_c: E(P) \rightarrow \{c\}$. Toutes les arêtes du chemin coloré $(P, \text{monochr}_c)$ ont la couleur c .

Avec ces notations, tout a - b chemin bien coloré (P, col) avec les couleurs (c_1, \dots, c_l) apparaissant dans cet ordre là, peut être écrit comme :

$$(\mu_{c_1}[x_1, x_2], \text{monochr}_{c_1}) \odot (\mu_{c_2}[x_2, x_3], \text{monochr}_{c_2}) \odot \dots \odot (\mu_{c_l}[x_l, x_{l+1}], \text{monochr}_{c_l})$$

pour des sommets $a = x_1, x_2, x_3, \dots, x_l, x_{l+1} = b$. En d'autres termes, les $P[x_i, x_{i+1}]$ sont les sous-chemins monochromatiques de (P, col) , colorés c_i .

Lemme 6.3 (bonne coloration). *Pour toute paire de sommets a et b de G , il existe un a - b chemin bien coloré (P, col) tel que P est un plus court a - b chemin.*

Preuve. Parmi tous les plus courts a - b chemins, en choisir un qui admet une coloration avec un nombre minimum de sous-chemins monochromatiques. Soit P ce chemin, et col la coloration correspondante. Supposons par contradiction que la coloration n'est pas bonne. Alors il existe trois arêtes $e_1 = \{y_1, z_1\}$, $e_2 = \{y_2, z_2\}$ and $e_3 = \{y_3, z_3\}$, apparaissant dans cet ordre, telles que $\text{col}(e_1) = \text{col}(e_3) \neq \text{col}(e_2)$. Supposons sans perte de généralité que les sommets apparaissent dans l'ordre $y_1, z_1, y_2, z_2, y_3, z_3$ de a à b (il est possible que $z_1 = y_2$ ou $z_2 = y_3$). Soit $c = \text{col}(e_1) = \text{col}(e_3)$. En particulier, z_1 et y_3 sont sur le même chemin de base μ_c . Soit P' le chemin obtenu

à partir de P en remplaçant $P[z_1, y_3]$ par $\mu_c[z_1, y_3]$ (voir la figure 6.3). Premièrement, P' n'est pas plus long que P vu que $\mu_c[z_1, y_3]$ est un plus court z_1 - y_3 chemin du graphe G (P' n'a pas de sommets qui se répètent). Deuxièmement, dans P' nous pouvons colorer toutes les arêtes de $P'[z_1, y_3]$ avec la couleur c , et garder les couleurs de toutes les autres arêtes inchangées. Ainsi P' a strictement moins de sous-chemins monochromatiques que P , une contradiction. \square

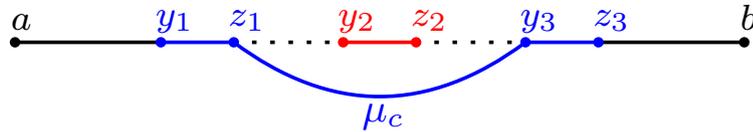


Figure 6.3: Illustration de la construction du chemin P' à partir du chemin P dans la preuve du lemme 6.3.

Soit (P, col) un a - b chemin bien coloré, avec les couleurs (c_1, \dots, c_l) dans cet ordre. Chaque sous-chemin monochromatique $P[x_i, x_{i+1}]$ de P , de couleur c_i , induit un sens sur le chemin de base μ_{c_i} correspondant (positif si $P[x_i, x_{i+1}]$ a la même direction que μ_{c_i} , négatif sinon). En conséquence, nous pouvons définir le *mot couleurs-sens* $\text{CouleurSens}(P, \text{col}) = ((c_1, s_1), (c_2, s_2), \dots, (c_l, s_l))$ sur l'alphabet $\{1, \dots, k\} \times \{+, -\}$, correspondant aux couleurs et aux sens des sous-chemins monochromatiques de P , selon l'ordre dans lequel ces sous-chemins apparaissent de a à b .

Nous pouvons observer sur la figure 6.4 qu'un même a - b chemin peut admettre plusieurs bonnes colorations (et éventuellement différents mots couleurs-sens), par exemple si un de ses sous-chemins est contenu dans plus d'un chemin de base. Similairement, deux plus courts chemins distincts entre des sommets a et b peuvent partager un même mot couleurs-sens. Enfin, notons qu'un même mot couleurs-sens peut être associé à des chemins ayant des extrémités différentes. Nous verrons dans la sous-section suivante qu'un mot couleurs-sens permet de définir un unique sommet à une distance fixée d'un sommet donné.

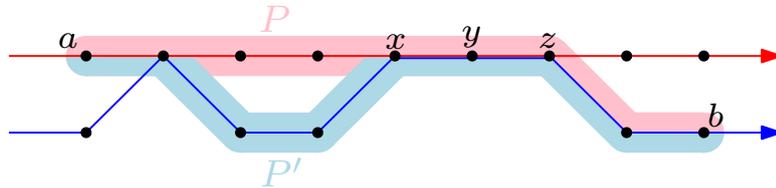


Figure 6.4: Deux a - b chemins bien colorés (P, col) et (P', col') avec le même mot couleurs-sens $\omega = ((\text{rouge}, +), (\text{bleu}, +))$ de même longueur. Le chemin P peut avoir plusieurs bonnes colorations, en fonction de $\text{col}(\{x, y\})$ et $\text{col}(\{y, z\})$ qui peuvent être *rouge* ou *bleu*. Le chemin P' n'admet qu'une seule bonne coloration.

6.2.1 Borne factorielle

Le mot couleurs-sens d'un chemin bien coloré a au plus k lettres sur un alphabet de taille $2k$. Par conséquent, le nombre de tels mots différents est borné supérieurement par une fonction de k .

Lemme 6.4. *Le nombre de mots couleurs-sens possibles, parmi tous les chemins bien colorés, est borné supérieurement par $h(k) = \sum_{l=1}^k 2^l \frac{k!}{(k-l)!}$.*

Preuve. Nous affirmons que le nombre de mots couleurs-sens de l lettres est borné supérieurement par $2^l \frac{k!}{(k-l)!}$. Observons que les couleurs forment un mot de longueur l sans répétition sur un alphabet de taille k . Le nombre de mots est $\frac{k!}{(k-l)!}$ (en choisissant l lettres parmi les k possibles, et en appliquant toutes les permutations possibles). Comme chaque lettre est également composée d'un sens $+$ ou $-$, nous multiplions ce nombre par 2^l , et la conclusion s'ensuit en additionnant sur toutes les valeurs possibles de l . \square

Le lemme suivant implique que, étant donné un sommet de départ a , une distance D et un mot couleurs-sens ω , il n'y qu'un seul sommet b à une distance D de a tel qu'un plus court a - b chemin bien coloré respecte le mot ω . Il permet de borner supérieurement le nombre de sommets à distance D de a par $h(k)$ la fonction du [lemme 6.4](#).

Lemme 6.5 (Encodage couleurs-sens). *Soit deux sommets b et c à même distance d'un sommet a de G . Soit (P, col) un plus court a - b chemin bien coloré et (P', col') un plus court a - c chemin bien coloré. Si $\text{CouleurSens}(P, \text{col}) = \text{CouleurSens}(P', \text{col}')$, alors $b = c$.*

Preuve. Nous procédons par induction sur le nombre de lettres du mot $\text{CouleurSens}(P, \text{col})$, que nous notons $\omega = ((c_1, s_1), (c_2, s_2), \dots, (c_l, s_l))$.

Soit $P[a, x_1]$ (resp. $P'[a, x'_1]$) le sous-chemin maximal de P (resp. P') de couleur c_1 partant de a . Supposons sans perte de généralité que $P[a, x_1]$ est au moins aussi long que $P'[a, x'_1]$. Puisque les deux sont des sous-chemins de μ_{c_1} , partant de a et ayant le même sens s_1 par rapport à μ_{c_1} , cela implique que $P'[a, x'_1]$ est contenu dans $P[a, x_1]$, en particulier x'_1 est entre a et x_1 dans P et dans μ_{c_1} (voir la [figure 6.5](#)).

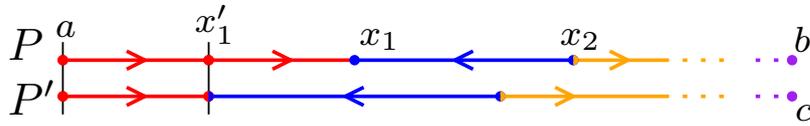


Figure 6.5: Illustration des chemins colorés P et P' dans la preuve du [lemme 6.5](#)

Observons que, si le mot ω n'a qu'une seule lettre, $P[a, x_1] = P$ et $P'[a, x'_1] = P'$, ils sont donc tous les deux de la même longueur. Comme ils sont de même sens par rapport à μ_{c_1} , cela implique que $x_1 = x'_1 = b = c$, ce qui prouve le cas de base de l'induction.

Supposons maintenant que ω a $l \geq 2$ lettres et que le lemme est vérifié pour les mots de longueur $l - 1$. Considérons premièrement le cas où $P[a, x_1]$ et $P'[a, x'_1]$ ont la même longueur. Observons que $x_1 = x'_1$ est aussi le premier sommet des sous-chemins de couleur c_2 de P et P' . Par conséquent, $(P[x'_1, b], \text{col})$ et $(P'[x'_1, c], \text{col}')$ sont des plus courts chemins bien colorés de même longueur et ont le même mot couleurs-sens $((c_2, s_2), \dots, (c_l, s_l))$, avec $l - 1$ lettres. Ainsi, $b = c$ par l'hypothèse d'induction.

Traisons maintenant le second et dernier cas, lorsque $P[a, x_1]$ est strictement plus long que $P'[a, x'_1]$. Soit x_2 le dernier sommet du sous-chemin coloré c_2 dans (P, col) . En particulier, x'_1, x_1 et x_2 sont tous des sommets de μ_{c_2} . Observons que sur μ_{c_2} , le sommet x_1 est situé entre x'_1 et x_2 . En effet, dans le chemin P , les sommets a, x'_1 et x_1 apparaissent dans cet ordre, et par construction x_1 apparaît entre a et x_2 . Donc a, x'_1, x_1, x_2 apparaissent dans cet ordre sur P , qui est un plus court chemin, ce qui implique $\text{dist}(x'_1, x_2) = \text{dist}(x'_1, x_1) + \text{dist}(x_1, x_2)$. Puisque les trois sommets x'_1, x_1, x_2 sont tous sur le plus court chemin μ_{c_2} , ils doivent y apparaître dans cet ordre. Par conséquent, $\mu_{c_2}[x'_1, x_2]$ induit le même sens s_2 sur μ_{c_2} que $P[x_1, x_2] = \mu_{c_2}[x_1, x_2]$. En particulier, dans le chemin :

$$(P[x'_1, b], \text{col}) = (P[x'_1, x_1], \text{monochr}_{c_1}) \odot (P[x_1, x_2], \text{monochr}_{c_2}) \odot (P[x_2, b], \text{col})$$

le premier sous-chemin $P[x'_1, x_1]$ coloré c_1 peut être remplacé par $\mu_{c_2}[x'_1, x_1]$, coloré c_2 , sans changer la longueur totale. Le chemin obtenu est un plus court x'_1 - b chemin bien coloré :

$$(\tilde{P}[x'_1, b], \tilde{\text{col}}) = (\mu_{c_2}[x'_1, x_2], \text{monochr}_{c_2}) \odot (P[x_2, b], \text{col}).$$

Son mot couleurs-sens est $((c_2, s_2), \dots, (c_l, s_l))$, le même que pour le plus court x'_1 - c chemin $(P'[x'_1, c], \text{col}')$. De plus, les deux chemins ont la même longueur, $|\tilde{P}| = |P[a, x'_1]|$, donc par hypothèse d'induction, $b = c$, ce qui prouve le lemme. \square

Corollaire 6.5. *Pour tout sommet a de G et tout entier D , il y au plus $h(k) = \sum_{l=1}^k 2^l \frac{k!}{(k-l)!}$ sommets à distance exactement D de a .*

Preuve. Pour tout sommet fixé a et entier fixé D , grâce au [lemme 6.5](#) le nombre de sommets à distance exactement D de a est borné supérieurement par le nombre de mots couleurs-sens, qui est lui même borné supérieurement par $h(k)$ par le [lemme 6.4](#). \square

Le [corollaire 6.5](#) et le [lemme 6.2](#) impliquent une version plus faible du [théorème 6.1](#) : la largeur linéaire d'un graphe arête-couvrable par k plus courts chemins est au plus $2h(k) - 1$, donnant une borne supérieure factorielle.

6.2.2 Borne simplement exponentielle

Dans cette section, nous montrons que le nombre de sommets à distance D d'un sommet a peut être borné par $O(3^k)$. Dans la section précédente nous avons montré que deux plus courts chemins bien colorés de longueur D commençant en a et avec le même mot couleurs-sens amènent au même sommet. Nous pouvons observer que des plus courts chemins avec des mots couleurs-sens différents peuvent aboutir sur un même sommet, voir la [figure 6.6](#). Dans cette section, nous prenons un ensemble de plus courts chemins bien colorés allant de a à tous les sommets à distance D de a et pour pouvoir borner leur nombre, nous essayons de les simplifier, dans le sens où nous utiliserons un nombre minimal de couleurs.

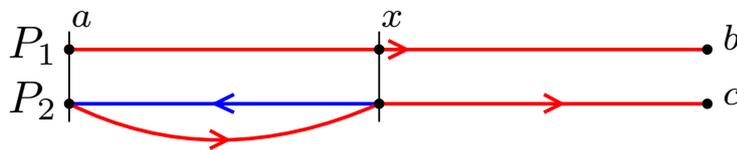


Figure 6.6: (P_1, col_1) et (P_2, col_2) sont des plus courts chemins bien colorés de même longueur. Le sommet x est partagé par les deux chemins. Le sous-chemin $\mu_{\text{bleu}}[a, x]$ in P_2 peut être remplacé par $\mu_{\text{rouge}}[a, x]$ sans augmenter la taille du chemin, prouvant que $b = c$.

Avant de rentrer dans les détails, nous donnons une description informelle de notre construction (qui est illustrée dans la [figure 6.9](#)). Soit un ensemble \mathcal{P} de plus courts chemins bien colorés, commençant tous en un sommet a mais terminant en des sommets différents. Ces sommets seront tous les sommets à une distance D de a . Nous allons transformer \mathcal{P} en une nouvelle famille de chemins, avec les mêmes propriétés, mais utilisant si possible un plus petit nombre de couleurs. Soit x_n le sommet le plus éloigné de a , contenu dans tous les chemins de \mathcal{P} et tel que ces chemins partagent tous le même sous-chemin coloré de a à x_n . Nous regardons ensuite l'ensemble \mathcal{C} des couleurs apparaissant sur les arêtes juste après x_n . Premièrement nous appliquons la transformation suivante : pour chaque chemin $(P, \text{col}) \in \mathcal{P}$, nous considérons l'arête la plus éloignée ayant sa couleur c dans \mathcal{C} (par exemple, dans la [figure 6.9](#), pour le chemin P_3 cette arête est colorée en rouge), et nous remplaçons tout le sous-chemin de P de x_n à cette arête par un sous-chemin de μ_c , de couleur c . Cette nouvelle famille de chemins est ensuite divisée en sous-familles en fonction de la couleur et du sens de la première arête après x_n , puis nous branchons et appliquons la même construction à chaque sous-famille. Nous finissons par prouver que ce processus de branchement correspond à un arbre (que nous appelons *arbre de branchement k -étiqueté*), dans le sens où les chemins construits sont en bijection avec les feuilles de l'arbre. De plus, nous allons borner le nombre de feuilles de l'arbre, et donc le nombre de chemins, à $O(3^k)$.

Nous définissons dans un premier temps les arbres de branchement k -étiquetés et montrons dans le [lemme 6.6](#) qu'ils ont $O(3^k)$ feuilles. Nous décrivons ensuite formellement le processus de branchement expliqué ci-dessus dans le [lemme 6.7](#), qui nous permet déduire la même borne supérieure sur le nombre de sommets à distance D de a .

Nous définissons un *arbre enraciné étiqueté* comme une paire (T, lab) où T est un arbre enraciné et $\text{lab}: E(T) \rightarrow \mathcal{CS}$ une fonction d'étiquetage des arêtes de T , où $\mathcal{CS} = \{1, \dots, k\} \times \{+, -\}$. Pour un noeud n de T , le sous-arbre de T enraciné en n est noté T_n . Pour un noeud n de T et ses enfants n_1, \dots, n_t , nous définissons les ensembles $\mathcal{L}(n) = \{\text{lab}(\{n, n_1\}), \dots, \text{lab}(\{n, n_t\})\}$ et $\mathcal{LS}(n) = \{(c, s), (c, \bar{s}) \mid (c, s) \in \mathcal{L}(n)\}$, où \bar{s} est le sens opposé de s .

Définition 6.1 (arbre de branchement k -étiqueté). Un arbre enraciné étiqueté (T, \mathbf{lab}) est appelé un *arbre de branchement k -étiqueté* si les propriétés suivantes sont vérifiées :

- (i) Dans tout chemin de la racine de T à une de ses feuilles, les arêtes avec la même étiquette forment un sous-chemin connexe,

Et pour tout noeud n de T avec pour enfants n_1, \dots, n_t :

- (ii) Les étiquettes sur les arêtes $\{n, n_i\}, i \in \{1, \dots, t\}$ sont deux à deux distinctes,
 (iii) Pour tout $i \in \{1, \dots, t\}$, T_{n_i} ne contient pas d'arête avec une étiquette dans $\mathcal{LS}(n) \setminus \{\mathbf{lab}(\{n, n_i\})\}$.

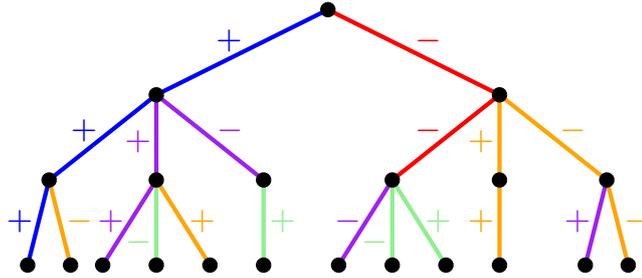


Figure 6.7: Exemple d'un arbre de branchement k -étiqueté. L'étiquette d'une arête est la paire composée de la couleur de l'arête et du sens $+$ ou $-$ associé.

Tout d'abord, nous montrons une borne combinatoire sur le nombre de feuilles dans un arbre de branchement k -étiqueté. Nous pouvons mentionner que dans un cadre plus général où la fonction d'étiquetage \mathbf{lab} est définie sur un alphabet Σ et où la deuxième condition est remplacée par « T_{n_i} ne contient aucune arête avec une étiquette dans $\mathcal{L}(n) \setminus \{\mathbf{lab}(\{n, n_i\})\}$ », une borne $O(2^{|\Sigma|})$ peut être obtenue plus simplement, soit $O(4^k)$ dans notre cas (dans cette version de la deuxième condition, \mathcal{LS} a simplement été remplacé par \mathcal{L}). Cependant, en exploitant les spécificités de l'alphabet \mathcal{CS} , nous pouvons obtenir la borne qui suit.

Lemme 6.6. *Un arbre de branchement k -étiqueté (T, \mathbf{lab}) contient $O(3^k)$ feuilles.*

Preuve. Soit (T, \mathbf{lab}) un arbre de branchement k -étiqueté, avec la fonction d'étiquetage $\mathbf{lab} : E(T) \rightarrow \mathcal{CS}$. Soit $\mathcal{B}(n)$ l'ensemble des étiquettes qui n'ont pas été interdites dans T_n lors de l'application de la condition (i) ou (iii) de la définition 6.1 à un ancêtre de n dans l'arbre T . En particulier, si n est la racine, alors $\mathcal{B}(n) = \mathcal{CS}$, et à tout noeud n , $\mathcal{B}(n)$ est un sur-ensemble de toutes les étiquettes des arêtes de T_n . Si n a exactement un enfant n' , par définition $\mathcal{B}(n') \subseteq \mathcal{B}(n)$ donc $|\mathcal{B}(n')| \leq |\mathcal{B}(n)|$.

Affirmation 6.1. Soit n un noeud de T avec $t \geq 2$ enfants n_1, \dots, n_t . Si n est la racine, alors $|\mathcal{B}(n_i)| \leq |\mathcal{B}(n)| - (t - 1)$, sinon $|\mathcal{B}(n_i)| \leq |\mathcal{B}(n)| - \max\{2, t - 1\}$.

Preuve. Puisque T est un arbre de branchement k -étiqueté, par la condition (ii) les étiquettes des arêtes $\{n, n_j\}, 1 \leq j \leq t$ sont distinctes deux à deux. En particulier, les $t - 1$ étiquettes distinctes des arêtes $\{n, n_j\}, j \neq i$, apparaissent dans $\mathcal{B}(n)$ mais pas dans $\mathcal{B}(n_i)$ par la condition (iii). Par conséquent, $|\mathcal{B}(n_i)| \leq |\mathcal{B}(n)| - (t - 1)$.

Il reste à prouver que si n n'est pas la racine et a exactement deux enfants n_1 et n_2 , alors $|\mathcal{B}(n_i)| \leq |\mathcal{B}(n)| - 2$ pour $i \in \{1, 2\}$. Soit p le noeud parent de n dans T , et définissons $(c, s) = \mathbf{lab}(\{p, n\})$ et $(c_i, s_i) = \mathbf{lab}(\{n, n_i\})$ pour $i \in \{1, 2\}$ (voir la figure 6.8).

Nous affirmons que si $c_i \neq c$ alors (c_i, s_i) et (c_i, \bar{s}_i) sont toutes deux dans $\mathcal{B}(n)$. En effet, dans ce cas là, la condition (i) implique que l'étiquette (c_i, s_i) n'apparaît pas sur le chemin de n à la racine de T . Par la condition (iii), l'étiquette (c_i, \bar{s}_i) n'apparaît pas non plus : cela interdirait l'étiquette (c_i, s_i) dans l'arbre en dessous et en particulier dans T_n . Les étiquettes (c_i, s_i) et (c_i, \bar{s}_i) n'apparaissent donc pas sur le chemin menant de n à la racine de T . Par la condition (iii), pour toute arête $\{n', n''\}$ de T telle que n' est un ancêtre de n et n'' n'est pas un ancêtre de

n (voir la figure 6.8), c_i ne peut pas apparaître dans l'étiquette $\text{lab}(\{n', n''\})$ autrement (c_i, s_i) et (c_i, \bar{s}_i) seraient interdites dans le sous-arbre de $T_{n'}$ contenant n (qui ne contient pas n''). En particulier, (c_i, s_i) ne pourrait pas être utilisée comme étiquette de $\{n, n_i\}$. Cela veut dire que la couleur c_i n'apparaît pas sur les arêtes incidentes au chemin allant de n à la racine de T . Nous pouvons donc en conclure que (c_i, s_i) et (c_i, \bar{s}_i) sont dans $\mathcal{B}(n)$ puisqu'elles ne peuvent pas être interdites par les conditions (i) ou (iii).

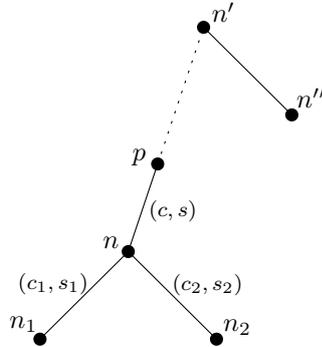


Figure 6.8: Illustration de la configuration obtenue quand n n'est pas la racine, a au moins deux enfants n_1 , et n_2 et p comme parent. Le noeud n' est un ancêtre de n tandis que n'' non.

Supposons sans perte de généralité que $i = 1$. Si $c_1 = c$ alors nous affirmons que $c_2 \neq c$. En effet, par la condition (ii), si $c_1 = c_2 = c$ alors les sens s_1 et s_2 sont opposés, donc l'un est l'opposé \bar{s} de s , contredisant que l'étiquette (c, \bar{s}) est dans $\mathcal{LS}(p)$ et donc a été interdite dans T_n par la condition (iii). Par conséquent (c_2, s_2) et (c_2, \bar{s}_2) sont dans $\mathcal{B}(n)$ et par la condition (iii) sont interdites dans T_{n_1} et donc $|\mathcal{B}(n_1)| \leq |\mathcal{B}(n)| - 2$.

Si $c_1 \neq c$, alors (c, s) n'est pas dans $\mathcal{B}(n_1)$ par la condition (i) mais est dans $\mathcal{B}(n)$ et (c_1, \bar{s}_1) n'est pas non plus dans $\mathcal{B}(n_1)$ par la condition (iii) mais est dans $\mathcal{B}(n)$, ainsi $|\mathcal{B}(n_1)| \leq |\mathcal{B}(n)| - 2$. \diamond

Nous prouvons maintenant que T a $O(3^k)$ feuilles. Pour un noeud n de l'arbre, nous voyons $|\mathcal{B}(n)|$ comme le *budget* du sous-arbre T_n et estimons le nombre de ses feuilles en fonction de ce budget. Nous rappelons que le budget de la racine est $2k$. Soit $F(b)$ le nombre maximum de feuilles dans n'importe quel arbre de branchement k -étiqueté avec un budget b a sa racine. Nous pouvons observer que si $b = 1$ alors $F(b) = 1$ et pour $b \leq b', F(b) \leq F(b')$. Soit n un noeud de T , alors $F(|\mathcal{B}(n)|)$ est le nombre maximum de feuilles de T_n . Si n a exactement un fils, il ne branche pas, donc $F(|\mathcal{B}(n_1)|) \leq F(|\mathcal{B}(n)|)$ puisque $\mathcal{B}(n_1) \subseteq \mathcal{B}(n)$. Supposons que n a $t \geq 2$ enfants n_1, \dots, n_t . Si n est la racine, alors par l'affirmation 6.1 :

$$F(|\mathcal{B}(n)|) \leq F(|\mathcal{B}(n_1)|) + \dots + F(|\mathcal{B}(n_t)|) \leq t \cdot F(|\mathcal{B}(n)|) - (t - 1)$$

Sinon si n n'est pas la racine :

$$F(|\mathcal{B}(n)|) \leq t \cdot F(|\mathcal{B}(n)|) - \max\{2, t - 1\}.$$

À la racine r de l'arbre, puisque $\mathcal{B}(r) = \mathcal{CS}$, nous avons $|\mathcal{B}(r)| = 2k$. Nous allons montrer que $F(2k) = O(3^k)$ à l'aide du théorème 2.5. Nous avons les récurrences $F(b) \leq t \cdot F(b - (t - 1))$, et $F(b) \leq 2 \cdot F(b - 2)$ (sauf à la racine, si la racine a deux enfants). Chacune de ces récurrences donne un facteur de branchement différent : $\alpha_t = \sqrt[t]{t}$ pour la première, et $\alpha_2 = \sqrt{2}$ pour la seconde. Observons que la valeur maximale de $\alpha_2 = \sqrt{2}$ et $\alpha_t = \sqrt[t]{t}$, $t \geq 3$ est atteinte pour $t = 3$ avec $\alpha_3 = \sqrt{3}$. Par conséquent, d'après le théorème 2.5, $F(b) = O(3^{b/2})$. Ceci est également valable si la racine a un ou deux enfants. En effet, pour chaque enfant de la racine, le nombre de feuilles dans le sous-arbre correspondant est $O(3^{b/2})$, et comme il y a au plus deux enfants de ce type, le nombre total de feuilles est $O(3^{b/2})$. Puisqu'à la racine $b = 2k$, nous concluons qu'un arbre de branchement k -étiqueté a $O(3^k)$ feuilles. \square

Nous détaillons maintenant la description haut niveau donnée au début de la section. Par un léger abus de notation, pour un chemin coloré (P, col) , nous définissons

$$\text{couleurs}(P, \text{col}) = \{\text{col}(e) \mid e \in E(P)\}$$

l'ensemble des couleurs apparaissant sur (P, col) . De la même manière, pour un ensemble de chemins colorés \mathcal{P} nous définissons l'ensemble

$$\text{couleurs}(\mathcal{P}) = \bigcup_{(P, \text{col}) \in \mathcal{P}} \text{couleurs}(P, \text{col})$$

Pour un a - b chemin coloré (P, col) et un sommet x de P , nous définissons $P[x:] = P[x, b]$ et pour un ensemble de chemins colorés \mathcal{P} partageant un sommet x nous définissons $\mathcal{P}[x:] = \bigcup_{(P, \text{col}) \in \mathcal{P}} (P[x:], \text{col})$.

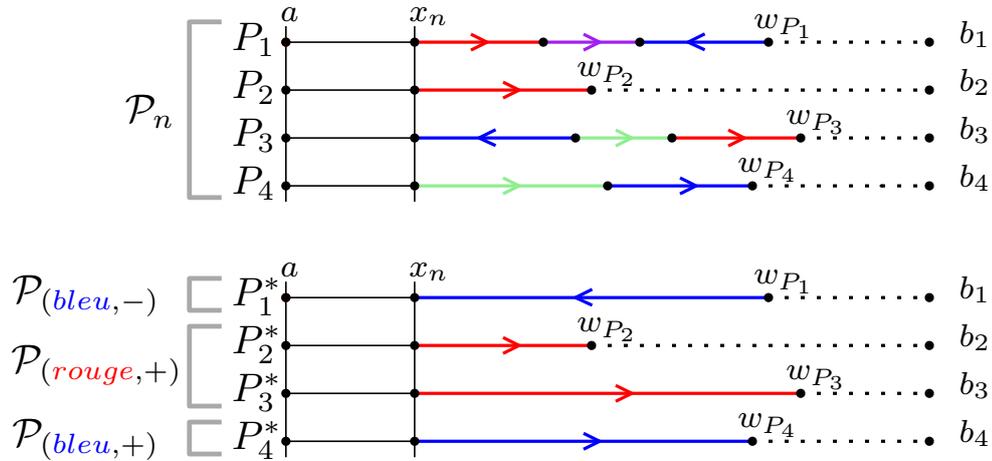


Figure 6.9: Exemple de la construction dans la preuve du lemme 6.7 pour un ensemble de chemins \mathcal{P}_n . Dans cet exemple, tous les chemins partagent le même sous-chemin $(P_1[a, x_n], \text{col}_1)$, $\mathcal{C}_n = \{\text{rouge}, \text{bleu}, \text{vert}\}$ et il n'y a pas d'arêtes avec des couleurs dans \mathcal{C}_n dans les chemins $(P_i[w_{P_i}:], \text{col}_i)$. Les ensembles $\mathcal{P}_{(\text{rouge}, -)}$, $\mathcal{P}_{(\text{vert}, +)}$, $\mathcal{P}_{(\text{vert}, -)}$ sont vides.

Lemme 6.7. *Pour tout sommet a de $V(G)$ et tout entier D , il y a $O(3^k)$ sommets à distance exactement D de a .*

Preuve. Soit $\{b_1 \dots, b_\ell\}$ l'ensemble des sommets à distance D de a dans G et soit \mathcal{P} un ensemble $\{(P_1, \text{col}_1), \dots, (P_\ell, \text{col}_\ell)\}$ de plus courts a - b_i chemins bien colorés. Notre objectif est de construire un arbre de branchement k -étiqueté (T, lab) tel que :

- chaque noeud n de T est associé à un ensemble de chemins \mathcal{P}_n ,
- pour chaque sommet b_i , il existe une feuille l de T telle que \mathcal{P}_l contient un plus court a - b_i chemin bien coloré,
- pour chaque feuille l , \mathcal{P}_l contient exactement un chemin.

Nous allons construire (T, lab) récursivement en commençant par la racine r de T . Initialement, $\mathcal{P}_r = \mathcal{P}$. Pour chaque noeud n de T , \mathcal{P}_n sera un ensemble de plus courts chemins bien colorés allant de a à un sous-ensemble de sommets de $\{b_1 \dots, b_\ell\}$. Soit n un noeud déjà calculé de T . Si $|\mathcal{P}_n| = 1$, ne rien faire : n est une feuille de l'arbre et aucun appel récursif ne sera effectué à partir de n . Si $|\mathcal{P}_n| > 1$, nous développons l'arbre à partir de n . Les ensembles de chemins qui seront associés aux enfants de n sont construits comme suit :

1. Soit x_n le sommet à la distance maximum de a tel que pour toute paire de chemins colorés $(P, \text{col}), (P', \text{col}') \in \mathcal{P}_n$, $(P[a, x_n], \text{col}) = (P'[a, x_n], \text{col}')$. L'existence de ce sommet est garantie, x_n peut être le sommet a .

2. Soit $\mathcal{C}_n = \{\text{col}(x_n, y_P) \mid (P, \text{col}) \in \mathcal{P}_n\}$ où $\{x_n, y_P\}$ est la première arête de $P[x_n:]$.
3. Pour chaque $(P, \text{col}) \in \mathcal{P}_n$, soit $\{z_P, w_P\}$ l'arête la plus éloignée de x_n sur $P[x_n:]$ dont la couleur est dans \mathcal{C}_n , et soit $c_P = \text{col}(\{z_P, w_P\})$. Construire le chemin :

$$(P^*, \text{col}^*) = (P[a, x_n], \text{col}) \odot (\mu_{c_P}[x_n, w_P], \text{monochr}_{c_P}) \odot (P[w_P:], \text{col})$$

4. Pour chaque $(c, s) \in \mathcal{CS}_n = (\mathcal{C}_n \times \{+, -\})$, soit s_P le sens de $\mu_{c_P}[x_n, w_P]$. Définir l'ensemble :

$$\mathcal{P}_{(c,s)} = \{(P^*, \text{col}^*) \mid (P, \text{col}) \in \mathcal{P}_n, (c_P, s_P) = (c, s)\}$$

Cette construction est illustrée dans la [figure 6.9](#). Ensuite, pour chaque ensemble non vide $\mathcal{P}_{(c,s)}, (c, s) \in \mathcal{CS}_n$, créer un enfant n' de n associé à $\mathcal{P}_{(c,s)}$, où l'arête $\{n, n'\}$ est étiquetée (c, s) . Enfin, le processus est appliqué récursivement à chacun des noeuds créés.

Affirmation 6.2. Pour tout noeud n de T , \mathcal{P}_n est un ensemble de plus courts chemins bien colorés.

Preuve. L'affirmation est trivialement vérifiée pour \mathcal{P}_r où r est la racine de T . Nous montrons que pour un noeud n de T , si $(P, \text{col}) \in \mathcal{P}_n$ est un plus court a - b_i chemin bien coloré, alors le chemin $(P^*, \text{col}^*) = (P[a, x_n], \text{col}) \odot (\mu_{c_P}[x_n, w_P], \text{monochr}_{c_P}) \odot (P[w_P:], \text{col}_i)$ construit dans le [point 3](#) est un plus court a - b_i chemin bien coloré. Il est clair d'après la construction que P' est un a - b_i chemin, de plus c'est aussi un plus court chemin puisque $\mu_{c_P}[x_n, w_P]$ est un plus court x_n - w_P chemin.

Vu que (P, col) est bien coloré, $(P[a, x_n], \text{col})$ et $(P[w_P:], \text{col})$ le sont aussi. Puisque $\{z_P, w_P\}$ est la dernière arête colorée c_P de (P, col) , cette couleur n'apparaît pas dans $(P[w_P:], \text{col})$. De plus, si $c_P \in \text{couleurs}(P[a, x_n])$, il doit s'agir de la dernière couleur à apparaître, sinon (P, col) ne serait pas bien coloré. Par conséquent le chemin (P^*, col^*) est bien coloré. Donc, pour chaque $(c, s) \in \mathcal{CS}_n$, l'ensemble $\mathcal{P}_{(c,s)}$ associé à un enfant n' de n est un ensemble de plus courts chemins bien colorés. Ainsi, par induction, pour tout noeud n de T , \mathcal{P}_n est un ensemble de plus courts chemins bien colorés. \diamond

Nous montrons maintenant que la construction se termine. En particulier, nous montrons que pour tout noeud non feuille n et tout enfant non feuille n_i , nous avons $\text{dist}(a, x_n) < \text{dist}(a, x_{n_i})$. Vu que pour tout noeud n de T , les chemins des ensembles \mathcal{P}_n sont des plus courts chemins de longueur D , cela implique que la construction se termine. Par construction (voir [point 3](#)), pour $(P', \text{col}') \in \mathcal{P}_{n_i}$, il existe un chemin $(P, \text{col}) \in \mathcal{P}_n$ tel que $(P', \text{col}') = (P^*, \text{col}^*)$, impliquant que $(P[a, x_n], \text{col}) = (P'[a, x_n], \text{col}')$. De plus, supposons que $\{x_n, y\}$ soit la première arête de $(P'[x_n:], \text{col}')$. Par définition de $\mathcal{P}_{n_i} = \mathcal{P}_{(c_P, s_P)}$ dans le [point 4](#), tous les chemins de \mathcal{P}_{n_i} partagent cette même arête et donc le sous-chemin $(P'[a, y], \text{col}')$. Par conséquent, $\text{dist}(a, x_n) < \text{dist}(a, y) \leq \text{dist}(a, x_{n_i})$ puisque les chemins de \mathcal{P}_n et \mathcal{P}_{n_i} sont des plus courts chemins par l'[affirmation 6.2](#). Il est possible de prouver que la hauteur de T est au plus k , mais ce n'est pas nécessaire.

L'observation suivante est vérifiée si n' est un enfant de n par les arguments précédents, et donc est aussi vérifiée pour tout noeud de T_n par induction.

Observation 6.1. Soit n un noeud de T et $n' \neq n$ un noeud de T_n . Alors, $\text{dist}(a, x_n) < \text{dist}(a, x_{n'})$, et pour $(P, \text{col}) \in \mathcal{P}_n$ et $(P', \text{col}') \in \mathcal{P}_{n'}$, $(P[a, x_n], \text{col}) = (P'[a, x_n], \text{col}')$.

Observation 6.2. Soit $\{n, n'\}$ une arête de T et $(P, \text{col}) \in \mathcal{P}_n$. Alors, la première lettre de $\text{CouleurSens}(P[x_n:], \text{col})$ est $\text{lab}(\{n, n'\})$. Si n' est un noeud non feuille, il peut y avoir plus d'une lettre dans $\text{CouleurSens}(P[x_n, x_{n'}], \text{col})$.

Affirmation 6.3. (T, lab) est un arbre de branchement k -étiqueté.

Preuve. Soit n un noeud de T et n_1, \dots, n_t ses enfants. La condition (ii) de la [définition 6.1](#) est vérifiée, en effet pour chaque $(c, s) \in \mathcal{CS}_n$, au plus une arête $\{n, n_i\}$ est étiquetée (c, s) . Nous allons premièrement prouver la condition (iii), soit que T_{n_i} ne contient aucune arête avec une étiquette dans $\mathcal{LS}(n) \setminus \{\text{lab}(\{n, n_i\})\}$. Rappelons que $\mathcal{LS}(n) = \{(c, s), (c, \bar{s}) \mid (c, s) \in \mathcal{L}(n)\}$, où

$\mathcal{L}(n) = \{\text{lab}(\{n, n_1\}), \dots, \text{lab}(\{n, n_t\})\}$. Nous pouvons observer que $\mathcal{LS}(n) \subseteq \mathcal{CS}_n$ (il peut y avoir des ensembles $\mathcal{P}_{(c,s)}$ vides), et que \mathcal{CS}_n est défini à partir de l'ensemble de couleurs \mathcal{C}_n qui apparaissent sur la première arête après x_n dans les chemins de \mathcal{P}_n .

Soit n_i un enfant de n et soit $(c_i, s_i) = \text{lab}(\{n, n_i\})$. Le cas où n_i est une feuille est trivial, puisque T_{n_i} n'a pas d'arêtes. Supposons donc que n_i n'est pas une feuille. Observons que pour tout $(P, \text{col}) \in \mathcal{P}_n$, par le choix de w_P , la seule couleur de \mathcal{C}_n utilisée dans $(P^*[x_n:], \text{col}^*)$ est c_P . Formellement, nous avons :

$$\text{couleurs}(P^*[x_n:], \text{col}^*) \subseteq \text{couleurs}(P[x_n:], \text{col}) \setminus (\mathcal{C}_n \setminus \{c_P\})$$

et donc, pour $1 \leq i \leq t$:

$$\text{couleurs}(\mathcal{P}_{n_i}[x_n:]) \subseteq \text{couleurs}(\mathcal{P}_n[x_n:]) \setminus (\mathcal{C}_n \setminus \{c_i\})$$

Par induction, cela implique que pour tout noeud n' de T_{n_i} , nous avons :

$$\text{couleurs}(\mathcal{P}_{n'}[x_n:]) \subseteq (\text{couleurs}(\mathcal{P}_n[x_n:]) \setminus (\mathcal{C}_n \setminus \{c_i\}))$$

Par l'observation 6.2, l'étiquette d'une arête $\{n', n''\}$ de T_{n_i} dépend des couleurs qui apparaissent après $x_{n'}$ dans les chemins de $\mathcal{P}_{n''}$, ainsi aucune arête de T_{n_i} a une étiquette dans $(\mathcal{C}_n \setminus \{c_i\}) \times \{+, -\}$.

Nous prouvons maintenant que (c_i, \bar{s}_i) n'est pas utilisée en tant qu'étiquette pour les arêtes de T_{n_i} . Soit $\{n', n''\}$ une arête de T_{n_i} et supposons par contradiction que $\text{lab}(\{n', n''\}) = (c_i, \bar{s}_i)$. Soit $(P, \text{col}) \in \mathcal{P}_{n_i}$ et $(P', \text{col}') \in \mathcal{P}_{n''}$, par l'observation 6.1, $(P[x_n, x_{n_i}], \text{col}) = (P'[x_n, x_{n_i}], \text{col}')$. Par l'observation 6.2, la première lettre de $\text{CouleurSens}(P'[x_n:], \text{col}')$ est (c_i, s_i) et la première lettre de $\text{CouleurSens}(P'[x_{n'}:], \text{col}')$ est (c_i, \bar{s}_i) , ce qui est une contradiction car le chemin (P', col') est bien coloré (affirmation 6.2). Ainsi, il n'y a pas d'arêtes de T_{n_i} avec des étiquettes dans $\mathcal{CS}_n \setminus \{(c_i, s_i)\}$, prouvant la condition (iii).

Il reste à montrer la condition (i), soit que dans chaque chemin de la racine de T à une de ses feuilles, les arêtes qui partagent la même étiquette forment un sous-chemin connexe. Soit $r = p_1, p_2, \dots, p_t = n$ le chemin de la racine r à un noeud non feuille n de T . S'il existe un enfant n' de n tel que $\text{lab}(\{n, n'\}) = \text{lab}(\{p_{i-1}, p_i\}) = (c, s), i \in \{2, \dots, t\}$, alors nous devons montrer que pour tout $i < j \leq t$, $\text{lab}(\{p_{j-1}, p_j\}) = (c, s)$. Supposons qu'il existe de tels n' et i . Pour $(P, \text{col}) \in \mathcal{P}_{p_i}$ et $(P', \text{col}') \in \mathcal{P}_{n'}$, par l'observation 6.1, $(P[x_{p_{i-1}}, x_{p_i}], \text{col}) = (P'[x_{p_{i-1}}, x_{p_i}], \text{col}')$. Par l'observation 6.2, la première arête de ces sous-chemins est colorée c , de même que la première arête de $(P'[x_n:], \text{col}')$. Puisque (P', col') est bien coloré, tout le sous-chemin $(P'[x_{p_{i-1}}, x_n], \text{col}')$ est entièrement coloré avec c . Pour tout $i < j \leq t$, $(P''[x_{p_{j-1}}, x_{p_j}], \text{col}'') = (P'[x_{p_{j-1}}, x_{p_j}], \text{col}')$, avec $(P'', \text{col}'') \in \mathcal{P}_{p_j}$, et cela implique que la première couleur de $(P''[x_{p_{j-1}}, x_{p_j}], \text{col}'')$ est c . Donc $\text{lab}(\{p_{j-1}, p_j\}) = (c, s)$ et ainsi la condition (i) est vérifiée. Par conséquent, (T, col) est un arbre de branchement k -étiqueté, ce qui conclut la preuve de l'affirmation 6.3. \diamond

Pour un noeud non feuille n de T , nous pouvons observer que pour tout a - b_i chemin dans \mathcal{P}_n , $b_i \in \{b_1, \dots, b_\ell\}$, il existe un de ses enfants n' tel que $\mathcal{P}_{n'}$ contient un a - b_i chemin. Cela implique que pour tout b_i , il existe une feuille l de T telle que \mathcal{P}_l contient un plus court a - b_i chemin bien coloré. Puisque T est un arbre de branchement k -étiqueté, par le lemme 6.6, il a $O(3^k)$ feuilles et pour chaque feuille l de T , $|\mathcal{P}_l| = 1$ (sinon notre construction aurait créé un enfant pour le noeud l). Cela implique qu'il y a $O(3^k)$ sommets à la distance D de a . \square

Pour finir la preuve du théorème 6.1 et montrer que $\text{pw}(G) = O(3^k)$, il suffit d'appliquer le lemme 6.2 avec $K = O(3^k)$.

6.3 Couverture des sommets avec k plus courts chemins

Dans cette section, $G = (V, E)$ désigne un graphe sommet-couvrable par k plus courts chemins μ_1, \dots, μ_k , appelés *chemins de base*. Nous montrons comment obtenir le théorème 6.2, bornant supérieurement la largeur linéaire des graphes sommet-couvrables par k plus courts chemins.

Comme précédemment, nous dotons d'une direction chaque chemin de base μ_c , mais cette fois ci les couleurs sont assignées aux sommets. La notion de bonne coloration s'adapte directement dans le cas de la couverture des sommets. Il est toujours possible de trouver un plus court chemin bien coloré entre deux sommets (lemme 6.8), définissant ainsi un mot couleurs-sens pour un chemin bien coloré.

La différence majeure avec le cas de la couverture des arêtes est que le lemme 6.5 (encodage couleurs-sens) n'est pas vérifié. En effet, deux sommets distincts b et c à la même distance D de a , et des plus courts chemins bien colorés $a-b$ et $a-c$ peuvent avoir le même mot couleurs-sens (voir la figure 6.10). La conséquence de cela est que nous ne pouvons pas directement adapter la borne factorielle obtenue dans le cas arête pour le cas sommet.

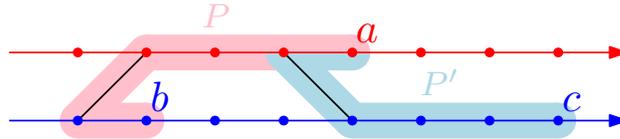


Figure 6.10: Deux chemins bien colorés (P, col) et (P', col') avec le même mot couleurs-sens $\omega = ((\text{rouge}, -), (\text{bleu}, +))$, de même longueur 5 et même sommet de départ a mais avec des sommets d'arrivée différents (b et c).

Dans la première version des résultats présentés dans ce chapitre [55], seule la borne factorielle avait été donnée dans le cas des graphes arête-couvrables. Pour le cas des graphes sommet-couvrables, une preuve spécifique, utilisant toujours la notion de mot couleurs-sens, avait été proposée pour obtenir également une borne factorielle. L'idée pour obtenir cette borne est de définir une façon unique de construire un chemin étant donné un sommet, un mot couleurs-sens et une distance. Les chemins ainsi définis ne sont pas nécessairement des plus courts chemins, mais nous pouvons montrer que leur longueur n'excède pas celle d'un plus court chemin, plus $2k$. Cela permet notamment de garantir qu'un mot couleurs-sens ne peut définir qu'un nombre linéaire en k de sommets à une distance fixée d'un sommet donné. De cette manière on obtient une borne factorielle $O(k \cdot h(k))$ dans le cas des graphes sommet-couvrables, où $h(k)$ est, comme précédemment, le nombre de mots couleurs-sens.

Nous présentons ici uniquement la borne simplement exponentielle pour le cas des graphes sommets-couvrables. L'approche est une généralisation de la preuve pour les cas de la couverture des arêtes, tout en utilisant certaines techniques qui rallongent légèrement les chemins, comme pour la borne factorielle de [55].

Comme dans le cas des graphes arête-couvrables, pour prouver cette borne dans le lemme 6.9, nous transformons un ensemble de plus courts chemins bien colorés d'un sommet a vers tous les sommets à distance D de a en un autre ensemble de chemins bien colorés qui peuvent être associés aux feuilles d'un arbre de branchement k -étiqueté. La première différence est les chemins obtenus ne sont plus nécessairement des plus courts chemins. En particulier, à chaque étape de l'algorithme (et il y en aura au plus k), la taille des chemins peut augmenter d'au plus 2 (voir la figure 6.11). La conséquence de cela est que l'ensemble de chemins associés à une feuille de l'arbre peut contenir plus d'un élément. Heureusement, il est possible de borner linéairement en k la taille des ensembles de chemins associés aux feuilles. La borne obtenue dépend donc du nombre de feuilles de l'arbre de branchement k -étiqueté et de la taille des ensembles associés à ses feuilles, soit $O(k \cdot 3^k)$.

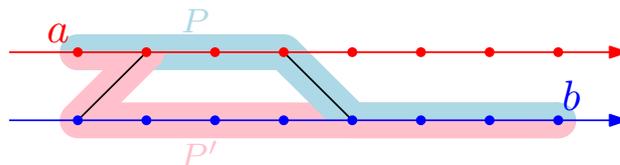


Figure 6.11: Deux $a-b$ chemins bien colorés (P, col) et (P', col') avec le même mot couleurs-sens $\omega = ((\text{rouge}, +), (\text{bleu}, +))$, mais tels que $|P'| = |P| + 2$.

Bonne coloration Pour chaque sommet v de G , notons $\text{couleurs}(v)$ l'ensemble des indices (couleurs) $c \in \{1, \dots, k\}$ tel que v est un sommet de μ_c . Soit P un a - b chemin de G , une *coloration* de P est une fonction $\text{col} : V(P) \rightarrow \{1, \dots, k\}$ attribuant à chaque sommet v de P l'une de ses couleurs $\text{col}(v) \in \text{couleurs}(v)$. Un chemin coloré est une paire (P, col) . La coloration col de P est dit *bonne* si, pour toute couleur c , le sous-graphe induit par l'ensemble des *sommets* utilisant cette couleur c forme un sous-chemin connecté $P[x, y]$ de P (ce qui implique que $P[x, y] = \mu_c[x, y]$). Un chemin coloré (P, col) où col est une bonne coloration est dit *bien coloré*.

Les opérateurs \oplus et \odot s'étendent naturellement aux chemins colorés (sur les sommets), avec la précaution que $(\nu, \text{col}) \odot (\eta, \text{col}')$ n'est défini que lorsque leur sommet commun x , le dernier de ν et le premier de η , satisfait $\text{col}(x) = \text{col}'(x)$. Étant donné un chemin coloré (P, col) , nous désignons à nouveau par $(P[x, y], \text{col})$ sa restriction à un sous-chemin $P[x, y]$ de P . Pour chaque couleur $1 \leq c \leq k$, nous désignons par $(P, \text{monochr}_c)$ la coloration monochromatique de $V(P)$ avec la couleur c . Avec ces notations, tout a - b chemin bien coloré (P, col) avec les couleurs (c_1, \dots, c_l) est de la forme :

$$(\mu_{c_1}[a_1, b_1], \text{monochr}_{c_1}) \oplus (\mu_{c_2}[a_2, b_2], \text{monochr}_{c_2}) \oplus \dots \oplus (\mu_{c_l}[a_l, b_l], \text{monochr}_{c_l})$$

pour des sommets $a = a_1, b_1, a_2, b_2, \dots, a_l, b_l = b$.

Lemme 6.8. *Pour toute paire de sommets a et b de G , il existe un a - b chemin bien coloré (P, col) tel que P est un plus court a - b chemin.*

Preuve. Parmi tous les plus courts a - b chemins, en choisir un qui admet une coloration avec un nombre minimum de sous-chemins monochromatiques. Soit P ce chemin, et col la coloration correspondante. Supposons par contradiction que la coloration n'est pas bonne. Il existe alors trois sommets x, y et z , apparaissant dans cet ordre dans P , tels que $\text{col}(x) = \text{col}(z) \neq \text{col}(y)$. En particulier x et z sont sur le même chemin de base μ_c . Soit P' le chemin obtenu à partir de P en remplaçant $P[x, z]$ par $\mu_c[x, z]$. Premièrement P' n'est pas plus long que P , puisque $\mu_c[x, z]$ est le plus court x - z chemin du graphe G . Deuxièmement, dans P' nous pouvons colorer tous les sommets de $P'[x, z]$ avec la couleur c , et garder toutes les autres couleurs inchangées. Ainsi, P' a strictement moins de sous-chemins monochromatiques que P , une contradiction. \square

Mot couleurs-sens Soit (P, col) un a - b chemin bien coloré; nous rappelons que nous le considérons comme dirigé de a vers b . Comme dans la [section 6.2](#), chaque sous-chemin monochromatique P' de P , de couleur c , induit un sens (+ ou -) dépendant de sa direction par rapport à μ_c , si P' contient au moins deux sommets. Si P' contient seulement un sommet, nous lui attribuons le sens +. Par conséquent, nous pouvons à nouveau définir le *mot couleurs-sens* $\text{CouleurSens}(P, \text{col}) = ((c_1, s_1), (c_2, s_2), \dots, (c_l, s_l))$ sur l'alphabet $\{1, \dots, k\} \times \{+, -\}$, correspondant aux couleurs et aux sens des sous-chemins monochromatiques de P selon l'ordre dans lequel ces sous-chemins apparaissent de a à b .

Nous allons maintenant montrer le [lemme 6.9](#) bornant le nombre de sommets à une distance donnée d'un sommet. À nouveau, par un léger abus de notation, pour un chemin coloré (P, col) nous définissons $\text{couleurs}(P, \text{col}) = \{\text{col}(x) \mid x \in V(P)\}$, l'ensemble des couleurs qui apparaissent sur (P, col) . Pour un ensemble de chemins colorés \mathcal{P} , $\text{couleurs}(\mathcal{P}) = \bigcup_{(P, \text{col}) \in \mathcal{P}} \text{couleurs}(P, \text{col})$. Pour un a - b chemin (P, col) et un sommet x de P , nous définissons $P[x :] = P[x, b]$ et pour un ensemble de chemins colorés \mathcal{P} qui partagent le même sommet x nous définissons $\mathcal{P}[x :] = \bigcup_{(P, \text{col}) \in \mathcal{P}} (P[x :], \text{col})$.

Lemme 6.9. *Pour tout sommet a de $V(G)$ et tout entier D , il y a $O(k \cdot 3^k)$ sommets à distance exactement D de a .*

Preuve. Soit $\{b_1, \dots, b_\ell\}$ l'ensemble des sommets à distance D de a dans G et soit \mathcal{P} un ensemble $\{(P_1, \text{col}_1), \dots, (P_\ell, \text{col}_\ell)\}$ de plus courts a - b_i chemins bien colorés. Notre objectif est de construire un arbre de branchement k -étiqueté (T, lab) tel que :

- chaque noeud n de T est associé à un ensemble de chemins \mathcal{P}_n ,

- pour chaque sommet b_i , il existe une feuille l de T telle que \mathcal{P}_l contient un a - b_i chemin bien coloré,
- l'ensemble \mathcal{P}_l correspondant à une feuille l contient au plus $2k + 1$ chemins.

Notons que la troisième condition diffère du cas arête-couvrables où chaque feuille de l'arbre est associée à exactement un chemin.

Nous allons construire (T, lab) récursivement en commençant par la racine r de T . Initialement, $\mathcal{P}_r = \mathcal{P}$. Pour chaque noeud n de T , \mathcal{P}_n sera un ensemble de chemins bien colorés allant de a à un sous-ensemble de sommets de $\{b_1 \dots, b_\ell\}$. Ces chemins ne seront pas nécessairement des plus courts chemins, mais ils seront au maximum de longueur $D + 2d$, où d est la profondeur du noeud n dans T (la profondeur de la racine étant 0). Soit n un noeud déjà calculé de T . S'il existe un chemin dans \mathcal{P}_n tel que tous les autres chemins de \mathcal{P}_n sont des sous-chemins colorés de ce chemin, ne rien faire : n est une feuille de l'arbre et aucun appel récursif ne sera effectué à partir de n . S'il n'y a pas de tel chemin, nous développons l'arbre à partir de n . Nous construisons maintenant les ensembles de chemins qui seront associés aux enfants de n .

Nous considérons d'abord le cas où n est la racine r de T . Notons qu'il peut exister (P, col) , $(P', \text{col}') \in \mathcal{P}_n$ tel que $\text{col}(a) \neq \text{col}'(a)$.

1. Soit $\mathcal{C}_n = \{\text{col}(a) \mid (P, \text{col}) \in \mathcal{P}_n\}$.
2. Pour chaque $(P, \text{col}) \in \mathcal{P}_n$, soit w_P le dernier sommet de P tel que $c_P = \text{col}(w_P) \in \mathcal{C}_n$. Construire le chemin :

$$(P^*, \text{col}^*) = (\mu_{c_P}[a, w_P], \text{monochr}_{c_P}) \odot (P[w_P :], \text{col})$$

3. Pour chaque $(c, s) \in \mathcal{CS}_n = (\mathcal{C}_n \times \{+, -\})$, soit s_P est le sens de $\mu_{c_P}[a, w_P]$. Définir l'ensemble :

$$\mathcal{P}_{(c,s)} = \{(P^*, \text{col}^*) \mid (P, \text{col}) \in \mathcal{P}_n, (c_P, s_P) = (c, s)\}$$

Si $n \neq r$, prendre (P_0, col_0) le chemin le plus long de \mathcal{P}_n et soit x_n le dernier sommet de P_0 tel que pour tout chemin $(P, \text{col}) \in \mathcal{P}_n$, soit $(P_0[a, x_n], \text{col}_0) = (P[a, x_n], \text{col})$ soit (P, col) est un sous-chemin coloré de $(P_0[a, x_n], \text{col}_0)$. Ce sommet est garanti d'exister, en effet, ce sommet peut être a . Nous procédons comme suit :

4. Soit \mathcal{P}_n° l'ensemble de chemins \mathcal{P}_n qui sont des sous-chemins colorés de $(P_0[a, x_n], \text{col}_0)$ et $\hat{\mathcal{P}}_n = \mathcal{P}_n \setminus \mathcal{P}_n^\circ$.
5. Soit $\mathcal{C}_n = \{\text{col}(z_P) \mid (P, \text{col}) \in \hat{\mathcal{P}}_n\}$ où z_P désigne le sommet apparaissant après x_n sur le chemin P .
6. Pour chaque $(P, \text{col}) \in \hat{\mathcal{P}}_n$, soit w_P le dernier sommet de P tel que $c_P = \text{col}(w_P) \in \mathcal{C}_n$. Soit y_P le second sommet de $\mu_{c_P}[x_n, w_P]$ si $x_n \in V(\mu_{c_P})$; sinon y_P est défini comme le premier sommet de μ_{c_P} adjacent à x_n . Construire le chemin :

$$(P^*, \text{col}^*) = (P[a, x_n], \text{col}) \oplus (\mu_{c_P}[y_P, w_P], \text{monochr}_{c_P}) \odot (P[w_P :], \text{col})$$

7. Pour chaque $(c, s) \in \mathcal{CS}_n = (\mathcal{C}_n \times \{+, -\})$ définir l'ensemble :

$$\mathcal{P}_{(c,s)} = \{(P^*, \text{col}^*) \mid (P, \text{col}) \in \hat{\mathcal{P}}_n, (c_P, s_P) = (c, s)\}$$

Ici s_P est le sens de $\mu_{c_P}[y_P, w_P]$, excepté quand $y_P = w_P$ et $\text{col}^*(x_n) = \text{col}^*(y_P)$, où dans ce cas s_P est le sens de $\mu_{c_P}[x_n, y_P]$.

8. Ajouter \mathcal{P}_n° arbitrairement à un ensemble $\mathcal{P}_{(c,s)}$ non-vide.

Cette construction est illustrée dans la [figure 6.9](#). Ensuite, pour chaque ensemble non vide $\mathcal{P}_{(c,s)}$, avec $(c, s) \in \mathcal{CS}_n$, créer un enfant n' de n associé à $\mathcal{P}_{(c,s)}$, où l'arête $\{n, n'\}$ est étiquetée (c, s) . Enfin, le processus est appliqué récursivement à chacun des noeuds créés.

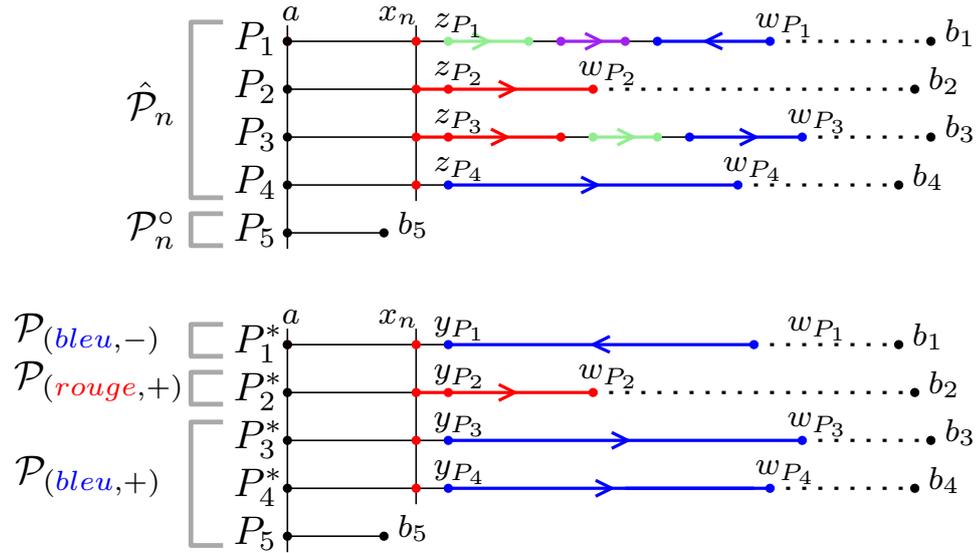


Figure 6.12: Exemple de la construction dans la preuve du [lemme 6.9](#) pour un ensemble de chemins. Dans cet exemple, tous les chemins partagent le même sous-chemin $(P_1[a, x_n], \text{col})$, à l'exception de P_5 qui est un sous-chemin de ce dernier. De plus, les autres chemins peuvent avoir des tailles différentes. Nous avons $\mathcal{C}_n = \{\text{rouge}, \text{bleu}, \text{vert}\}$ et il n'y a pas de sommets avec des couleurs dans \mathcal{C}_n dans les chemins $(P_i[w_{P_i}, :], \text{col}_i)$ à l'exception des sommets w_{P_i} . Les ensembles $\mathcal{P}(\text{rouge}, -)$, $\mathcal{P}(\text{vert}, +)$, $\mathcal{P}(\text{vert}, -)$ sont vides. Les sommets y_{P_1} and $y_{P_3} = y_{P_4}$ sont les mêmes si $x_n \notin V(\mu_{\text{bleu}})$ et peuvent être distincts sinon. Le chemin P_2^* a la même longueur que P_1 , les autres chemins qui ont été modifiés ont pu voir leur longueur modifiée, même P_4^* puisque z_{P_4} peut être différent de y_{P_4} . Le chemin P_5 a été arbitrairement ajouté à $\mathcal{P}(\text{bleu}, +)$.

Affirmation 6.4. Pour tout noeud n de profondeur d de T , \mathcal{P}_n est un ensemble de chemins bien colorés de longueur au plus $D + 2d$.

Preuve. La propriété est trivialement vérifiée pour \mathcal{P}_r où r est la racine de T . Nous montrons que pour un noeud n de T , si $(P, \text{col}) \in \mathcal{P}_n$ est un a - b chemin bien coloré, alors le chemin (P^*, col^*) construit dans le [point 2](#) ou le [point 6](#) est un a - b chemin bien coloré tel que $|P^*| \leq |P| + 2$.

Dans les deux cas, il est clair d'après la construction que P^* est un a - b chemin. Dans le cas du [point 2](#) où $n = r$, puisque $(P[w_P :], \text{col})$ est bien coloré et $\text{col}(w_P) = c_P$, par conséquent (P^*, col^*) est bien coloré. De plus, $(\mu_{c_P}[a, w_P], \text{monochr}_{c_P})$ est un plus court a - w_P chemin, donc $|P^*| = |P|$. Dans le cas du [point 6](#), puisque (P, col) est bien coloré, $(P[a, x_n], \text{col})$ et $(P[w_P :], \text{col})$ le sont aussi. Le sommet w_P est coloré c_P et si $c_P \in \text{couleurs}(P[a, x_n])$ ce doit être la dernière couleur à apparaître, sinon (P, col) ne serait pas bien coloré. Par conséquent le chemin (P^*, col^*) est bien coloré.

Nous montrons maintenant que $|P^*| \leq |P| + 2$. Ceci est vérifié si $x_n \in V(\mu_{c_P})$ puisque y_P est le second sommet du plus court chemin x_n - w_P $\mu_{c_P}[x_n, w_P]$ et donc :

$$P^* = P[a, x_n] \odot \mu_{c_P}[x_n, w_P] \odot P[w_P :]$$

et $|P^*| \leq |P|$. Sinon, dans le cas où $x_n \notin V(\mu_{c_P})$, y_P est le premier sommet de μ_{c_P} adjacent à x_n . Le chemin $\mu_{c_P}[y_P, w_P]$ est le plus court y_P - w_P chemin, donc :

$$|\mu_{c_P}[y_P, w_P]| \leq |(y_P) \oplus P[x_n, w_P]| \leq 1 + |P[x_n, w_P]|$$

Cela implique ce qui suit :

$$\begin{aligned} |P^*| &= |P[a, x_n] \oplus \mu_{c_P}[y_P, w_P] \odot P[w_P :]| \\ &= |P[a, x_n]| + 1 + |\mu_{c_P}[y_P, w_P]| + |P[w_P :]| \\ &\leq |P[a, x_n]| + 2 + |P[x_n, w_P]| + |P[w_P :]| = |P| + 2 \end{aligned}$$

Ainsi, pour chaque $(c, s) \in \mathcal{CS}_n$, l'ensemble $\mathcal{P}_{(c,s)}$ associé à un enfant n' de n est un ensemble de chemins bien colorés de longueur maximale $D' + 2$, où D' est la longueur maximale des chemins dans \mathcal{P}_n . Par induction, pour tout noeud n de profondeur d de T , \mathcal{P}_n est un ensemble de chemins bien colorés de longueur au plus $D + 2d$. \diamond

Soit $n \neq r$ un noeud de T . Nous pouvons observer que pour $(c, s) \in \mathcal{CS}_n$, tous les chemins de $\mathcal{P}_{(c,s)}$ partagent le même successeur de x_n , le sommet y_P défini dans le point 6. Ainsi, pour chaque noeud $n \neq r$ de T avec un parent $p \neq r$, nous pouvons définir un sommet y_n , le successeur de x_p dans les chemins de $\hat{\mathcal{P}}_n$, voir la figure 6.13. Si $n = r$ ou $p = r$, définissons $y_n = a$.



Figure 6.13: Structure d'un chemin $(P, \text{col}) \in \hat{\mathcal{P}}_n$ avec p le parent de n . Il est possible que $y_n = x_n$.

Ensuite, soit $n \neq r$ un noeud de T et n' l'un de ses enfants. Pour $(P', \text{col}') \in \hat{\mathcal{P}}_{n'}$ il y a $(P, \text{col}) \in \hat{\mathcal{P}}_n$ tel que (P', col') correspond au chemin coloré (P^*, col^*) construit à partir de (P, col) dans le noeud n (point 6). Par conséquent, $(P[a, x_n], \text{col}) = (P'[a, x_n], \text{col}')$. De plus, le sommet $y_{n'}$ est le successeur de x_n partagé entre tous les chemins de $\hat{\mathcal{P}}_{n'}$. Donc, si n' n'est pas une feuille, le sommet $x_{n'}$ apparaît strictement après x_n sur les chemins de $\hat{\mathcal{P}}_{n'}$ puisque $x_{n'}$ est $y_{n'}$ ou bien vient après lui. Ainsi, l'observation suivante est vérifiée par induction.

Observation 6.3. Soit $n \neq r$ un noeud de T , $n' \neq n$ un noeud de T_n , $(P, \text{col}) \in \hat{\mathcal{P}}_n$ et $(P', \text{col}') \in \hat{\mathcal{P}}_{n'}$. Alors $(P[a, x_n], \text{col}) = (P'[a, x_n], \text{col}')$ et si n' n'est pas une feuille, x_n apparaît strictement avant $x_{n'}$ sur le chemin P' .

L'observation suivante est une conséquence directe de notre construction.

Observation 6.4. Soit $\{n, n'\}$ une arête de T et $(P, \text{col}) \in \hat{\mathcal{P}}_{n'}$. Alors $\text{lab}(\{n, n'\})$ est une lettre de $\text{CouleurSens}(P[a, x_{n'}], \text{col})$. De plus, la couleur de $\text{lab}(\{n, n'\})$ est la couleur de la première lettre de $\text{CouleurSens}(P[y_{n'} :], \text{col})$ (les sens peuvent être différents, voir l'exception dans la définition de s_P dans le point 7).

L'affirmation suivante implique que la construction termine.

Affirmation 6.5. La hauteur de T est au plus k .

Preuve. Soit $n \neq r$ un noeud de T et $(P, \text{col}) \in \hat{\mathcal{P}}_n$. Nous montrons que la propriété suivante est vérifiée :

$$\text{couleurs}(P^*[y_P :]) \subsetneq \text{couleurs}(P[y_n :])$$

D'après le choix de w_P dans le point 6 nous avons :

$$\text{couleurs}(P^*[y_P :]) \subseteq \text{couleurs}(P[y_n :]) \setminus (\mathcal{C}_n \setminus \{c_P\})$$

Si $|\mathcal{C}_n| \geq 2$, alors $\text{couleurs}(P^*[y_P :]) \subsetneq \text{couleurs}(P[y_n :])$. Si $|\mathcal{C}_n| = 1$, alors $c = \text{col}(y_n) \notin \mathcal{C}_n$, en effet si c était la seule couleur dans \mathcal{C}_n , puisque les chemins dans $\hat{\mathcal{P}}_n$ sont bien colorés et que $y_n \neq y_P$, ils partagent tous le sous-chemin $(\mu_c[y_n, y_P], \text{monochr}_c)$, une contradiction avec le choix de x_n . Par conséquent :

$$\text{couleurs}(P^*[y_P :]) \subseteq \text{couleurs}(P[y_n :]) \setminus \{c\}$$

Ainsi, pour tout enfant n' de n , $\text{couleurs}(\hat{\mathcal{P}}_{n'}[y_{n'} :]) \subsetneq \text{couleurs}(\hat{\mathcal{P}}_n[y_n :])$. Nous avons donc pour un noeud $n \neq r$ de T , $|\text{couleurs}(\hat{\mathcal{P}}_n[y_n :])| \geq 1$, $|\text{couleurs}(\hat{\mathcal{P}}_{n'}[y_{n'} :])| < |\text{couleurs}(\hat{\mathcal{P}}_n[y_n :])|$ pour un enfant n' de n et $|\text{couleurs}(\mathcal{P}_r)| \leq k$. Par conséquent, la hauteur de T est au plus k . \diamond

Affirmation 6.6. (T, lab) est un arbre de branchement k -étiqueté.

Preuve. Soit n un noeud de T et n_1, \dots, n_t ses enfants. La condition (ii) de la [définition 6.1](#) est vérifiée, en effet pour chaque $(c, s) \in \mathcal{CS}_n$, au plus une arête $\{n, n_i\}$ est étiquetée (c, s) . Nous allons premièrement prouver la condition (iii), soit que T_{n_i} ne contient aucune arête avec une étiquette dans $\mathcal{LS}(n) \setminus \{\mathbf{lab}(\{n, n_i\})\}$. Rappelons que $\mathcal{LS}(n) = \{(c, s), (c, \bar{s}) \mid (c, s) \in \mathcal{L}(n)\}$, où $\mathcal{L}(n) = \{\mathbf{lab}(\{n, n_1\}), \dots, \mathbf{lab}(\{n, n_t\})\}$. Nous pouvons observer que $\mathcal{LS}(n) \subseteq \mathcal{CS}_n$ (il peut y avoir des ensembles $\mathcal{P}_{(c,s)}$ vides), et que \mathcal{CS}_n est défini à partir de l'ensemble de couleurs \mathcal{C}_n qui apparaissent sur le premier sommet après x_n dans les chemins de $\hat{\mathcal{P}}_n$.

Soit n_i un enfant de n et soit $(c_i, s_i) = \mathbf{lab}(\{n, n_i\})$. Le cas où n_i est une feuille est trivial, puisque T_{n_i} n'a pas d'arêtes. Supposons donc que n_i n'est pas une feuille. Observons que pour tout $(P, \mathbf{col}) \in \mathcal{P}_n$, par le choix de w_P , $\mathbf{couleurs}(P^*[y_P:], \mathbf{col}^*) \subseteq (\mathbf{couleurs}(P[y_n:], \mathbf{col}) \setminus (\mathcal{C}_n \setminus \{c_P\}))$ (remplacer y_P par a si n est la racine). Ainsi :

$$\mathbf{couleurs}(\hat{\mathcal{P}}_{n_i}[y_{n_i}:]) \subseteq (\mathbf{couleurs}(\hat{\mathcal{P}}_n[y_n:]) \setminus (\mathcal{C}_n \setminus \{c_i\}))$$

Par induction, cela implique que pour tout noeud n' de T_{n_i} , nous avons :

$$\mathbf{couleurs}(\hat{\mathcal{P}}_{n'}[y_{n'}:]) \subseteq (\mathbf{couleurs}(\hat{\mathcal{P}}_n[y_n:]) \setminus (\mathcal{C}_n \setminus \{c_i\}))$$

Par l'[observation 6.4](#), l'étiquette d'une arête $\{n', n''\}$ de T_{n_i} contient la couleur de $y_{n''}$ dans les chemins de $\hat{\mathcal{P}}_{n''}$, ainsi aucune arête de T_{n_i} n'a une étiquette dans $(\mathcal{C}_n \setminus \{c_i\}) \times \{+, -\}$.

Nous prouvons maintenant que (c_i, \bar{s}_i) n'est pas utilisée en tant qu'étiquette pour les arêtes de T_{n_i} . Soit $\{n', n''\}$ une arête de T_{n_i} et supposons que $\mathbf{lab}(\{n', n''\}) = (c_i, \bar{s}_i)$. Soit $(P, \mathbf{col}) \in \hat{\mathcal{P}}_{n_i}$ et $(P', \mathbf{col}') \in \hat{\mathcal{P}}_{n''}$. Par l'[observation 6.3](#), $(P[a, x_{n_i}], \mathbf{col}) = (P'[a, x_{n_i}], \mathbf{col}')$, donc par l'[observation 6.4](#) $\mathbf{lab}(\{n, n_i\}) = (c_i, s_i)$ est une lettre de $\mathbf{CouleurSens}(P'[a, x_{n_i}], \mathbf{col}')$ et (c_i, \bar{s}_i) est une lettre de $\mathbf{CouleurSens}(P', \mathbf{col}')$. Ceci est une contradiction puisque le chemin (P', \mathbf{col}') est bien coloré ([affirmation 6.4](#)). Il n'y a donc pas d'arêtes de T_{n_i} avec des étiquettes dans $\mathcal{CS}_n \setminus \{(c_i, s_i)\}$, ce qui prouve la condition (iii).

Il reste à montrer la condition (i), soit que dans chaque chemin de la racine de T à une de ses feuilles, les arêtes qui partagent la même étiquette forment un sous-chemin connexe. Soit $r = p_1, p_2, \dots, p_t = n$ le chemin de la racine r à un noeud non feuille n de T . S'il existe un enfant n' de n tel que $\mathbf{lab}(\{n, n'\}) = \mathbf{lab}(\{p_{i-1}, p_i\}) = (c, s)$, $i \in \{2, \dots, t\}$, alors nous devons montrer que pour tout $i < j \leq t$, $\mathbf{lab}(\{p_{j-1}, p_j\}) = (c, s)$. Supposons qu'il existe de tels n' et i . Pour $(P, \mathbf{col}) \in \hat{\mathcal{P}}_{p_i}$ et $(P', \mathbf{col}') \in \hat{\mathcal{P}}_{n'}$, par l'[observation 6.3](#) $(P[a, x_{p_i}], \mathbf{col}) = (P'[a, x_{p_i}], \mathbf{col}')$. Par l'[observation 6.4](#), le sommet y_{p_i} (qui apparaît avant x_{p_i}) est coloré c dans ces deux sous-chemins, de même que le sommet $y_{n'}$ dans (P', \mathbf{col}') . Puisque (P', \mathbf{col}') est bien coloré, le sous-chemin $(P'[y_{p_i}, y_{n'}], \mathbf{col}')$ est entièrement coloré par c . Pour tout $i < j \leq t$, $(P''[a, x_{p_j}], \mathbf{col}'') = (P'[a, x_{p_j}], \mathbf{col}')$, avec $(P'', \mathbf{col}'') \in \hat{\mathcal{P}}_{p_j}$. Cela implique que la première couleur de $(P''[y_{p_j}:], \mathbf{col}'')$ est c . Comme il n'y a pas d'arête de T_{n_i} étiquetée (c, \bar{s}) , cela implique que $\mathbf{lab}(\{p_{j-1}, p_j\}) = (c, s)$ et ainsi la condition (i) est vérifiée. Par conséquent, (T, \mathbf{col}) est un arbre de branchement k -étiqueté, ce qui conclut la preuve de l'[affirmation 6.6](#). \diamond

Pour un noeud non feuille n de T , nous pouvons observer que pour tout a - b_i chemin dans \mathcal{P}_n , $b_i \in \{b_1, \dots, b_\ell\}$, il existe un de ses enfants n' tel que $\mathcal{P}_{n'}$ contient un a - b_i chemin (il a pu être ajouté via l'ensemble \mathcal{P}_n°). Cela implique que pour tout b_i , il existe une feuille l de T telle que \mathcal{P}_l contient un plus court a - b_i chemin bien coloré. Pour une feuille l de T , par construction, \mathcal{P}_l contient un chemin (P, \mathbf{col}) incluant tout autre chemin de \mathcal{P}_l . De plus, la longueur de P est au plus $D + 2k$ par les [affirmations 6.4](#) et [6.5](#). Cela implique que $|\mathcal{P}_l| \leq 2k + 1$, car \mathcal{P}_l ne contient que des sous-chemins de P , de longueur comprise entre D et $D + 2k$. Puisque T est un arbre de branchement k -étiqueté, par le [lemme 6.6](#), il a $O(3^k)$ feuilles. Cela implique qu'il y a $O(k \cdot 3^k)$ sommets à distance D de a . \square

Pour finir la preuve du [théorème 6.2](#) et montrer que $\text{pw}(G) = O(k \cdot 3^k)$, il suffit d'appliquer le [lemme 6.2](#) avec $K = O(k \cdot 3^k)$.

6.4 Conséquences algorithmiques

Preuves du théorème 6.3 et du corollaire 6.4 Rappelons que, par souci de simplicité, nous supposons que notre graphe d'entrée est connexe, mais tous les résultats s'étendent facilement aux graphes non connexes. Nous montrons tout d'abord la première partie du théorème 6.3, à savoir que le problème ISOMETRIC PATH COVER WITH TERMINALS est FPT lorsqu'il est paramétré par k , le nombre de paires de terminaux. Une première conséquence est que le problème STRONG GEODETIC SET WITH TERMINALS, un cas particulier de ISOMETRIC PATH COVER WITH TERMINALS avec $\binom{k}{2}$ paires de terminaux, l'est aussi, ce qui montre la deuxième partie du théorème 6.3.

Le corollaire 6.4 affirme que ces mêmes problèmes dans leur version sans terminaux sont dans la classe de complexité XP. Il est directement impliqué par le théorème 6.3 puisque pour ISOMETRIC PATH COVER et STRONG GEODETIC SET il suffit d'essayer tous les ensembles de terminaux possibles et d'utiliser les algorithmes FPT pour les versions avec terminaux. En particulier, pour ISOMETRIC PATH COVER, l'énumération de toutes les paires de k terminaux possibles se fait en temps $n^{O(k)}$ et il en est de même pour STRONG GEODETIC SET, d'où la complexité totale XP pour chacun des problèmes.

Nous nous concentrons sur ISOMETRIC PATH COVER WITH TERMINALS, avec le paramètre k , l'entrée G et les k paires de terminaux $(s_1, t_1), \dots, (s_k, t_k)$. Nous pouvons supposer que la largeur linéaire (et donc la largeur arborescente) du graphe d'entrée est bornée supérieurement par une fonction de k par le théorème 6.2, et qu'en entrée est fournie une décomposition arborescente d'une telle largeur. En effet, une telle décomposition peut être obtenue en prenant une décomposition linéaire construite à partir d'un parcours en largeur comme vu dans le lemme 6.2 (nous rappelons qu'une décomposition linéaire est aussi une décomposition arborescente). Si la décomposition obtenue via le parcours est de trop grande largeur, par le théorème 6.2, nous pouvons directement conclure que notre graphe d'entrée est une instance négative.

Le problème ISOMETRIC PATH COVER WITH TERMINALS peut être réduit à un problème EMSO₂ linéaire d'optimisation. Soit $\varphi(E_1, E_2, \dots, E_k)$ la formule sur les ensembles d'arêtes E_1, \dots, E_k qui exprime la propriété qu'il existe k sous-graphes connexes $(V_1, E_1), \dots, (V_k, E_k)$ de G tels que les ensembles V_1, \dots, V_k couvrent tous les sommets de G , et qu'un graphe (V_i, E_i) contient les terminaux s_i et t_i , pour tout $1 \leq i \leq k$. Formellement :

$$\begin{aligned} \varphi(E_1, E_2, \dots, E_k) &= \exists V_1, V_2, \dots, V_k [(s_1 \in V_1) \wedge (t_1 \in V_1) \wedge \dots \wedge (s_k \in V_k) \wedge (t_k \in V_k) \\ &\quad \wedge \text{Couverture}(V_1, \dots, V_k) \wedge \text{Connexe}(V_1, E_1) \wedge \dots \wedge \text{Connexe}(V_k, E_k)] \end{aligned}$$

Rappelons que les prédicats MSO₂ Connexe et Couverture ont déjà été décrits dans le chapitre Préliminaires, section 2.2.3.

Considérons la version d'optimisation de ce problème, où le but est de trouver des ensembles d'arêtes E_1, \dots, E_k satisfaisant $\varphi(E_1, \dots, E_k)$ et minimisant $|E_1| + \dots + |E_k|$. Soit CouvertureOpt cet optimum. D'après le théorème 2.4, ce problème peut être résolu en temps linéaire sur des graphes de largeur arborescente bornée. En d'autres termes, le problème est FPT paramétré par la largeur arborescente plus le nombre de terminaux, et le temps d'exécution est linéaire en n .

Nous observons maintenant que notre entrée est une instance positive de ISOMETRIC PATH COVER WITH TERMINALS si et seulement si :

$$\text{CouvertureOpt} = \text{dist}(s_1, t_1) + \text{dist}(s_2, t_2) + \dots + \text{dist}(s_k, t_k)$$

En effet, s'il existe les plus courts s_i - t_i chemins P_1, \dots, P_k requis couvrant l'ensemble des sommets du graphe, alors leurs ensembles d'arêtes E_1, \dots, E_k fournissent une solution pour le problème d'optimisation dont l'objectif est égal à la somme des longueurs des chemins. Inversement, pour chacun des k sous-graphes connexes (V_i, E_i) de G tel que $s_i, t_i \in V_i$, nous avons $|E_i| \geq \text{dist}(s_i, t_i)$. Donc en vérifiant si CouvertureOpt correspond à la somme des distances entre les paires de terminaux, nous décidons si l'entrée satisfait ISOMETRIC PATH COVER WITH TERMINALS. Ainsi, ce problème est FPT paramétré par k , ce qui conclut la preuve de théorème 6.3.

Comme mentionné précédemment, la même approche s'étend (en modifiant la formule MSO₂ φ) aux variantes où les plus courts chemins doivent couvrir les arêtes ou bien partitionnent les arêtes ou les sommets. Pour la couverture des arêtes il suffit dans φ de remplacer Couverture(V_1, \dots, V_k) par une version portant sur les arêtes. Pour les versions de partition il suffit d'ajouter dans φ que les k ensembles qui couvrent le graphe soient disjoints sur les sommets ou les arêtes.

6.5 Conclusion et perspectives

Nous avons montré que les graphes arête-couvrables ou sommet-couvrables par k plus courts chemins ont leur largeur linéaire bornée supérieurement par une fonction exponentielle en k . Naturellement, la première question ouverte est donc de savoir si elle peut être améliorée en une borne polynomiale. Une telle amélioration ne peut pas reposer sur une décomposition linéaire basée sur les niveaux d'un parcours en largeur arbitraire. En effet, il existe des exemples (voir la figure 6.14) où un même niveau contient 2^k sommets.

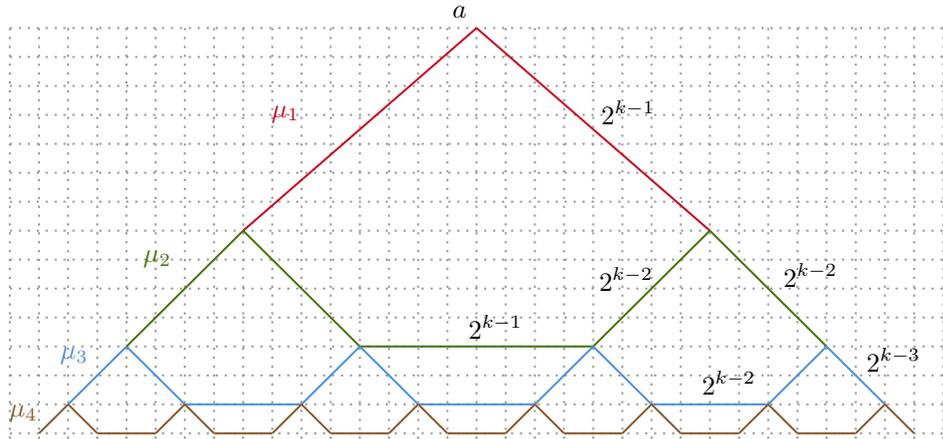


Figure 6.14: Un graphe dont les arêtes peuvent être couvertes par k plus courts chemins et possédant 2^k sommets à une distance $2^k - 1$ de a . Une décomposition linéaire obtenue par un parcours en largeur à partir de a a une largeur exponentielle en k . Néanmoins, il peut facilement être prouvé que ce graphe a une largeur linéaire d'au plus k .

Nous laissons donc ouverte la question d'une borne polynomiale pour la largeur linéaire ou la largeur arborescente.

Problème ouvert. Les graphes arête-couvrables et sommet-couvrables par k plus courts chemins ont-ils :

- une largeur linéaire polynomiale en k ?
- une largeur arborescente polynomiale en k ?

Une approche pour répondre à ces questions pourrait être de montrer que les graphes couvrables par k plus courts chemins n'admettent pas de grilles $k \times k$ en tant que mineur, respectivement une subdivision d'un arbre binaire complet de hauteur k . En effet, les graphes n'admettant pas de grilles $k \times k$ en tant que mineur ont une largeur arborescente polynomiale en k [35]. Les graphes qui en plus ne contiennent pas une subdivision d'un arbre binaire complet de hauteur k ont une largeur linéaire polynomiale en k [74].

Nous pouvons observer que notre résultat ne se généralise pas aux graphes couverts avec un petit nombre de chemins induits, en effet, les grilles ont une largeur arborescente arbitrairement large mais sont arête-couvrables par quatre chemins induits. Pour les graphes orientés sommet-couvrables ou arête-couvrables par k plus courts chemins (orientés) nous pouvons nous demander quelle est la largeur arborescente de leur graphe non-orienté sous-jacent. Dans le cas sommet-couvrable il existe des exemples pour lesquels la largeur arborescente n'est pas bornée, par exemple, prenons un chemin orienté (x_1, \dots, x_n) , et ajoutons tous les arcs (x_i, x_j) avec $1 \leq j < i \leq n$. Le graphe orienté obtenu peut être couvert par le chemin orienté (x_1, \dots, x_n) , cependant le graphe non-orienté sous-jacent est une clique et est donc de largeur arborescente $n - 1$. Un autre exemple permet de montrer que c'est également le cas pour les graphes orientés acycliques. Considérons deux chemins orientés disjoints de longueur n et ajoutons tous les arcs des sommets du premier chemin à tous les sommets du second. Le graphe obtenu peut être couvert par deux plus courts

chemins orientés mais le sous-graphe induit a une largeur arborescente linéaire en n . Les résultats présentés dans ce chapitre ne peuvent donc pas être directement étendus aux graphes orientés, ni même aux graphes orientés acycliques.

Pour la partie algorithmique, nous avons prouvé que les problèmes ISOMETRIC PATH COVER WITH TERMINALS et STRONG GEODETIC SET WITH TERMINALS sont FPT paramétrés par le nombre de terminaux. Cela implique directement que les problèmes ISOMETRIC PATH COVER et STRONG GEODETIC SET sont XP par rapport au même paramètre, en énumérant simplement toutes les paires (resp. les ensembles) de terminaux possibles. La question ouverte naturelle est donc de savoir si ces problèmes sont FPT.

Problème ouvert. ISOMETRIC PATH COVER et STRONG GEODETIC SET sont-ils FPT paramétrés par la taille de la solution ?

D'après le [théorème 6.2](#), cela revient à demander si les problèmes sont FPT lorsqu'ils sont paramétrés par la taille de la solution (c'est-à-dire le nombre de chemins/terminaux) plus la largeur linéaire. En effet, si k est le nombre de terminaux, le [théorème 6.2](#) garantit que soit la largeur linéaire pw du graphe d'entrée est bornée par une fonction $f(k)$, soit nous pouvons directement rejeter l'entrée comme étant une instance négative. Par conséquent, si l'un des problèmes est FPT paramétré par $k+pw$, nous obtenons un algorithme FPT paramétré par k comme suit. L'algorithme vérifie que $pw \leq f(k)$ comme dans le [théorème 6.2](#), par un simple parcours en largeur à partir d'un sommet arbitraire. Si l'inégalité n'est pas vérifiée, l'algorithme rejette. Dans le cas contraire, il suffit d'appliquer l'algorithme paramétré par $k + pw$ sur le paramètre $k + f(k)$. Néanmoins, la réponse à la question de savoir si ces problèmes sont FPT pour le paramètre $k + pw$ ne semble pas triviale. En effet, alors que de nombreux problèmes d'optimisation sont FPT lorsqu'ils sont paramétrés par la largeur arborescente/linéaire, plusieurs problèmes incluant des contraintes sur les distances restent $W[1]$ -difficiles même lorsqu'ils sont paramétrés par de tels paramètres structurels, plus la taille de la solution. Nous pouvons citer des résultats récents de difficulté pour d -SCATTERED SET [86], dont le but est de trouver un grand ensemble de sommets éloignés (distance d'au moins d entre chaque paire de sommets de l'ensemble) ou, encore plus proche de nos problèmes, GEODETIC SET [88], où l'on cherche à trouver un petit ensemble de terminaux du graphe d'entrée tel que l'ensemble de tous les plus courts chemins entre chaque paire de terminaux couvre tout le graphe.

Chapitre 7

Conclusion et perspectives de la thèse

Dans cette thèse nous nous sommes intéressés à des problèmes de modification de graphes et de couverture de graphes dans le cadre de la complexité paramétrée.

Modification d'arêtes Dans la première partie de cette thèse nous avons présenté des noyaux polynomiaux pour plusieurs problèmes de modifications d'arêtes. Plus en détail, nous avons montré que :

- TRIVIALY PERFECT EDITION ainsi que ses versions complétion et délétion admettent un noyau qui contient $O(k^2)$ sommets,
- BLOCK GRAPH EDITION et BLOCK GRAPH DELETION admettent un noyau qui contient $O(k^2)$ sommets,
- STRICTLY CHORDAL EDITION admet un noyau qui contient $O(k^4)$ sommets,
- STRICTLY CHORDAL COMPLETION et STRICTLY CHORDAL DELETION admettent un noyau qui contient $O(k^3)$ sommets.

Pour élaborer ces noyaux, nous avons utilisé une approche que nous appelons « décomposition et sommets affectés ». Elle exploite la structure arborescente des classes de graphes considérées. En particulier, dans chacun de nos noyaux nous avons caractérisé et réduit des « bonnes parties » du graphe initial qui se rattachent en un ou deux points au reste du graphe ou de sa décomposition.

Cette approche s'avère être idéale pour la mise au point de noyaux pour des classes de graphes qui ont une structure arborescente, ce qui est notamment le cas pour des sous-classes des graphes cordaux. Des approches se basant sur des idées similaires ont notamment été utilisées pour d'autres classes de graphes telles que les graphes 3-leaf power [13], les graphes ptolémaïques [42], les graphes d'intervalles propres [14] ou les cographes [43]. Cependant, comme nous avons pu le voir dans la [partie I](#), bien que les idées employées soient similaires, elles nécessitent une compréhension fine de la classe de graphes étudiée et les résultats ne s'adaptent pas de manière automatique d'une classe à l'autre.

Une des motivations à l'étude de l'existence de noyaux polynomiaux pour des problèmes de modification d'arêtes vers des sous-classes de graphes cordaux était la conjecture de Bessy et Perez [14].

Conjecture. Soit \mathcal{G} une classe de graphes caractérisée par une famille d'obstructions contenant les trous plus un ensemble fini de graphes. Alors le problème \mathcal{G} -COMPLETION admet un noyau polynomial paramétré par la taille de la solution.

Montrer une telle conjecture permettrait d'unifier de nombreux résultats de noyaux polynomiaux pour des sous-classes des graphes cordaux. Cela nécessiterait la mise au point d'une approche générique, d'un « méta-noyau », ce qui semble être un défi de taille. Inversement, il serait intéressant d'infirmer cette conjecture en montrant la non-existence d'un noyau polynomial pour une classe de graphes respectant les conditions de la conjecture.

De nombreux résultats sont connus pour les problèmes de modifications d'arêtes, mais beaucoup d'autres restent ouverts. C'est particulièrement le cas pour des problèmes d'édition ou de délétion.

Problème ouvert. Les problèmes suivants admettent-ils un noyau polynomial ?

- CHORDAL EDITION et CHORDAL DELETION,
- INTERVAL COMPLETION et ses versions édition et délétion,
- PROPER INTERVAL EDITION et PROPER INTERVAL DELETION,
- r -LEAF POWER COMPLETION et ses versions édition et délétion pour $r \geq 4$,
- DISTANCE HEREDITARY COMPLETION et ses versions édition et délétion,

Mentionnons que les problèmes 4-LEAF POWER COMPLETION et PROPER INTERVAL EDITION semblent être des problèmes pour lesquels l'approche utilisée dans cette thèse pourrait permettre de montrer l'existence d'un noyau polynomial. L'existence de noyaux polynomiaux pour les problèmes CHORDAL EDITION et INTERVAL COMPLETION reste une question ouverte majeure.

Pour terminer, nous demandons s'il est possible de trouver des noyaux de plus petite taille pour les problèmes que nous avons étudiés dans la [partie I](#). En particulier nous posons le problème ouvert suivant :

Problème ouvert. TRIVIALY PERFECT EDITION et THRESHOLD EDITION ainsi que leurs versions complétion et délétion admettent-ils un noyau avec $O(k^{2-\epsilon})$ sommets pour $\epsilon > 0$?

Couverture par plus courts chemins Dans la seconde partie de cette thèse nous nous sommes intéressés à divers problèmes de couverture de graphes par des plus courts chemins. Notre résultat principal est le suivant :

- Soit G un graphe dont l'ensemble d'arêtes peut être couvert par au plus k plus courts chemins. Alors la largeur linéaire de G est $O(3^k)$.
- Soit G un graphe dont l'ensemble de sommets peut être couvert par au plus k plus courts chemins. Alors la largeur linéaire de G est $O(k \cdot 3^k)$.

En particulier nous avons montré dans le cas de la couverture des arêtes (resp. sommets) que le nombre de sommets à distance D d'un sommet fixé est borné par $O(3^k)$ (resp. $O(k \cdot 3^k)$). Ces résultats nous permettent de montrer, à l'aide d'une version du théorème de Courcelle ([théorème 2.4](#)) étendu aux problèmes d'optimisation, que les problèmes ISOMETRIC PATH COVER WITH TERMINALS et STRONG GEODETIC SET WITH TERMINALS sont FPT. Une conséquence immédiate est que les versions sans terminaux de ces problèmes, à savoir ISOMETRIC PATH COVER et STRONG GEODETIC SET sont dans XP. De plus, nos résultats se généralisent de manière immédiate aux versions de ces problèmes où les chemins partitionnent les sommets ou les arêtes du graphe.

La première question naturelle est de demander s'il est possible d'améliorer la borne sur la largeur linéaire et sur la largeur arborescente.

Problème ouvert. Les graphes arête-couvrables et sommet-couvrables par k plus courts chemins ont-ils :

- une largeur linéaire polynomiale en k ?
- une largeur arborescente polynomiale en k ?

Une seconde question naturelle est de savoir si les problèmes ISOMETRIC PATH COVER et STRONG GEODETIC SET sont FPT quand ils sont paramétrés par le paramètre taille de la solution ou bien s'ils sont $W[1]$ -difficiles.

Problème ouvert. ISOMETRIC PATH COVER et STRONG GEODETIC SET sont-ils FPT paramétrés par la taille de la solution ?

En raison de notre borne sur la largeur linéaire en fonction du nombre de chemins ou de terminaux, cette question revient à demander si les problèmes sont FPT lorsqu'ils sont paramétrés par la taille de la solution (c'est-à-dire le nombre de chemins/terminaux) plus la largeur linéaire.

Bibliographie

- [1] Faisal N. Abu-Khzam. A kernelization algorithm for d-Hitting Set. *Journal of Computer and System Sciences*, 76(7):524–531, 2010.
- [2] Akanksha Agrawal, Sudeshna Kolay, Daniel Lokshantov, and Saket Saurabh. A faster FPT algorithm and a smaller kernel for block graph vertex deletion. In *Proceedings of the 12th Latin American Symposium on Theoretical Informatics, LATIN*, pages 1–13, 2016.
- [3] Akanksha Agrawal, Daniel Lokshantov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Feedback vertex set inspired kernel for chordal vertex deletion. *ACM Transactions on Algorithms*, 15(1):11:1–11:28, 2019.
- [4] Akanksha Agrawal, Daniel Lokshantov, Pranabendu Misra, Saket Saurabh, and Meirav Zehavi. Polynomial kernel for interval vertex deletion. *ACM Transactions on Algorithms*, 19(2):11:1–11:68, 2023.
- [5] M. Aigner and M. Fromme. A game of cops and robbers. *Discrete Applied Mathematics*, 8(1):1–12, 1984.
- [6] Andris Ambainis, Kaspars Balodis, Jānis Iraids, Martins Kokainis, Krišjānis Prūsis, and Jevgēnijs Vihrovs. Quantum speedups for exponential-time dynamic programming algorithms. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1783–1793, 2019.
- [7] David Applegate, Robert Bixby, Vašek Chvátal, and William Cook. TSP cuts which do not conform to the template paradigm. *Computational Combinatorial Optimization: Optimal or Provably Near-Optimal Solutions*, pages 261–303, 2001.
- [8] Stefan Arnborg, Jens Lagergren, and Detlef Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12(2):308–340, 1991.
- [9] Hans-Jürgen Bandelt and Henry Martyn Mulder. Distance-hereditary graphs. *Journal of Combinatorial Theory, Series B*, 41(2):182–208, 1986.
- [10] Gabriel Bathie, Nicolas Bousquet, Yixin Cao, Yuping Ke, and Théo Pierron. (Sub) linear kernels for edge modification problems toward structured graph classes. *Algorithmica*, pages 1–27, 2022.
- [11] Richard Bellman. Dynamic programming treatment of the travelling salesman problem. *Journal of the ACM*, 9(1):61–63, 1962.
- [12] Claude Berge. Path partitions in directed graphs. In *North-Holland Mathematics Studies*, volume 75, pages 59–63. Elsevier, 1983.
- [13] Stéphane Bessy, Christophe Paul, and Anthony Perez. Polynomial kernels for 3-leaf power graph modification problems. *Discrete Applied Mathematics*, 158(16):1732–1744, 2010.
- [14] Stéphane Bessy and Anthony Perez. Polynomial kernels for proper interval completion and related problems. *Information and Computation*, 231:89–108, 2013.
- [15] Ivan Bliznets, Fedor V Fomin, Marcin Pilipczuk, and Michał Pilipczuk. Subexponential parameterized algorithm for interval completion. *ACM Transactions on Algorithms*, 14(3):1–62, 2018.

- [16] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.
- [17] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8):423–434, 2009.
- [18] Hans L. Bodlaender, Fedor V. Fomin, Daniel Lokshtanov, Eelko Penninkx, Saket Saurabh, and Dimitrios M. Thilikos. (Meta) kernelization. *Journal of the ACM (JACM)*, 63(5):1–69, 2016.
- [19] Hans L. Bodlaender, John R. Gilbert, Hjalmtyr Hafsteinsson, and Ton Kloks. Approximating treewidth, pathwidth, frontsize, and shortest elimination tree. *Journal of Algorithms*, 18(2):238–255, 1995.
- [20] Hans L. Bodlaender, Bart MP Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM Journal on Discrete Mathematics*, 28(1):277–305, 2014.
- [21] Ulrik Brandes, Michael Hamann, Luise Häuser, and Dorothea Wagner. Skeleton-based clustering by quasi-threshold editing. In *Algorithms for Big Data: DFG Priority Program 1736*, pages 134–151. Springer, 2023.
- [22] Ulrik Brandes, Michael Hamann, Ben Strasser, and Dorothea Wagner. Fast quasi-threshold editing. In *Proceedings of the 23rd European Symposium on Algorithms, ESA*, pages 251–262, 2015.
- [23] Andreas Brandstädt and Van Bang Le. Structure and linear time recognition of 3-leaf powers. *Information Processing Letters*, 98(4):133–138, 2006.
- [24] Andreas Brandstädt, Van Bang Le, and Jeremy P Spinrad. *Graph classes: a survey*. SIAM, 1999.
- [25] Andreas Brandstädt, Van Bang Le, and R. Sritharan. Structure and linear-time recognition of 4-leaf powers. *ACM Transactions on Algorithms (TALG)*, 5(1):1–22, 2008.
- [26] Pablo Burzyn, Flavia Bonomo, and Guillermo Durán. NP-completeness results for edge modification problems. *Discrete Applied Mathematics*, 154(13):1824–1844, 2006.
- [27] Pablo Burzyn, Flavia Bonomo, and Guillermo Durán. NP-completeness results for edge modification problems. *Discrete Applied Mathematics*, 154(13):1824–1844, 2006.
- [28] Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.
- [29] Leizhen Cai and Yufei Cai. Incompressibility of H -free edge modification problems. *Algorithmica*, 71(3):731–757, 2015.
- [30] Liming Cai, Jianer Chen, Rodney G. Downey, and Michael R. Fellows. Advice classes of parameterized tractability. *Annals of Pure and Applied Logic*, 84(1):119–138, 1997.
- [31] Yixin Cao. Unit interval editing is fixed-parameter tractable. *Information and Computation*, 253:109–126, 2017.
- [32] Yixin Cao and Jianer Chen. Cluster editing: Kernelization based on edge cuts. *Algorithmica*, 64:152–169, 2012.
- [33] Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75:118–137, 2016.
- [34] Dibyayan Chakraborty, Antoine Dailly, Sandip Das, Florent Foucaud, Harmender Gahlawat, and Subir K. Ghosh. Complexity and algorithms for isometric path cover on chordal graphs and beyond. In *Proceedings of the 33rd International Symposium on Algorithms and Computation, ISAAC*, pages 12:1–12:17, 2022.
- [35] Chandra Chekuri and Julia Chuzhoy. Polynomial bounds for the grid-minor theorem. *Journal of the ACM*, 63(5):1–65, 2016.

- [36] Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40-42):3736–3756, 2010.
- [37] Jianer Chen and Jie Meng. A 2k kernel for the cluster editing problem. *Journal of Computer and System Sciences*, 78(1):211–220, 2012.
- [38] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing, STOC*, pages 151–158, 1971.
- [39] Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.
- [40] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. On the fixed parameter complexity of graph enumeration problems definable in monadic second-order logic. *Discrete applied mathematics*, 108(1-2):23–52, 2001.
- [41] Christophe Crespelle, Pål Grønås Drange, Fedor V. Fomin, and Petr A. Golovach. A survey of parameterized algorithms and the complexity of edge modification. *Computer Science Review*, 48:100556, 2023.
- [42] Christophe Crespelle, Benjamin Gras, and Anthony Perez. Completion to chordal distance-hereditary graphs: a quartic vertex-kernel. In *Proceedings of the 47th International Workshop on Graph-Theoretic Concepts in Computer Science, WG*, pages 156–168, 2021.
- [43] Christophe Crespelle, Rémi Pellerin, and Stéphan Thomassé. A quasi-quadratic vertex kernel for cograph edge editing. *preprint*, arXiv:2212.14814, 2022.
- [44] Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michał Pilipczuk, Johan MM Van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. *ACM Transactions on Algorithms*, 18(2):17:1–17:31, 2022.
- [45] Tom Davot, Lucas Isenmann, and Jocelyn Thiebaut. On the approximation hardness of geodesic set and its variants. In *Proceedings of the 27th International Computing and Combinatorics Conference, COCOON*, pages 76–88, 2021.
- [46] João Henrique Gonçalves de Sousa. Exact algorithms and computational complexity for the strong geodesic set problem. Master’s thesis, Universidade Federal de Minas Gerais, Brazil, 2018.
- [47] Holger Dell and Dieter Van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. *Journal of the ACM (JACM)*, 61(4):1–27, 2014.
- [48] Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- [49] Michael Dom, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Error compensation in leaf power problems. *Algorithmica*, 44(4):363–381, 2006.
- [50] Michael Dom, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Closest 4-leaf power is fixed-parameter tractable. *Discrete Applied Mathematics*, 156(18):3345–3361, 2008.
- [51] Rodney G. Downey and Michael R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, 1999.
- [52] Pål Grønås Drange, Markus Fanebust Dregi, Daniel Lokshtanov, and Blair D. Sullivan. On the threshold of intractability. *Journal of Computer and System Sciences*, 124:1–25, 2022.
- [53] Pål Grønås Drange, Fedor V. Fomin, Michał Pilipczuk, and Yngve Villanger. Exploring the subexponential complexity of completion problems. *ACM Transactions on Computation Theory (TOCT)*, 7(4):1–38, 2015.
- [54] Pål Grønås Drange and Michał Pilipczuk. A polynomial kernel for trivially perfect editing. *Algorithmica*, 80(12):3481–3524, 2018.
- [55] Maël Dumas, Florent Foucaud, Anthony Perez, and Ioan Todinca. On graphs coverable by k shortest paths. In *Proceedings of the 33rd International Symposium on Algorithms and Computation, ISAAC*, pages 40:1–40:15, 2022.

- [56] Maël Dumas, Florent Foucaud, Anthony Perez, and Ioan Todinca. On graphs coverable by k shortest paths. *preprint*, arXiv:2206.15088, 2022.
- [57] Maël Dumas and Anthony Perez. An improved kernelization algorithm for trivially perfect editing. In *Proceedings of the 18th International Symposium on Parameterized and Exact Computation, IPEC*, pages 15:1–15:17, 2023.
- [58] Maël Dumas, Anthony Perez, Mathis Rocton, and Ioan Todinca. Polynomial kernels for edge modification problems towards block and strictly chordal graphs. *preprint*, arXiv:2201.13140, 2022.
- [59] Maël Dumas, Anthony Perez, and Ioan Todinca. A cubic vertex-kernel for trivially perfect editing. In *Proceedings of the 46th International Symposium on Mathematical Foundations of Computer Science MFCS*, pages 45:1–45:14, 2021.
- [60] Maël Dumas, Anthony Perez, and Ioan Todinca. Polynomial kernels for strictly chordal edge modification problems. In *Proceedings of the 16th International Symposium on Parameterized and Exact Computation, IPEC*, pages 17:1–17:16, 2021.
- [61] Maël Dumas, Anthony Perez, and Ioan Todinca. A cubic vertex-kernel for trivially perfect editing. *Algorithmica*, 85(4):1091–1110, 2023.
- [62] Jack Edmonds. Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467, 1965.
- [63] Qilong Feng, Qian Zhou, and Jianxin Wang. Kernelization and randomized parameterized algorithms for co-path set problem. *Journal of Combinatorial Optimization*, 32:67–78, 2016.
- [64] Henning Fernau, Florent Foucaud, Kevin Mann, Rajath Rao K. N., and Utkarsh Paradiya. Parameterizing path partitions. In *Proceedings of the 13th International Symposium on Algorithms and Complexity, CIAC*, pages 187–201, 2023.
- [65] David C. Fisher and Shannon L. Fitzpatrick. The isometric number of a graph. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 38(1):97–110, 2001.
- [66] Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2010.
- [67] Fedor V. Fomin, Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Planar F-deletion: Approximation, kernelization and optimal FPT algorithms. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 470–479, 2012.
- [68] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press, 2019.
- [69] Fedor V. Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. *SIAM Journal on Computing*, 42(6):2197–2216, 2013.
- [70] Philippe Galinier, Michel Habib, and Christophe Paul. Chordal graphs and their clique graphs. In Manfred Nagl, editor, *Proceedings of the 21st International Workshop on Graph-Theoretic Concepts in Computer Science, WG*, pages 358–371, 1995.
- [71] Tilbor Gallai and Arthur Norton Milgram. Verallgemeinerung eines graphentheoretischen Satzes von Rédei. *Acta Sc. Math*, 21:181–186, 1960.
- [72] Martin Charles Golumbic and Uri N. Peled. Block duplicate graphs and a hierarchy of chordal graphs. *Discrete applied mathematics*, 124(1-3):67–71, 2002.
- [73] Lars Gottesbüren, Michael Hamann, Philipp Schoch, Ben Strasser, Dorothea Wagner, and Sven Zühlsdorf. Engineering exact quasi-threshold editing. In *Proceedings of the 18th International Symposium on Experimental Algorithms, SEA*, pages 10:1–10:14, 2020.
- [74] Carla Groenland, Gwenaël Joret, Wojciech Nadara, and Bartosz Walczak. Approximating pathwidth for graphs of small treewidth. *ACM Transactions on Algorithms*, 19(2):1–19, 2023.

- [75] Sylvain Guillemot, Frédéric Havet, Christophe Paul, and Anthony Perez. On the (non-)existence of polynomial kernels for P_l -free edge modification problems. *Algorithmica*, 65(4):900–926, 2013.
- [76] Jiong Guo. Problem kernels for NP-Complete edge deletion problems: split and related graphs. In *Proceedings of the 18th International Symposium on Algorithms and Computation, ISAAC*, pages 915–926, 2007.
- [77] Michel Habib, Ross McConnell, Christophe Paul, and Laurent Viennot. Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234(1-2):59–84, 2000.
- [78] Peter L. Hammer and Bruno Simeone. The splittance of a graph. *Combinatorica*, 1:275–284, 1981.
- [79] Frank Harary. A characterization of block-graphs. *Canadian Mathematical Bulletin*, 6(1):1–6, 1963.
- [80] Frank Harary, Emmanuel Loukakis, and Constantine Tsouros. The geodetic number of a graph. *Mathematical and Computer Modelling*, 17(11):89–95, 1993.
- [81] Irith Ben-Arroyo Hartman. Berge’s conjecture on directed path partitions—a survey. *Discrete mathematics*, 306(19-20):2498–2514, 2006.
- [82] Michael Held and Richard M. Karp. A dynamic programming approach to sequencing problems. *Journal of SIAM*, 10(1):196–210, 1962.
- [83] John Hopcroft and Robert E. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
- [84] Haim Kaplan, Ron Shamir, and Robert E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM Journal on Computing*, 28(5):1906–1922, 1999.
- [85] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.
- [86] Ioannis Katsikarelis, Michael Lampis, and Vangelis Th. Paschos. Structurally parameterized d-scattered set. *Discrete Applied Mathematics*, 308:168–186, 2022.
- [87] Ken-ichi Kawarabayashi and Bruce Reed. Computing crossing number in linear time. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, STOC*, pages 382–390, 2007.
- [88] Leon Kellerhals and Tomohiro Koana. Parameterized complexity of geodetic set. In *Proceedings of the 15th International Symposium on Parameterized and Exact Computation, IPEC*, pages 20:1–20:14, 2020.
- [89] William Kennedy, Guohui Lin, and Guiying Yan. Strictly chordal graphs are leaf powers. *Journal of Discrete Algorithms*, 4(4):511–525, 2006.
- [90] Eun Jung Kim and O-joung Kwon. A polynomial kernel for distance-hereditary vertex deletion. *Algorithmica*, 83:2096–2141, 2021.
- [91] Christian Komusiewicz and Johannes Uhlmann. A cubic-vertex kernel for flip consensus tree. *Algorithmica*, 68(1):81–108, 2014.
- [92] Tuukka Korhonen and Daniel Lokshtanov. An improved parameterized algorithm for treewidth. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC*, pages 528–541, 2023.
- [93] Stefan Kratsch and Magnus Wahlström. Two edge modification problems without polynomial kernels. *Discret. Optim.*, 10(3):193–199, 2013.
- [94] Stefan Kratsch and Magnus Wahlström. Compression via matroids: A randomized polynomial kernel for odd cycle transversal. *ACM Transactions on Algorithms*, 10(4):20:1–20:15, 2014.

- [95] Mirko Křivánek and Jaroslav Morávek. NP-hard problems in hierarchical-tree clustering. *Acta informatica*, 23(3):311–323, 1986.
- [96] Manuel Lafond. Recognizing k -leaf powers in polynomial time, for constant k . In *Proceedings of the 2022 Annual ACM-SIAM symposium on discrete algorithms, SODA*, pages 1384–1410, 2022.
- [97] Michael Lampis. A kernel of order $2k - c \log k$ for vertex cover. *Information Processing Letters*, 111(23):1089–1091, 2011.
- [98] Leonid Anatolevich Levin. Universal sequential search problems. *Problemy peredachi informatsii*, 9(3):115–116, 1973.
- [99] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.
- [100] Carlos V. G. C. Lima, Vinícius Fernandes dos Santos, João H. G. Sousa, and Sebastián Alberto Urrutia. On the computational complexity of the strong geodetic recognition problem. *preprint*, arXiv:2208.01796, 2022.
- [101] Yunlong Liu, Jianxin Wang, Jie You, Jianer Chen, and Yixin Cao. Edge deletion problems: Branching facilitated by modular decomposition. *Theoretical Computer Science*, 573:63–70, 2015.
- [102] Daniel Lokshantov. Wheel-free deletion is $W[2]$ -hard. In *Proceedings of the Third International Workshop on Parameterized and Exact Computation, IWPEC*, pages 141–147. Springer, 2008.
- [103] Nadimpalli V.R. Mahadev and Uri N. Peled. *Threshold graphs and related topics*. Elsevier, 1995.
- [104] Paul Manuel. Revisiting path-type covering and partitioning problems. *preprint*, arXiv:1807.10613, 2018.
- [105] Paul Manuel. On the isometric path partition problem. *Discussiones Mathematicae Graph Theory*, 41(4):1077–1089, 2021.
- [106] Paul Manuel, Sandi Klavžar, Antony Xavier, Andrew Arokiaraj, and Elizabeth Thomas. Strong edge geodetic problem in networks. *Open Mathematics*, 15(1):1225–1235, 2017.
- [107] Paul Manuel, Sandi Klavžar, Antony Xavier, Andrew Arokiaraj, and Elizabeth Thomas. Strong geodetic problem in networks. *Discussiones Mathematicae Graph Theory*, 40(1):307–321, 3920.
- [108] Lilian Markenzon and Christina Fraga Esteves Maciel Waga. New results on ptolemaic graphs. *Discrete Applied Mathematics*, 196:135–140, 2015.
- [109] Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57:747–768, 2010.
- [110] Dániel Marx and R.B. Sandeep. Incompressibility of H -free edge modification problems: Towards a dichotomy. *Journal of Computer and System Sciences*, 125:25–58, 2022.
- [111] Dániel Marx and Ildikó Schlotter. Obtaining a planar graph by vertex deletion. *Algorithmica*, 62(3-4):807–822, 2012.
- [112] Mauro Mezzini. An $O(mn^2)$ algorithm for computing the strong geodetic number in outer-planar graphs. *Discussiones Mathematicae Graph Theory*, 42(2):591–599, 2022.
- [113] James Nastos and Yong Gao. Familial groups in social networks. *Social Networks*, 35(3):439–450, 2013.
- [114] Assaf Natanzon. *Complexity and approximation of some graph modification problems*. Master’s Thesis, University of Tel-Aviv, 1999.
- [115] Assaf Natanzon, Ron Shamir, and Roded Sharan. A polynomial approximation algorithm for the minimum fill-in problem. *SIAM Journal on Computing*, 30(4):1067–1079, 2000.

- [116] Jun-Jie Pan and Gerard J. Chang. Isometric-path numbers of block graphs. *Information Processing Letters*, 93(2):99–102, 2005.
- [117] Dieter Rautenbach. Some remarks about leaf roots. *Discrete mathematics*, 306(13):1456–1461, 2006.
- [118] Neil Robertson and Paul D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.
- [119] Donald J. Rose, Robert E. Tarjan, and George S. Lueker. Algorithmic aspects of vertex elimination on graphs. *SIAM Journal on Computing*, 5(2):266–283, 1976.
- [120] Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.
- [121] Roded Sharan. *Graph modification problems and their applications to genomic research*. PhD thesis, Tel-Aviv University, 2002.
- [122] George Steiner. On the k -path partition of graphs. *Theoretical Computer Science*, 290(3):2147–2155, 2003.
- [123] Robert E. Tarjan. Depth-first search and linear graph algorithms. *SIAM journal on computing*, 1(2):146–160, 1972.
- [124] Marc Tedder, Derek Corneil, Michel Habib, and Christophe Paul. Simpler linear-time modular decomposition via recursive factorizing permutations. In *Proceedings of the 35th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 634–645, 2008.
- [125] Maximilian Thiessen and Thomas Gaertner. Active learning of convex halfspaces on graphs. In *Proceedings of the 35th Conference on Neural Information Processing Systems, NeurIPS*, pages 23413–23425, 2021.
- [126] Stéphan Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2):32:1–32:8, 2010.
- [127] Yngve Villanger, Pinar Heggernes, Christophe Paul, and Jan Arne Telle. Interval completion is fixed parameter tractable. *SIAM Journal on Computing*, 38(5):2007–2020, 2009.
- [128] Mingyu Xiao and Hiroshi Nagamochi. Exact algorithms for maximum independent set. *Information and Computation*, 255:126–146, 2017.
- [129] Jing-Ho Yan, Jer-Jeong Chen, and Gerard J Chang. Quasi-threshold graphs. *Discrete applied mathematics*, 69(3):247–255, 1996.

Modification et couverture de graphes : Approches paramétrées

Résumé : Cette thèse traite de problèmes de modification d'arêtes et de couverture de graphes par des plus courts chemins dans le cadre de la complexité paramétrée.

Dans un problème de modification d'arêtes nous avons en entrée un graphe quelconque et l'objectif est d'ajouter et de supprimer un nombre minimum d'arêtes à ce graphe de manière à ce que le graphe modifié appartienne à une classe de graphes donnée. Ce sont des problèmes généralement NP-difficiles à résoudre. Nous nous intéressons à un certain type d'algorithmes paramétrés pour aborder ces problèmes, appelés noyaux. Étant donné un paramètre k correspondant au nombre maximum de modifications autorisées, un noyau est un algorithme polynomial, qui transforme le graphe en entrée en une instance réduite équivalente, mais de taille bornée par une fonction de k (indépendante de la taille de l'entrée). Nous présentons des noyaux pour plusieurs problèmes, tels que TRIVIALY PERFECT EDITION, BLOCK GRAPH EDITION et STRICTLY CHORDAL EDITION.

Nous considérons également des problèmes de couverture de graphes par des plus courts chemins. Nous montrons que les graphes dont les sommets ou les arêtes peuvent être couverts par k plus courts chemins ont une largeur arborescente bornée par une fonction exponentielle en k . Cela nous permet de montrer que l'on peut décider si un graphe est couvrable par k plus courts chemins en temps XP, c'est à dire en temps $n^{f(k)}$ pour n le nombre de sommets du graphe et une certaine fonction f . Nous montrons que le problème ISOMETRIC PATH COVER WITH TERMINALS, où l'on spécifie en entrée les k paires d'extrémités des plus courts chemins recherchés, est même FPT, c'est à dire qu'il admet un algorithme de complexité $f(k) \cdot n^{O(1)}$ pour une certaine fonction f .

Mots-clés : Graphe, Complexité paramétrée, Algorithme de noyau, Modification de graphes, Couverture de graphes

Graph modification and covering: Parameterized approaches

Abstract: This thesis deals with problems of edge modification and graph covering by shortest paths in the context of parameterized complexity.

In an edge modification problem we have as input an arbitrary graph and the objective is to add and remove a minimum number of edges to this graph so that the modified graph belongs to a specified class. These are generally NP-hard problems. We are interested in a certain type of parameterized algorithm for tackling these problems, called kernels. Given a parameter k corresponding to the maximum number of modifications allowed, a kernel is a polynomial algorithm, which transforms the input graph into an equivalent reduced instance, but of size bounded by a function of k (independent of the size of the input). We present kernels for several problems, such as TRIVIALY PERFECT EDITION, BLOCK GRAPH EDITION and STRICTLY CHORDAL EDITION.

We also consider graph covering problems using shortest paths. We show that graphs whose vertices or edges can be covered by k shortest paths have a treewidth bounded by an exponential function in k . This allows us to show that we can decide whether a graph can be covered by k shortest paths in XP time, i.e. in $n^{f(k)}$ time for n the number of vertices of the graph and a certain function f . We show that the problem ISOMETRIC PATH COVER WITH TERMINALS, where we specify as input the k pairs of extremities of the shortest paths sought, is even FPT, i.e. it admits an algorithm of complexity $f(k) \cdot n^{O(1)}$ for a certain function f .

Keywords: Graph, Parameterized complexity, Kernelization algorithm, Graph modification, Graph covering