



HAL
open science

Stabilité et coloration des graphes sans P5

Grégory Morel

► **To cite this version:**

Grégory Morel. Stabilité et coloration des graphes sans P5. Mathématique discrète [cs.DM]. Université de Grenoble, 2011. Français. ⟨NNT : 2011GREN042⟩. ⟨tel-00651941v1⟩

HAL Id: tel-00651941

<https://theses.hal.science/tel-00651941v1>

Submitted on 14 Dec 2011 (v1), last revised 4 Apr 2012 (v3)

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.



HAL Authorization

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques et Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Grégory Morel

Thèse dirigée par **Zoltán Szigeti**
et codirigée par **Frédéric Maffray**

préparée au sein du Laboratoire **G-SCOP**
et de l'École Doctorale **MSTII**

Stabilité et coloration des graphes sans P_5

Thèse soutenue publiquement le **30 septembre 2011**,
devant le jury composé de :

M. Sylvain Gravier

Directeur de Recherche C.N.R.S., Institut Fourier, Président

M. Jean-Luc Fouquet

Professeur des Universités, Université d'Orléans, Rapporteur

M. Hamamache Kheddouci

Professeur des Universités, Université Claude Bernard Lyon 1, Rapporteur

M. Gábor Bacsó

Senior Researcher, Eötvös L. University (Budapest), Examineur

M. Zoltán Szigeti

Professeur des Universités, Grenoble INP, Laboratoire G-SCOP, Directeur de thèse

M. Frédéric Maffray

Directeur de Recherche C.N.R.S., Laboratoire G-SCOP, Co-Directeur de thèse



À ma famille...

REMERCIEMENTS

JE tiens à remercier en premier lieu Frédéric Maffray. Tout d'abord pour avoir accepté de m'encadrer en Master, puis de continuer comme co-directeur de cette thèse. Il ne saura jamais assez ma gratitude pour tout ce qu'il m'a appris en théorie des graphes, mais aussi pour sa disponibilité et surtout sa patience.

Toute ma gratitude va également vers Zoltán Szigeti qui a accepté d'être co-directeur de cette thèse. Bien que nous n'ayons été que peu amenés à travailler ensemble, les discussions que nous avons eues par ailleurs, qu'elles concernent la science ou d'autres centres d'intérêt communs (comme tout ce qu'il est possible de faire en une journée de 24 heures... chrono), ont toujours été très sympathiques.

J'adresse mes plus sincères remerciements à Jean-Luc Fouquet et Hamamache Kheddouci pour avoir accepté d'être les rapporteurs de cette thèse.

Je témoigne aussi toute ma reconnaissance à Gábor Bacsó pour avoir accepté de participer à mon jury et dont les remarques ont contribué à améliorer la qualité et la clarté du présent manuscrit, et à Sylvain Gravier qui m'a fait l'honneur de présider mon jury.

Un merci également à Martin C. Golumbic, notre coauteur, qui nous a incités à étudier le problème présenté au chapitre 4.

Un merci aussi à Stéphane Bonnevey, qui a été mon professeur à l'I.U.T. de Bourg-en-Bresse puis à l'Université de Lyon, pour m'avoir transmis le goût de la théorie des graphes, pour ses conseils, son soutien, et sans qui je n'aurais peut-être pas continué dans cette voie. Sa présence lors de ma soutenance a été pour moi un grand honneur.

Je vous adresse un **immense** merci à vous tous, membres du laboratoire G-SCOP, que vous soyez permanents ou doctorants, administratifs ou même encore stagiaires, passés ou présents, pour tous ces bons moments passés ces dernières années, une convivialité que l'on ne retrouve clairement pas dans tous les laboratoires, et surtout pour la confiance que vous avez su me témoigner. Vous avez indéniablement contribué à faire de moi quelqu'un d'autre, et je l'espère meilleur.

Il serait illusoire de vouloir tous vous citer, mais je ne peux m'empêcher de citer pêle-mêle un certain nombre d'entre vous qui avez joué un rôle particulier ces quatre dernières années. Que ceux que je ne citerais pas

ici ne se sentent pas oubliés et qu'ils sachent qu'ils ont eux aussi une très grande importance.

Merci donc à Hélène, Valérie, Julien, Susann, Lilia, Marwane, Alexandre, Grégory, Damien, Valentin, Khaled, Marie-Anne, Denis (pour ses conseils, et pas seulement au backgammon), Olivier(s), Jean-Philippe, Gülgün, Coralie, Laura, Bertrand (à qui j'aurais tenté en vain d'expliquer que, non, *Le mariage de Figaro* n'a pas été écrit par Éric Zemmour pour un certain président), Samuel, Christophe-Marie, Ariel, Pierre(s), Abed, Mireille, Van-Dat, Louis, Amandine, Roxane, Souad, Kheira, ainsi qu'à tous les membres de l'équipe Optimisation Combinatoire ; un très grand merci à Yannick Frein pour nous avoir permis, au travers de l'A-DOC, d'instaurer une ambiance si particulière à ce laboratoire, mais également pour les échanges que nous avons eus, et les conseils qu'il m'a prodigués.

Un merci, encore, à toute l'équipe de l'école de Génie Industriel, à commencer par Jeanne Duvallat. J'ai pu réaliser, avec toute la liberté dont je pouvais espérer, un souhait qui était celui d'enseigner, de transmettre un savoir. Pendant quatre ans, pas une seule fois l'on ne m'a fait ressentir que je n'étais qu'un « moniteur ». Au contraire, tous les échanges ont toujours été d'égal à égal, et la confiance que l'on m'a accordée à toujours été une grande source de motivation. Merci donc aux enseignants, aux membres du service administratif, et à tous ces étudiants, qui sont pour certains devenus bien plus que des « connaissances », qui, par un mot, un mail, parfois juste un signe de reconnaissance, ont contribué à ce que mon expérience d'enseignant à G.I. restera un immense bonheur.

Je n'oublie pas non plus mes amis, Cédric et Stéphanie ; il est bien trop souvent rare de savoir que l'on peut compter sur quelqu'un. Je leur souhaite beaucoup de joie avec leur petit Maxime.

Je ne peux évidemment terminer ces remerciements sans un mot pour ma famille : mes parents, mon frère, ma grand-mère, pour leur soutien depuis tant d'années... même s'ils n'ont pas toujours compris l'intérêt que l'on peut avoir à colorer des graphes.

Grenoble, le 30 septembre 2011.

TABLE DES MATIÈRES

TABLE DES MATIÈRES	vii
LISTE DES FIGURES	xi
Introduction	1
1 NOTIONS DE BASE	7
1.1 THÉORIE DES GRAPHES	9
1.1.1 Concepts de base	9
1.1.2 Représentations des graphes en machine	13
1.1.3 Graphes particuliers	14
1.1.4 Sous-graphes	16
1.1.5 Stables et cliques	17
1.1.6 Coloration de graphe	18
1.1.7 Isomorphisme de graphes	20
1.1.8 Quelques classes de graphes	21
1.2 THÉORIE DE LA COMPLEXITÉ	25
1.2.1 Concepts de base	26
1.3 THÉORIE POLYÉDRALE	29
1.3.1 Concepts de base	29
1.3.2 Résolution des programmes linéaires	30
1.3.3 Approche polyédrale pour la résolution d'un problème d'optimisation combinatoire	31
CONCLUSION	33
2 PROBLÈME DU STABLE MAXIMUM : TECHNIQUES DE RÉSOLU- TION ET APPLICATIONS AUX GRAPHES SANS P_5	35
2.1 LE PROBLÈME DU STABLE MAXIMUM	37
2.1.1 Énoncé et applications	37
2.1.2 Problèmes connexes	38
2.1.3 Complexité	42
2.1.4 Quelques exemples de classes MIS-faciles	44
2.1.5 Théorèmes d'Alekseev	49
2.2 LA CLASSE DES GRAPHES SANS P_5	52
2.2.1 Propriétés des graphes sans P_5	52
2.2.2 Trouver une clique dominante dans un graphe sans P_5	54
2.2.3 Propriétés des graphes sans P_5 bipartis	57
2.3 LES GRAPHES AUGMENTANTS	59
2.3.1 Notion de chaîne S -augmentante	59
2.3.2 Méthode des graphes augmentants	60
2.3.3 Caractérisation des graphes augmentants	63

2.3.4	Comment trouver un graphe augmentant en temps polynomial?	64
2.3.5	Application aux graphes sans P_5	65
2.4	LA DÉCOMPOSITION MODULAIRE	72
2.4.1	Premières définitions et propriétés	72
2.4.2	Théorème de décomposition modulaire	75
2.4.3	Arbre de décomposition modulaire	75
2.4.4	Application aux cographes	77
2.4.5	Graphes sans P_5 de largeur de clique bornée	80
2.4.6	Application aux graphes sans P_5	82
2.5	LA STRUCTION	86
2.5.1	Fonctions pseudo-booléennes, posiformes et graphe de conflit	86
2.5.2	La struction	89
2.5.3	Construction directe	90
2.5.4	Complexité	91
2.5.5	D'autres types de transformations	91
2.5.6	Application aux graphes sans P_5	96
2.6	LA DÉCOMPOSITION PAR CLIQUES D'ARTICULATION	98
2.6.1	Présentation	98
2.6.2	Calcul d'un stable de poids maximum	100
2.6.3	Application aux graphes sans P_5	101
2.7	GÉNÉALOGIE DES GRAPHES SANS P_5	102
	CONCLUSION	107
3	GRAPHES SANS P_5 3-COLORABLES	109
3.1	À PROPOS DES GRAPHES SANS P_5 k -COLORABLES.	111
3.1.1	k -colorabilité des graphes sans P_5	111
3.1.2	Stable de poids maximum dans les graphes sans P_5 k -colorables	113
3.1.3	Caractérisation des graphes sans P_5 3-colorables	116
3.2	ALGORITHMES LINÉAIRES POUR LES GRAPHES SANS P_5 3-COLORABLES	116
3.2.1	Structures de données utilisées	117
3.2.2	Initialisation	118
3.2.3	Cordalité de \overline{G}	119
3.2.4	Cas où G contient un $2K_2$	120
3.2.5	Cas où G contient un C_5	125
3.2.6	Cas où G contient un \overline{C}_6	145
	CONCLUSION	150
4	CARACTÉRISATION DES GRAPHES CHAIN-PROBE	151
4.1	PROBLÈMES DE TYPE <i>sandwich</i> ET PROBLÈMES DE TYPE <i>probe</i>	153
4.1.1	Problèmes de type <i>sandwich</i>	153
4.1.2	Problèmes de type <i>probe</i>	154
4.2	CARACTÉRISATION DES GRAPHES CHAIN-PROBE	155
4.2.1	Les six graphes minimaux interdits	155
4.2.2	Preuve du théorème 4.1	157
4.2.3	Considérations algorithmiques	162
	CONCLUSION	163

CONCLUSION GÉNÉRALE	166
A ANNEXES	167
A.1 HISTORIQUE À TRAVERS CINQ MATHÉMATIENS ET CINQ PRO- BLÈMES	169
A.1.1 Euler et le problème des 7 ponts de Königsberg	169
A.1.2 Cayley et le dénombrement des arbres	170
A.1.3 Petersen et la théorie des graphes réguliers	172
A.1.4 Guthrie et le problème des quatre couleurs	175
A.1.5 Berge et les graphes parfaits	178
A.2 HISTOIRE DE LA THÉORIE DE LA COMPLEXITÉ	182
BIBLIOGRAPHIE	185
INDEX	211

LISTE DES FIGURES

1	Le graphe $S_{i,j,k}$.	2
1.1	Un exemple de pseudographe, avec des arêtes multiples (en vert) et une boucle (en bleu).	9
1.2	Un exemple de graphe orienté.	10
1.3	Un graphe et son complémentaire.	10
1.4	Un graphe et la construction de son graphe adjoint.	12
1.5	La griffe.	12
1.6	Un graphe (la maison) et sa matrice d'adjacence.	13
1.7	La matrice d'incidence du graphe de la figure 1.6.	14
1.8	Représentation par listes d'adjacence du graphe de la figure 1.6.	14
1.9	Quelques graphes particuliers.	15
1.10	Différence entre un stable <i>maximal</i> et un stable <i>maximum</i> .	18
1.11	Deux graphes isomorphes et la bijection explicitée.	20
1.12	Un graphe d'intervalles (source : Wikipédia)	23
1.13	Un graphe parfait, mais non parfaitement ordonnable.	25
2.1	Le graphe $S_{i,j,k}$, homéomorphe à la griffe (i, j, k peuvent être nuls).	50
2.2	(a) La chaise et (b) le taureau	51
2.3	Un exemple d'augmentation de stable. Le graphe de la figure (a) possède un stable de cardinalité 3 (sommets bleus); ce stable admet un graphe augmentant (représenté par les arêtes épaisses sur la figure (b)); les sommets de B sont les sommets bleus et les sommets de R sont les sommets rouges); la figure (c) montre le stable obtenu après augmentation.	61
2.4	Un drapeau.	64
2.5	Illustration de la définition 2.7. S est constitué des sommets noirs, et on a $H(v, S) = \{u_1, u_2\}$ et $H(w, S) = \{u_1, u_3\}$	66
2.6	(a) La patte et (b) le grillon.	68
2.7	Exemple de module dans un graphe	73
2.8	Un graphe à décomposer.	76
2.9	Décomposition modulaire du graphe de la figure 2.8 et arbre correspondant.	76
2.10	Un cographe et son arbre.	78
2.11	(a) La 2-patte (b) le cerf-volant (c) la gemme (d) le diamant	82
2.12	Les graphes A et domino, et leur complémentaire.	85
2.13	Recouvrement des arêtes d'un graphe par des bipartis complets.	88
2.14	Struction du graphe G .	89

2.15	Application de l'algorithme 6.	90
2.16	(a) L'antenne et (b) le filet.	91
2.17	Repliage de sommet.	92
2.18	Struction totale de G avec $H = G[1, 2, 3]$ et $p = 2$	93
2.19	Démagnétisation de l'aimant $\{u, v\}$	95
2.20	(a) $K_{3,3} - e$ et (b) la maison jumelée.	97
2.21	(a) Un graphe et (b), (c) deux décompositions par cliques d'articulation possibles.	99
2.22	Le graphe $\overline{P_2 + P_3}$	102
2.23	Diagramme de relation d'inclusion entre différentes classes de graphes sans P_5	103
2.24	Le graphe $T_{i,j,k}$	106
3.1	Le graphe de Clebsch, un graphe sans P_6 , mais non 3- colorable.	111
3.2	Connaissances actuelles sur la complexité du problème de la k -colorabilité des graphes sans P_t	112
3.3	Un exemple de graphe 2-colorable, mais pas 2-listes-colorable.	113
3.4	Les 12 sous-graphes minimaux sans P_5 non 3-colorables.	116
4.1	Les 6 graphes minimaux qui ne sont pas chain-probe	157
4.2	Forme générale d'un graphe <i>chain-probe</i> contenant un P_6	161
4.3	Forme générale d'un graphe <i>chain-probe</i> contenant un P_5 mais pas de P_6	162
A.1	Les sept ponts de Königsberg.	169
A.2	Un arbre au sens de la théorie des graphes.	171
A.3	Le graphe de Petersen, dans sa forme originale et tel qu'on le représente aujourd'hui.	178

Avertissement au lecteur

Bien que nous les ayons manipulés, tordus, contractés, étirés, séparés, colorés... aucun des graphes qui suivent n'a souffert au cours de ces recherches.

INTRODUCTION

Sans que l'on en soit toujours conscient, la Théorie des Graphes est aujourd'hui très présente dans notre société moderne. Cette branche des mathématiques, dont on fait remonter l'origine à Euler, a connu un essor spectaculaire au cours des cinquante dernières années, notamment grâce aux travaux de Claude Berge qui a grandement participé à sa diffusion. Parce qu'elle permet de modéliser aussi bien des problématiques de réseaux informatiques que de réseaux routiers, de transport de marchandises que d'emplois du temps, d'électronique que de mécanique du solide, la théorie des graphes a bénéficié et bénéficie encore d'un engouement considérable non seulement de la part des mathématiciens, mais également de la communauté scientifique toute entière : on observe ainsi depuis quelques années un grand nombre de publications ayant trait à des problèmes en biochimie, en génétique ou encore en sociologie, en lien direct avec la théorie des graphes.

Le plus souvent, ces problèmes se modélisent en la recherche d'une structure optimale particulière dans un graphe, structure qui doit satisfaire un certain nombre de contraintes. Par exemple, il peut s'agir d'identifier un *couplage maximum* (i.e. un ensemble d'arêtes deux à deux non adjacentes de taille maximum), un *stable maximum* (i.e. un ensemble de sommets deux à deux non adjacents de taille maximum), ou encore une *coloration minimum* (i.e. une partition de taille minimum des sommets en stables).

Parfois, on a la chance de disposer d'algorithmes « efficaces » (c'est-à-dire qui exécutent un nombre d'instructions majoré par un polynôme en la taille du problème, avant de se terminer) pour résoudre un problème donné. Par exemple, le problème du couplage maximum peut être résolu en temps polynomial grâce à l'algorithme d'Edmonds [121].

Mais très souvent, et c'est le cas pour les problèmes du stable maximum et de la coloration minimum, la difficulté vient justement de ce qu'un tel algorithme n'existe pas (ou du moins, la communauté s'accorde désormais à penser qu'il ne peut exister). On parle de *problèmes NP-difficiles*, et on se tourne alors vers d'autres solutions, comme les algorithmes d'approximation ou les métaheuristiques.

Il existe un troisième type de problèmes : ce sont ceux pour lesquels on ne sait pas encore s'ils sont soit polynomiaux, soit NP-difficiles (voire même s'ils n'appartiendraient pas à une troisième classe « intermédiaire »). Parmi eux se trouve le problème de l'isomorphisme de deux

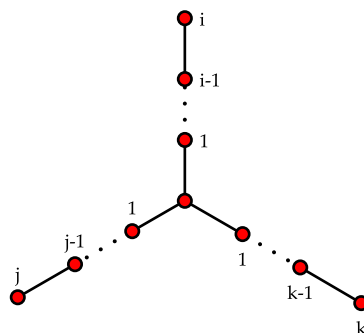


FIGURE 1 – Le graphe $S_{i,j,k}$.

graphes : peut-on déterminer « efficacement » si deux graphes sont les mêmes, ou ce problème est-il « difficile » ? Cette question n'a, à l'heure actuelle, pas de réponse. Un autre problème est dans ce cas : il s'agit du *problème du stable maximum dans les graphes sans P_5* . Le problème du stable maximum (aussi noté MIS), dont on trouve des applications dans les problèmes d'emplois du temps ou d'affectation de ressources, est bien connu pour être \mathcal{NP} -difficile dans un graphe quelconque, mais polynomial dans de nombreuses classes de graphes particuliers. En 1982, Alekseev a donné une condition suffisante pour qu'il soit \mathcal{NP} -difficile :

Théorème 0.1 Si un graphe H n'est pas isomorphe au graphe $S_{i,j,k}$ de la figure 1, le problème du stable maximum est \mathcal{NP} -difficile dans la classe des graphes ne contenant pas H .

Cependant, ce théorème ne permet pas de statuer sur la complexité de MIS dans la classe des graphes sans H quand H est justement de la forme $S_{i,j,k}$. Au fil des ans, les chercheurs ont réussi à exhiber un algorithme polynomial pour chaque configuration de $S_{i,j,k}$ contenant au plus quatre sommets (le résultat le plus célèbre étant sans doute le cas du graphe $S_{1,1,1}$, plus communément appelé *griffe*).

Lorsque $S_{i,j,k}$ contient cinq sommets, il n'y a que deux configurations possibles (à isomorphisme près) : $S_{1,1,2}$ (plus communément appelé *chaise*) et le graphe $S_{0,2,2}$ (appelé P_5). Longtemps, les statuts des problèmes du stable maximum dans la classe des graphes sans chaise et dans la classe des graphes sans P_5 sont restés indéterminés. En 1999, Alekseev a publié un algorithme polynomial pour une sous-classe commune, la classe des graphes sans chaise **et** sans P_5 ; enfin, en 2004, le même Alekseev a réussi à montrer que le problème est polynomial dans la classe des graphes sans chaise. En revanche, jusqu'à aujourd'hui, personne n'a été en mesure de confirmer ou d'infirmer que MIS est polynomial dans la classe des graphes sans P_5 . Autrement dit, à ce jour,

Observation 0.1 La classe des graphes sans P_5 est la plus petite classe de graphes définie par un seul sous-graphe connexe induit et interdit pour laquelle le statut de MIS est indéterminé.

Cette observation est le point de départ de cette thèse. Nos contributions consistent essentiellement en une solution à deux problèmes liés aux graphes sans P_5 :

- un algorithme linéaire permettant de colorer et calculer un stable maximum dans une sous-classe des graphes sans P_5 non encore explorée, la classe des *graphes sans P_5 3-colorables*, ainsi qu'un algorithme linéaire pour la reconnaissance de ces graphes, répondant ainsi à une question posée par d'autres chercheurs quant à l'existence d'un algorithme polynomial, et qui serait de complexité meilleure que $O(n^4)$;
- la solution du problème *probe* pour la classe des graphes bipartis sans P_5 . Ce problème, qui a des applications en intelligence artificielle, nous a été proposé par Martin Golumbic lors d'une visite à Grenoble. Il consiste à caractériser les graphes qui admettent un stable S tel qu'en rajoutant des arêtes entre des sommets de S , le graphe obtenu soit biparti sans P_5 .

Le présent manuscrit est organisé comme suit :

Chapitre 1 : Notions de base en théorie des graphes, théorie de la complexité et théorie polyédrale

Dans ce premier chapitre, nous rappelons toutes les définitions de théorie des graphes nécessaires à une bonne compréhension de la suite du mémoire ; en particulier nous redonnons les notions d'ensemble stable maximum, de coloration, et certaines classes de graphes que nous rencontrerons souvent dans les chapitres suivants. Nous donnons également un bref aperçu de la théorie de la complexité des algorithmes en insistant sur les notions fondamentales de classes de problèmes \mathcal{P} et \mathcal{NP} . Nous présentons enfin, très rapidement, les bases de la théorie polyédrale et dans quels cas elle permet de résoudre efficacement des problèmes d'optimisation combinatoire habituellement difficiles.

Chapitre 2 : Le problème du stable maximum : techniques de résolution et applications aux graphes sans P_5

Comme nous l'avons mentionné ci-dessus, le problème du stable maximum (MIS) est un problème aux très nombreuses applications pratiques, mais \mathcal{NP} -difficile en général. Nous commençons par dresser une liste de problèmes qui lui sont étroitement liés, et nous redonnons la brève démonstration de sa \mathcal{NP} -difficulté. Ensuite nous rappelons qu'il est polynomial dans un certain nombre de classes, notamment dans les graphes sans griffe, ce qui nous amène au théorème plus général d'Alekseev, et finalement à ce problème, encore ouvert, du stable maximum dans la classe des graphes sans P_5 , classe dont nous rappelons les propriétés. Enfin, nous dressons un panorama des différentes techniques développées pour résoudre le problème du stable (quand c'est possible), et leurs applications aux graphes sans P_5 , ainsi qu'un diagramme synthétisant les relations d'inclusions entre les différentes classes présentées tout au long

de ce chapitre. Nous avons essayé d'établir une synthèse aussi exhaustive que possible, afin de faire de ce document une référence (sans prétention aucune!) que le lecteur pourra consulter quand il voudra vérifier un résultat en rapport avec cette classe de graphes.

Le lecteur trouvera également dans ce chapitre deux de nos résultats : tout d'abord une amélioration, grâce à une meilleure utilisation des structures de données, de l'algorithme permettant d'identifier une clique dominante dans un graphe sans P_5 et sans C_5 , ramenant ainsi sa complexité de $\mathcal{O}(n^3)$ à $\mathcal{O}(m)$; puis, un algorithme polynomial permettant de calculer un stable de poids maximum dans une certaine classe de graphes (résultat qui avait en fait été énoncé auparavant sous une autre forme par Hoáng et Reed).

Chapitre 3 : Les graphes sans P_5 3-colorables

Quelques résultats concernant les problèmes du stable maximum et de la k -colorabilité des graphes sans P_5 (lorsque k est fixé) ont commencé à être publiés ces dernières années. Nous nous sommes intéressés au cas des graphes sans P_5 3-colorables, que nous avons caractérisés en terme d'une liste de douze graphes minimaux interdits. Nous avons également été en mesure d'élaborer des algorithmes de complexité *linéaire* pour résoudre les problèmes de la reconnaissance, du stable de poids maximum et de la coloration minimum de ces graphes. Une conséquence majeure de ces résultats est une diminution d'une unité de la complexité de l'algorithme connu pour résoudre le problème du stable de poids maximum dans les graphes k -colorables, pour tout $k > 3$; il s'agit également d'une réponse positive à une question laissée ouverte par une autre équipe de chercheurs.

Chapitre 4 : La caractérisation des graphes *chain-probe*

Un graphe est dit \mathcal{C} -*probe* s'il est possible de rajouter des arêtes entre certains sommets de sorte que le graphe résultant appartienne à la classe \mathcal{C} . Le problème qui consiste à déterminer si un graphe est \mathcal{C} -*probe* est un cas particulier d'un autre problème dit du \mathcal{C} -sandwich, et a été étudié à l'origine pour résoudre un problème en génétique (lorsque \mathcal{C} est la classe des graphes d'intervalles). Lors d'une visite à Grenoble, Martin C. Golumbic a soumis l'idée d'étudier la classe des graphes *chain-probe* dont la définition sera donnée en temps utile, et qui a des applications en intelligence artificielle, et plus précisément en raisonnement automatisé. Nous avons été en mesure de caractériser cette classe, en montrant qu'il existe six graphes minimaux qui n'en font pas partie. Nous prouvons également dans ce chapitre qu'il est possible de réduire considérablement la complexité de l'algorithme de reconnaissance des graphes de cette classe, par rapport à l'algorithme trivial qu'implique la caractérisation.

Annexe : La théorie des graphes à travers cinq mathématiciens et cinq problèmes

Cette partie du manuscrit s'adresse aussi bien au néophyte de la théorie des graphes qu'au lecteur déjà familier de ces notions. Lorsque nous avons

commencé la rédaction du présent mémoire, nous avons dans l'esprit de juste mentionner rapidement au début du premier chapitre le problème des ponts de Königsberg résolu par Euler, considéré comme le premier résultat en théorie des graphes, puis quelques résultats majeurs comme le théorème des graphes parfaits. Nous voulions également signaler les réponses à des questions que nous nous sommes posées : *D'où vient le mot graphe ?*, *Pourquoi Berge a-t-il parlé de graphes parfaits ?* Pour répondre à ces questions, nous sommes partis à la recherche des tout premiers articles en théorie des graphes (celui d'Euler sur les ponts de Königsberg, l'article de Sylvester, paru en 1878 dans *Nature*, où apparaît pour la première fois le mot *graphe* dans le sens où nous l'entendons aujourd'hui, ...). Rapidement, de citations en références, ces quelques questions ont évolué vers un véritable et passionnant travail de recherche documentaire, dont la synthèse est présentée dans 22 pages retraçant les grandes étapes de la théorie des graphes à travers cinq mathématiciens. Nous faisons ainsi mention à plus de 45 articles, écrits pour la plupart avant 1950 ; et, pour satisfaire encore plus la curiosité du lecteur, nous indiquons dans la bibliographie les liens vers des sites web permettant de consulter les articles en ligne (libres de droits, parfois réimprimés, traduits), à chaque fois que nous avons été en mesure de les trouver (ce qui parfois n'a pas été sans peine !), afin de les rendre directement accessibles. Il pourra ainsi (re)découvrir en un clic les premiers articles de Cayley sur les arbres, ou encore les travaux de Petersen sur les facteurs. C'est, à notre connaissance, l'une des rares synthèses de ce type. Le choix des mathématiciens présentés est bien sûr subjectif ; on pourra ainsi s'étonner que König ne soit pas mis plus en avant, tant il a contribué à redynamiser la recherche en théorie des graphes et faire de cette dernière une discipline mathématique à part entière. La seule explication tient uniquement aux coupes inévitables qu'impose la taille d'un tel document.

NOTIONS DE BASE



SOMMAIRE

1.1	THÉORIE DES GRAPHES	9
1.1.1	Concepts de base	9
1.1.2	Représentations des graphes en machine	13
1.1.3	Graphes particuliers	14
1.1.4	Sous-graphes	16
1.1.5	Stables et cliques	17
1.1.6	Coloration de graphe	18
1.1.7	Isomorphisme de graphes	20
1.1.8	Quelques classes de graphes	21
1.2	THÉORIE DE LA COMPLEXITÉ	25
1.2.1	Concepts de base	26
1.3	THÉORIE POLYÉDRALE	29
1.3.1	Concepts de base	29
1.3.2	Résolution des programmes linéaires	30
1.3.3	Approche polyédrale pour la résolution d'un problème d'optimisation combinatoire	31
	CONCLUSION	33

CE premier chapitre fournit au lecteur les bases en théorie des graphes, en théorie de la complexité et en théorie polyédrale nécessaires à la bonne compréhension des sujets abordés dans la suite de ce mémoire. La première partie, consacrée à la théorie des graphes, reprend ainsi ce qu'on peut trouver dans la majeure partie des ouvrages d'introduction sur le sujet. Nous nous sommes notamment efforcés à employer les notations et noms les plus usuels en français; toutefois nous avons jugé utile d'adjoindre, pour chaque terme nouvellement introduit, sa traduction anglophone, mise en évidence par une *emphase* et entre parenthèses. Dans la seconde partie, dédiée aux notions fondamentales de la théorie de la complexité, le lecteur pourra retrouver les concepts d'*algorithme polynomial*, de *problème NP-complet* ou encore de *réduction* que l'on rencontre en informatique. Enfin, la troisième partie donne les bases nécessaires à la compréhension de l'approche polyédrale pour la résolution d'un problème d'optimisation combinatoire.

1.1 THÉORIE DES GRAPHES

1.1.1 Concepts de base

Premières définitions

Intuitivement un **graphe** (*graph*) est un ensemble de points ou **sommets** (*vertices*) (que nous supposerons toujours non vide et fini dans la suite) dont certaines paires sont reliées, formant ainsi les **extrémités** (*endpoints*) d'une **arête** (*edge*). Plus formellement :

Définition 1.1 Un graphe G est un couple (V_G, E_G) où :

- V_G est l'ensemble des sommets,
- $E_G \subseteq \{\{x, y\}, x \in V_G, y \in V_G\}$.

Lorsque le contexte ne laissera aucune place à l'ambiguïté quant au graphe auquel il sera fait référence, nous écrirons plus simplement V et E . Le nombre de sommets d'un graphe, ou **ordre** (*order*), égal donc à $|V|$, est plus généralement noté n ; le nombre d'arêtes est quant à lui noté m .

Notons que la définition 1.1 laisse la possibilité pour une arête d'avoir ses deux extrémités identiques; une telle arête est appelée **boucle** (*loop*). De plus, pour de nombreuses applications il peut être utile d'avoir plusieurs arêtes ayant mêmes extrémités; on parle alors d'**arêtes multiples** (*multiple edges*). Un graphe ayant des arêtes multiples est un **multigraphe** (*multigraph*); s'il contient également des boucles, on parle de **pseudographe** (*pseudograph*)¹ (cf. figure 1.1). Un graphe ne contenant ni boucle ni arêtes multiples est qualifié de **simple** (*simple graph*).

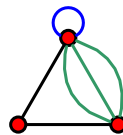


FIGURE 1.1 – Un exemple de pseudographe, avec des arêtes multiples (en vert) et une boucle (en bleu).

Il est également courant de rencontrer des graphes pour lesquels on distingue l'extrémité initiale de l'extrémité terminale des arêtes, c'est-à-dire que l'on ne considère plus des *paires* mais des *couples* de sommets; on parle alors d'**arcs** (*arcs*) et non plus d'arêtes. Un tel graphe est qualifié d'**orienté** (*directed graph* ou *digraph*) (cf. figure 1.2). Dans la suite, sauf mention contraire explicite, tous les graphes que nous considérerons seront simples et non orientés.

De plus, il est courant d'attribuer un nom ou un numéro aux sommets d'un graphe. Un tel procédé est appelé **étiquetage** (*labeling*). Ainsi, nous parlerons souvent « du sommet u » ou « du sommet v » d'un graphe. Ceci nous autorise donc à parler de l'arête $\{u, v\}$ pour évoquer, si elle existe,

1. L'emploi de ces deux termes est assez variable d'un ouvrage à l'autre.

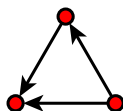


FIGURE 1.2 – Un exemple de graphe orienté.

l'arête ayant pour extrémités les sommets u et v ²; afin d'alléger les notations, nous nous autoriserons même à écrire plus simplement « l'arête uv ».

Enfin, nous serons fréquemment amenés à considérer des graphes **pondérés** (*weighted graphs*), c'est-à-dire muni d'une fonction de poids sur les sommets et/ou les arêtes.

Complémentaire

Quand on recherche les propriétés d'un graphe, il est parfois plus simple d'étudier son complémentaire :

Définition 1.2 Le **complémentaire** (complement) d'un graphe G est le graphe noté \overline{G} défini par :

- $V_{\overline{G}} = V_G$,
- l'arête uv ($u \neq v$) $\in E_{\overline{G}}$ ssi $uv \notin E_G$.

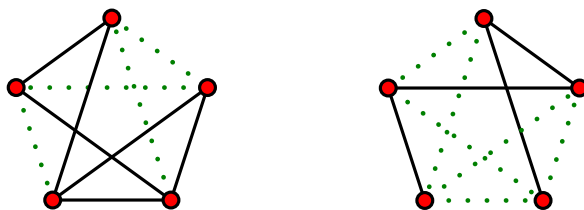


FIGURE 1.3 – Un graphe et son complémentaire.

Voisinage et degré

Deux sommets v et w formant les extrémités d'une même arête sont dit **adjacents** (*adjacent vertices*) ou **voisins** (*neighbors*), ce que l'on note $v \sim w$ (et $v \approx w$ dans le cas contraire). On définit alors le **voisinage** d'un sommet :

Définition 1.3

- Le **voisinage ouvert** ((open) neighborhood) d'un sommet v , noté $N(v)$, est l'ensemble des sommets qui lui sont adjacents.
- Le **voisinage fermé** (closed neighborhood) d'un sommet v , noté $N[v]$, est égal à $N(v) \cup \{v\}$.

2. Nos graphes étant supposés *simples*, on peut sans problème parler de l'arête plutôt que d'une arête.

Un sommet n'ayant aucun voisin est qualifié d'**isolé** (*isolated vertex*). On définit également le voisinage d'un **ensemble** de sommets :

Définition 1.4 Le voisinage (ouvert) d'un ensemble de sommets S , noté $N(S)$, est l'ensemble $\{x \notin S \mid \exists y \in S, xy \in E\}$.

Définition 1.5 Le **degré** (degree) d'un sommet v , noté $d(v)$, est égal au nombre $|N(v)|$ de ses voisins.

Le **plus petit degré** d'un graphe G est noté $\delta(G)$ et le **plus grand degré** est noté $\Delta(G)$; ils sont respectivement égaux au degré d'un sommet ayant le moins et le plus de voisins.

Il est facile de démontrer le résultat suivant, que l'on trouve déjà dans l'article d'Euler sur les sept ponts de Königsberg (cf. Annexe) :

Lemme 1.1 ((des poignées de mains) (*handshaking lemma*)) Dans un (pseudo)graphe,

$$\sum_{v \in V} d(v) = 2m$$

Démonstration. Il suffit de constater que chaque arête compte pour deux degrés (un pour chacune de ses extrémités). □

Graphe adjoint d'un graphe

Nous dirons également que deux arêtes sont **adjacentes** si elles possèdent une extrémité commune. On peut alors définir le « graphe d'adjacence des arêtes » de G , ou *graphe adjoint*³ de G .

Définition 1.6 Le **graphe adjoint** (line graph) d'un graphe $G = (V, E)$ est le graphe noté $L(G)$ tel que :

- $V_{L(G)} = E_G$,
- deux sommets de $L(G)$ sont adjacents si et seulement si les arêtes correspondantes dans G sont adjacentes.

La figure 1.4 illustre cette construction. Par définition, tout graphe possède un graphe adjoint; mais un graphe n'est pas forcément le graphe adjoint d'un autre graphe. Considérons par exemple le graphe de la figure 1.5, appelé **griffe** (*claw*) que nous rencontrerons souvent par la suite. Appelons a le sommet central, b , c et d les trois autres sommets, et essayons de reconstruire un graphe G dont la griffe serait le graphe adjoint. Dans

3. En français comme en anglais, il existe de nombreux termes (*derivative, edge-to-vertex dual, interchange graph, conjugate, derived graph* ou même encore *theta-obrazom...*) pour désigner un graphe adjoint. Bien que dans les deux langues *line graph* ait tendance à s'imposer, nous préférons employer dans la suite l'expression française.

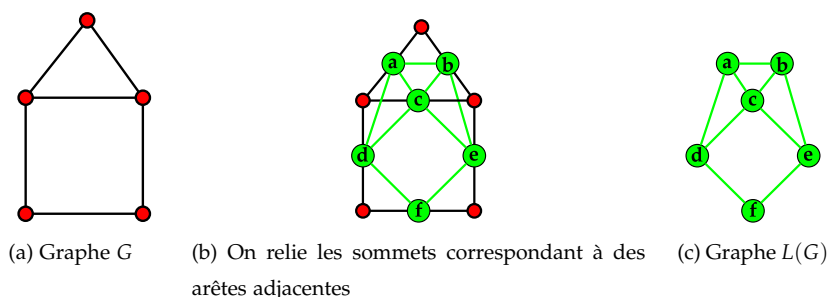


FIGURE 1.4 – Un graphe et la construction de son graphe adjoint.

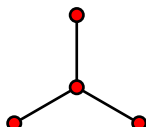


FIGURE 1.5 – La griffe.

G , a est une arête, dont les extrémités peuvent être notées u et v . Comme a est adjacent à b , on peut supposer que ces arêtes ont pour extrémité commune u dans G ; de même a et c ont une extrémité commune dans G , mais ce ne peut être u puisque b et c n'ont pas d'extrémité commune; c est donc v . On voit alors qu'il n'est pas possible que l'arête d soit adjacente à l'arête a sans être adjacente ni à l'arête b ni à l'arête c . La griffe ne peut donc pas être le graphe adjoint d'un graphe.

En 1970, Beineke [22] a caractérisé les graphes adjoints⁴, en donnant la liste complète des graphes minimaux qui ne sont pas des graphes adjoints, au nombre de neuf (et qui comprend donc la griffe); autrement dit, tout graphe qui n'est pas le graphe adjoint d'un autre graphe contient un de ces neuf graphes.

Il existe également un graphe qui peut être le graphe adjoint de plusieurs graphes : c'est le triangle (un graphe constitué de trois sommets adjacents deux à deux). Il est en effet facile de vérifier que le triangle est à la fois le graphe adjoint de la griffe et son propre graphe adjoint. Un résultat fondamental dû à Whitney en 1932 [343] affirme qu'il s'agit là d'une exception, et que dans tous les autres cas, il est possible de reconstruire un graphe à partir de son graphe adjoint.

Connexité

La connexité d'un graphe est une mesure importante de sa robustesse quand on le considère comme un réseau (réseau de transport, réseau informatique...).

Définition 1.7 Un graphe est *connexe* (connected) si pour toute paire de sommets il est possible de passer de l'un à l'autre par une suite de sommets adjacents.

4. Beineke avait déjà énoncé, sans démonstration, ce résultat en 1968.

La connexité définit une relation d'équivalence sur l'ensemble des sommets, et chacune des classes d'équivalence est appelée **composante connexe** (*connected component*) du graphe. Autrement dit, un graphe est connexe si et seulement s'il ne contient qu'une seule composante connexe. Intuitivement, le nombre de composantes connexes correspond au nombre de « morceaux » du graphe quand on le dessine. Un résultat classique énonce que :

Lemme 1.2 Pour tout graphe G , G est connexe ou \overline{G} est connexe.

Définition 1.8 Un *sommet* (resp. *ensemble*) *d'articulation* (cutvertex (resp. cutset)) est un sommet (resp. ensemble de sommets) tel que sa suppression rend le graphe non connexe.

1.1.2 Représentations des graphes en machine

Matrice d'adjacence

Définition 1.9 La *matrice d'adjacence* (adjacency matrix) d'un graphe d'ordre n est la matrice carrée A de taille $n \times n$ telle que $a_{ij} = 1$ s'il existe une arête entre les sommets i et j , et $a_{ij} = 0$ sinon.

Notons que la matrice d'un graphe non orienté est par conséquent symétrique. De plus, une matrice d'adjacence dépend de la numérotation des sommets du graphe qu'elle représente. Ainsi la notion de matrice d'adjacence n'a-t-elle de sens que pour un graphe *étiqueté*, et l'on parle alors bien de **la** matrice d'adjacence du graphe.

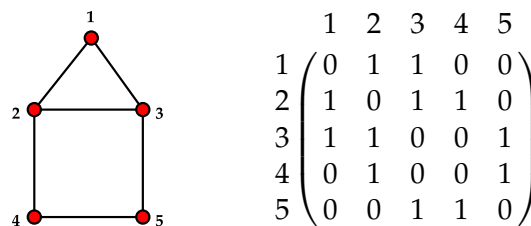


FIGURE 1.6 – Un graphe (la maison) et sa matrice d'adjacence.

Matrice d'incidence

Définition 1.10 La *matrice d'incidence* (incidence matrix) d'un graphe d'ordre n est la matrice B de taille $n \times m$ (m désignant comme vu plus haut le nombre d'arêtes du graphe) telle que $b_{ij} = 1$ si le sommet i est une extrémité de l'arête j , 0 sinon.

$$\begin{array}{c}
 \\
 \\
 \\
 \\
 \\
 \end{array}
 \begin{array}{cccccc}
 1-2 & 1-3 & 2-3 & 2-4 & 3-5 & 4-5 \\
 \begin{pmatrix}
 1 & 1 & 0 & 0 & 0 & 0 \\
 1 & 0 & 1 & 1 & 0 & 0 \\
 0 & 1 & 1 & 0 & 1 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 \\
 0 & 0 & 0 & 0 & 1 & 1
 \end{pmatrix}
 \end{array}$$

FIGURE 1.7 – La matrice d’incidence du graphe de la figure 1.6.

La figure 1.7 donne la matrice d’incidence du graphe de la figure 1.6.

Les représentations par matrice ont de nombreuses propriétés. Par exemple, une matrice d’adjacence permet de tester l’adjacence de deux sommets en une seule opération (il suffit de regarder la valeur de l’entrée correspondante). Elles sont cependant mal adaptées pour représenter des graphes peu denses (c’est-à-dire ayant peu d’arêtes en regard du nombre de sommets), car elles sont alors très creuses (c’est-à-dire contiennent beaucoup de zéros). Pour ces graphes, on leur préfère souvent une représentation par listes d’adjacence.

Listes d’adjacence

Définition 1.11 Une *liste d’adjacence* (*adjacency list*) est une structure de données dans laquelle on associe à chaque sommet sa liste de voisins.

Ainsi, on ne stocke que les informations « utiles » concernant l’adjacence dans le graphe, ce qui permet un gain d’espace mémoire non négligeable par rapport à une représentation par matrice. En revanche, déterminer si deux sommets sont adjacents requiert en général plus d’opérations avec une liste d’adjacence qu’avec une matrice d’adjacence, puisqu’il peut falloir parcourir la liste entière des voisins d’un sommet.

$$\begin{array}{l}
 1 \rightarrow 2 \rightarrow 3 \\
 2 \rightarrow 1 \rightarrow 3 \rightarrow 4 \\
 3 \rightarrow 1 \rightarrow 2 \rightarrow 5 \\
 4 \rightarrow 2 \rightarrow 5 \\
 5 \rightarrow 3 \rightarrow 4
 \end{array}$$

FIGURE 1.8 – Représentation par listes d’adjacence du graphe de la figure 1.6.

1.1.3 Graphes particuliers

Certains graphes, comme la griffe ou la maison que l’on vient de rencontrer, interviennent si souvent que les chercheurs leur ont attribué un nom ou une notation spéciale. Ce sont notamment les graphes complets, les graphes bipartis, les cycles et les chaînes.

Définition 1.12 Le *graphe complet* (*complete graph*) d’ordre n , noté K_n , est le graphe où chaque sommet est adjacent aux $n - 1$ autres sommets.

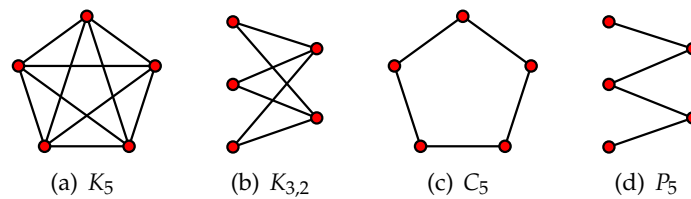


FIGURE 1.9 – Quelques graphes particuliers.

Il faudra veiller à ne pas confondre cette notion avec celle d'*ensembles de sommets complets*⁵ (resp. *anticomplets*) l'un à l'autre, utilisée pour signifier que tout sommet de l'un des ensembles est adjacent à tous les sommets (resp. aucun) de l'autre ensemble.

Définition 1.13 Un **graphe biparti** (bipartite graph) est un graphe dont on peut partitionner l'ensemble des sommets en deux ensembles A et B non vides tels que toute arête a une extrémité dans A et l'autre dans B . On note alors $G = (A, B; E)$.

Définition 1.14 Un **graphe biparti complet** est un graphe biparti $G = (A, B; E)$ où tout sommet de A est adjacent à tout sommet de B . On le note $K_{i,j}$, où $i = |A|$ et $j = |B|$.

Définition 1.15 Le **cycle** (cycle) d'ordre n , noté C_n , est le graphe connexe dont tous les sommets sont de degré 2.

Définition 1.16 La **chaîne** (path) d'ordre n , noté P_n , est le graphe connexe dont tous les sommets sont de degré 2, sauf les deux extrémités de la chaîne qui sont de degré 1.

La longueur d'un cycle ou d'une chaîne est le nombre d'arêtes qu'il/elle contient. Par abus de langage, on parlera plus simplement de « cycle (im)pair » pour désigner un cycle de *longueur* (im)paire.

La figure 1.9 illustre quelques-uns de ces graphes particuliers, que nous rencontrerons très souvent ultérieurement.

Enfin, nous utiliserons la notation $G_1 + G_2$ pour désigner le graphe ayant pour composantes connexes les graphes (connexes) G_1 et G_2 ; en particulier, nous noterons mK_2 le graphe possédant m composantes connexes formées d'un K_2 .

5. Les anglophones utilisent le terme de *join* pour définir cette relation d'adjacence.

1.1.4 Sous-graphes

Définition 1.17 Un graphe $G' = (V', E')$ est un **sous-graphe** (subgraph) de $G = (V, E)$ si $V' \subseteq V$ et $E' \subseteq E$; G est alors un **sur-graphe** (supergraph) de G' .

Définition 1.18 Un sous-graphe **couvrant** (ou parfois **graphe partiel**) (spanning subgraph) de $G = (V, E)$ est un sous-graphe $G' = (V, E')$ (c'est-à-dire qu'on peut l'obtenir à partir de G en supprimant uniquement des arêtes).

Définition 1.19 Le sous-graphe de $G = (V, E)$ **induit** par $V' \subseteq V$ (induced subgraph), noté $G[V']$, est le sous-graphe ayant pour ensemble de sommets V' et pour ensemble d'arêtes toutes les arêtes ayant leurs deux extrémités dans V' .

Par la suite, lorsque nous dirons qu'un graphe « en contient un autre », nous l'entendrons toujours au sens de sous-graphe *induit*, sauf mention explicite du contraire.

Exemple 1.1 (tiré de [24]) On représente la carte des routes nationales et autoroutes de France par un graphe : les villes sont les sommets, et les routes sont les arêtes. Si l'on ne considère que les autoroutes, on est en présence d'un sous-graphe couvrant de la carte. Si l'on se restreint aux villes de la région Rhône-Alpes, on a le sous-graphe induit par ces villes.

Graphes \mathcal{F} -libres et classes héréditaires

Définition 1.20 Étant donnée une famille de graphes \mathcal{F} , un graphe G est dit **\mathcal{F} -libre** ou **sans \mathcal{F}** (\mathcal{F} -free) s'il ne contient aucun élément de \mathcal{F} comme sous-graphe induit.

En particulier, si \mathcal{F} est réduite à un seul graphe H , nous écrirons *sans H* plutôt que *sans $\{H\}$* . L'objet même de cette thèse est l'étude de la classe des graphes sans P_5 . De plus, étant donnée une famille \mathcal{F} telle que G est \mathcal{F} -libre, seuls les éléments de \mathcal{F} minimaux par inclusion sont utiles : si un graphe G est sans H , a fortiori il ne peut pas non plus contenir les sur-graphes de H . Ces éléments minimaux de \mathcal{F} sont appelés **sous-graphes induits minimaux interdits** de G .

Définition 1.21 On dit qu'une classe de graphes \mathcal{C} est **héréditaire** (hereditary class) si tout sous-graphe induit d'un graphe de \mathcal{C} appartient lui-même à \mathcal{C} (i.e. \mathcal{C} est close par suppression de sommet).

Beaucoup de classes sont héréditaires : la classe des graphes bipartis, la classe des graphes adjoints et la classe des graphes sans P_5 en sont des illustrations.

Théorème 1.1 (Alekseev (1982) [2]) *Toute classe héréditaire peut être caractérisée par l'unique famille de ses sous-graphes induits interdits.*

1.1.5 Stables et cliques

Définition 1.22

- Un **stable** (stable set ou independent set) est un ensemble de sommets deux à deux non adjacents.
- Une **clique** (clique) est un ensemble de sommets deux à deux adjacents.

Ainsi une clique dans un graphe est un stable dans son complémentaire et inversement. De plus, tout graphe induit par une clique étant un graphe complet, les deux termes sont le plus souvent confondus ; de même, un graphe sans arête est régulièrement appelé *stable*⁶.

Les stables et les cliques interviennent dans de très nombreux problèmes pratiques (voir par exemple la section A.1.5 en annexe pour une application à la théorie de l'information). Pour la plupart de ces problèmes, on modélise la situation à l'aide d'un graphe, et la solution consiste à déterminer le plus grand stable ou la plus grande clique du graphe⁷ : on parle alors de stable ou clique *maximum*. Il convient d'ailleurs de veiller au point de vocabulaire suivant, légèrement différent de l'usage « normal » :

Définition 1.23

- Un ensemble (de sommets, d'arêtes...) est **minimal** (resp. **maximal**) pour une propriété P s'il ne contient (resp. n'est contenu dans) aucun ensemble vérifiant la propriété P .
- Un ensemble (de sommets, d'arêtes...) est **minimum** (resp. **maximum**) pour une propriété P s'il est de cardinalité minimale (resp. maximale) pour la propriété P .

Il est facile de voir qu'un ensemble maximum est aussi maximal ; mais un ensemble maximal n'est pas nécessairement maximum. Par exemple, le stable formé par les sommets rouges sur la figure 1.10(a) est maximal (il est en effet impossible d'ajouter à ce stable un autre sommet pour former un stable de cardinalité supérieure), mais il n'est pas maximum (puisque'il est possible de trouver un autre stable, de cardinalité supérieure (fig. 1.10(b))).

6. On trouve également dans la littérature les termes *graphe nul* ou *graphe vide* pour désigner un graphe sans arête, mais ces expressions désignent chez d'autres auteurs le graphe sans aucun sommet (et donc sans arête).

7. En toute rigueur, il conviendrait d'écrire plutôt « un plus grand stable » et « une plus grand clique. »

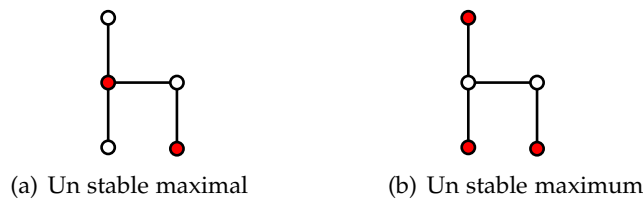


FIGURE 1.10 – Différence entre un stable maximal et un stable maximum.

La taille d'un stable maximum de G est notée $\alpha(G)$, et celle d'une clique maximum est notée $\omega(G)$. Naturellement, on a

$$\alpha(G) = \omega(\overline{G}).$$

Déterminer un stable maximum ou une clique maximum dans un graphe arbitraire n'est pas du tout trivial. Nous reviendrons plus longuement sur ces problèmes au chapitre 2.

1.1.6 Coloration de graphe

Pendant 120 ans, on s'est posé la question de savoir s'il est possible de colorier une carte géographique de sorte que deux pays frontaliers soient de couleur différente, tout en utilisant au plus quatre couleurs (cf. section A.1.4). Pour simplifier et sans perte d'information quant au problème que l'on vient d'évoquer, on peut représenter chaque pays de la carte par un sommet d'un graphe, et relier deux sommets s'ils correspondent à deux pays frontaliers. Colorier les pays de la carte pour vérifier une coloration se ramène alors à colorer⁸ les sommets dudit graphe. Cette idée a donné naissance au champ très vaste de la **coloration de graphes** (*graph coloring*), qui consiste à attribuer une « étiquette » à chaque sommet (on parle de **coloration de sommets** (*vertex coloring*)), ou à chaque arête (**coloration d'arêtes** (*edge coloring*)), voire aux deux (auquel cas on parle de **coloration totale** (*total coloring*)). Naturellement, les « étiquettes » peuvent être d'une autre nature que des couleurs, et on utilise souvent dans la pratique des nombres entiers ; néanmoins il est d'usage d'employer le mot *couleur* pour désigner une étiquette, quelle que soit sa nature. Formellement on a donc :

Définition 1.24 *Étant donné un graphe $G = (V, E)$, une coloration des sommets de G est une application $f : V \rightarrow C$, où C désigne l'ensemble des couleurs.*

Évidemment, attribuer des couleurs aux sommets d'un graphe n'aurait que peu d'intérêt si celles-ci n'étaient pas soumises à un certain nombre de contraintes. Par exemple, si l'on reprend le cas des cartes géographiques, deux sommets adjacents doivent être de couleurs différentes. Ce

8. Les dictionnaires que nous avons consultés indiquent :

- colorer : donner une certaine teinte à quelque chose,
- colorier : appliquer une couleur sur une surface.

Ainsi, nous utiliserons plutôt le verbe *colorer* pour désigner l'association d'une couleur à un sommet d'un graphe, ce qui par ailleurs cohérent avec l'usage qui veut que l'on parle de *coloration* et non de *coloriage* de graphe.

type de coloration, dite **coloration propre** des sommets (*proper coloring*), est tellement courant, que la littérature la désigne de plus en plus par le seul mot *coloration*, et nous nous conformerons à cet usage dans la suite ; on gardera toutefois à l'esprit que plusieurs problèmes importants (notamment en *théorie de Ramsey*) font intervenir des colorations qui ne sont pas nécessairement propres. On s'autorise même parfois à colorer les sommets avec plusieurs couleurs (coloration **fractionnaire** (*fractional coloring*)) ; et pour d'autres applications, il est nécessaire de restreindre les couleurs possibles pour chaque sommet à un sous-ensemble de l'ensemble des couleurs : ce type de coloration porte le nom de **coloration par liste** (*list coloring*).

Il est toujours possible de colorer un graphe « proprement » : il suffit d'attribuer une couleur différente à chaque sommet. Mais cette solution n'est clairement pas à privilégier, dans la mesure où pour la majorité des problèmes les couleurs symbolisent des ressources que l'on cherche justement à économiser. Ainsi, le problème de la coloration d'un graphe consiste-t-il à déterminer le nombre *minimum* de couleurs pour le colorer :

PROBLÈME DE LA COLORATION DE GRAPHE
(VERTEX COLORING (GT4)⁹)

INSTANCE : Un graphe $G = (V, E)$

QUESTION : Déterminer une coloration de G utilisant un nombre minimum de couleurs

Ce nombre est appelé **nombre chromatique** (*chromatic number*) de G , et est noté $\chi(G)$.

Définition 1.25

- Un graphe est *k-colorable* s'il est possible de le colorer avec k couleurs .
- Un graphe est *k-chromatique* si son nombre chromatique est égal à k .

Pour terminer ce paragraphe, notons que les sommets d'une même couleur étant deux à deux non adjacents, ils forment un stable. Autrement dit :

Proposition 1.1 *Un graphe est k-colorable si et seulement s'il est possible de partitionner son ensemble de sommets en k stables.*

On en déduit qu'un graphe est biparti si et seulement s'il est 2-colorable.

⁹. Cette référence renvoie au problème correspondant dans l'ouvrage de Garey et Johnson [150].

1.1.7 Isomorphisme de graphes

Définition 1.26 Deux graphes $G = (V_G, E_G)$ et $H = (V_H, E_H)$ sont **isomorphes** (isomorphic) s'il existe une bijection $f : V_G \rightarrow V_H$ telle que $\forall u, v \in V_G$, u et v sont adjacents dans G si et seulement si $f(u)$ et $f(v)$ sont adjacents dans H . On note alors $G \simeq H$.

Intuitivement, deux graphes sont isomorphes s'ils ont la même « structure », i.e. s'il est possible de déplacer les sommets de l'un pour qu'il soit la copie conforme de l'autre (au nom des sommets près). La figure suivante illustre cette définition :

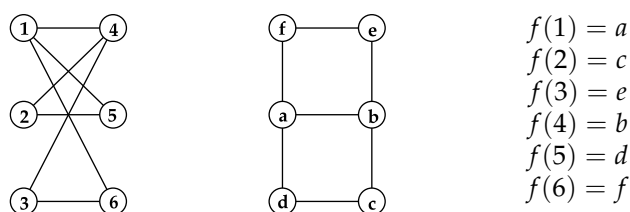


FIGURE 1.11 – Deux graphes isomorphes et la bijection explicitée.

Remarque 1.1 Lorsqu'un graphe est isomorphe à son complémentaire, on dit qu'il est **autocomplémentaire** (self-complementary). C'est le cas par exemple de P_4 ou de C_5 .

Problème de l'isomorphisme de graphes

Comme l'on peut s'en douter, le problème de l'isomorphisme de graphes consiste à déterminer si deux graphes sont isomorphes :

PROBLÈME DE L'ISOMORPHISME DE GRAPHES (GRAPH ISOMORPHISM (OPEN1))
INSTANCE : Un graphe $G = (V_G, E_G)$ et un graphe $H = (V_H, E_H)$
QUESTION : G est-il isomorphe à H ?

Cette question trouve des applications importantes en chimie (afin de savoir si deux composés chimiques sont identiques) ou encore en électronique (pour vérifier que deux représentations différentes de circuits sont équivalentes).

Le problème de l'isomorphisme de graphes est une curiosité en théorie de la complexité, dans le sens où il s'agit de l'un des rares problèmes appartenant à \mathcal{NP} dont on ne sait pas s'il est résoluble en temps polynomial, ni s'il est \mathcal{NP} -complet (cf. section 1.2.1). Le problème du stable maximum dans les graphes sans P_5 , sujet principal de cette thèse, est également dans ce cas.

Problème de l'isomorphisme de sous-graphes

Le problème précédent est un cas particulier du problème de l'isomorphisme de sous-graphes :

PROBLÈME DE L'ISOMORPHISME DE SOUS-GRAPHES (SUBGRAPH ISOMORPHISM (GT48))
INSTANCE : Un graphe $G = (V_G, E_G)$ et un graphe $H = (V_H, E_H)$
QUESTION : G contient-il un sous-graphe isomorphe à H ?

En effet, si $V_G = V_H$, ce problème revient à demander si G et H sont isomorphes. Notons pour terminer que le problème de l'isomorphisme de sous-graphes est l'un des tout premiers problèmes à avoir été prouvé \mathcal{NP} -complet.

1.1.8 Quelques classes de graphes

Nous présentons ici rapidement quelques classes de graphes particulières qui occupent une place majeure en théorie des graphes et que nous rencontrerons souvent par la suite, avec leurs propriétés et les liens entre elles.

Graphes bipartis, arbres et forêts

Nous avons déjà mentionné les graphes bipartis, définis comme étant les graphes dont l'ensemble de sommets peut être partitionné en deux stables (ou, de façon équivalente, les graphes 2-colorables). Il est intéressant de noter que les graphes bipartis peuvent également être caractérisés comme suit :

Proposition 1.2 *Un graphe est biparti si et seulement s'il ne contient aucun cycle de longueur impaire.*

Dans l'introduction, nous avons également évoqué les arbres, étudiés par Cayley :

Définition 1.27

- Une **forêt** (forest) est un graphe sans cycle.
- Un **arbre** (tree) est un graphe connexe sans cycle.

Autrement dit, une forêt est un ensemble d'arbres. De plus, puisqu'une forêt ne contient aucun cycle, elle ne contient *a fortiori* aucun cycle impair ; par conséquent, toute forêt (et donc tout arbre) est un graphe biparti.

Graphes parfaits

Les graphes parfaits ont été introduits par Berge qui essayait de résoudre un problème de théorie de la communication. L'histoire des graphes parfaits est donnée en annexe.

Définition 1.28 Un *graphe parfait* (perfect graph) G est un graphe vérifiant $\chi(H) = \omega(H)$ pour tout sous-graphe induit H de G .

En 1972, Lovász a démontré la *conjecture faible des graphes parfaits* : le complémentaire d'un graphe parfait est parfait. Dans son étude des graphes parfaits, Berge a été amené à définir une classe de graphes qui porte désormais son nom ¹⁰ :

Définition 1.29

- Un *trou* (hole) est un cycle de longueur au moins 4 ¹¹.
- Un *antitrou* (antihole) est le complémentaire d'un trou.
- Un *graphe de Berge* (Berge graph) est un graphe sans trou impair ni antitrou impair.

Berge avait conjecturé que la classe des graphes parfaits coïncide avec la classe des graphes de Berge : c'était la fameuse *conjecture forte des graphes parfaits*, démontrée en 2002 :

Théorème 1.2 (Chudnovsky, Robertson, Seymour et Thomas (2002) [77]) *Un graphe est parfait si et seulement s'il est de Berge.*

A titre d'exemple, nous pouvons montrer le résultat suivant :

Proposition 1.3 *Les graphes bipartis sont parfaits.*

Démonstration. (A partir de la définition des graphes parfaits) : un graphe biparti G ne contient aucun cycle impair, donc a fortiori aucune clique de taille supérieure ou égale à 3, i.e. $\omega(H) \leq 2$ pour tout sous-graphe induit H de G . Si $\omega(H) = 1$, $\chi(H) = 1$ aussi, et si $\omega(H) = 2$, $\chi(H) = 2$. Donc G est parfait.

(A partir du théorème fort des graphes parfaits) : un graphe biparti G ne contient aucun trou impair (car un trou est un cycle) ; il ne contient pas non plus d'antitrou de taille 5 (car c'est un cycle de taille 5), ni d'antitrou impair de taille supérieure à 7 (car un tel antitrou contient une clique de taille 3). Donc G est parfait. \square

Graphes cordaux

Définition 1.30 Un graphe est *cordal* ou *triangulé* (chordal graph) s'il ne contient pas de trou.

10. L'expression *Graphe de Berge* a été proposée par Chvátal et Sbihi dans [92].

11. Dans la littérature, il est très fréquent de lire qu'un trou est de longueur au moins 5.

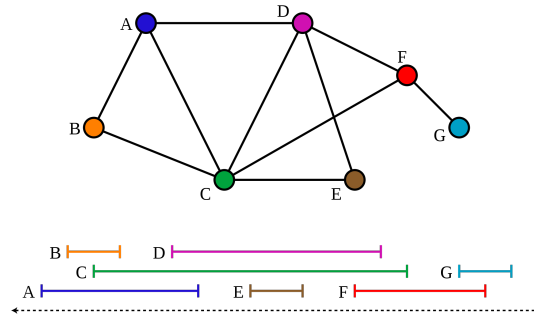


FIGURE 1.12 – Un graphe d'intervalles (source : Wikipédia)

Autrement dit, dans un graphe cordal tout cycle de longueur au moins 4 contient une corde (une arête joignant deux sommets non consécutifs du cycle), i.e. les seuls cycles induits sont des triangles.

Proposition 1.4 *Les graphes cordaux sont parfaits.*

Démonstration. Un antitrou d'ordre 5 est un trou d'ordre 5 ; et tout antitrou d'ordre 6 ou plus contient un trou d'ordre 4, donc si G est cordal, il ne contient pas de « grand » antitrou. Donc il est de Berge, et donc parfait. \square

Un sommet v est **simplicial** (*simplicial vertex*) si $N(v)$ est une clique. Un célèbre théorème de Dirac [117] affirme qu'on peut toujours trouver un sommet simplicial dans un graphe cordal (et même deux, si le graphe n'est pas une clique). Un **ordre d'élimination simplicial** (*perfect elimination ordering*) d'un graphe est un ordre $v_1 \dots v_n$ sur les sommets, tel que v_i est simplicial dans le graphe ne contenant que les sommets $v_i \dots v_n$. Fulkerson et Gross ont donné la caractérisation suivante des graphes cordaux :

Théorème 1.3 (Fulkerson et Gross (1965) [142]) *Un graphe est cordal si et seulement s'il admet un ordre d'élimination simplicial.*

Rose, Tarjan et Lueker ont introduit dans [306] un algorithme appelé **parcours en largeur lexicographique** (ou *Lex-BFS* pour *Lexicographic Breadth-First Search*) qui permet de trouver efficacement un ordre d'élimination simplicial d'un graphe cordal, et qui permet par conséquent de déterminer rapidement si un graphe est cordal ou non.

Graphes d'intervalles

Définition 1.31 *Un graphe d'intervalles* (interval graph) *est le graphe d'intersection d'un ensemble d'intervalles de la droite réelle, i.e. chaque sommet représente un intervalle et une arête relie deux sommets lorsque les intervalles correspondants s'intersectent.*

La figure 1.12 montre un graphe d'intervalles.

A partir de la définition ci-dessus, on peut établir la caractérisation suivante :

Proposition 1.5 *Un graphe est d'intervalles si et seulement s'il est possible d'ordonner toutes ses cliques maximales M_1, M_2, \dots, M_k de sorte que pour tout sommet $v \in (M_i \cap M_k)$ ($i < k$), $v \in M_j, \forall i \leq j \leq k$.*

Plusieurs algorithmes efficaces ont été proposés pour reconnaître les graphes d'intervalles [35], [181]. Enfin, notons que :

Proposition 1.6 *Les graphes d'intervalles sont cordaux, et donc parfaits.*

Graphes parfaitement ordonnables

Certains algorithmes bien connus, comme ceux de Prim ou de Kruskal pour calculer un arbre couvrant de poids maximum, sont des algorithmes dits **gloutons** (*greedy algorithm*), car leur principe consiste, à chaque étape de leur exécution, à effectuer le meilleur choix possible, indépendamment de l'incidence de ce choix sur la qualité du résultat final. Dans certains cas, comme les trois algorithmes que nous venons de citer, cette approche fonctionne bien et conduit effectivement à une solution optimale ; mais pour d'autres problèmes, elle ne conduit généralement qu'à une solution approchée, voire parfois à la pire solution ! Considérons par exemple l'algorithme glouton de coloration des sommets d'un graphe suivant :

Algorithme 1: Coloration gloutonne des sommets

Entrées : Un graphe $G = (V, E)$

Sorties : Une coloration de G

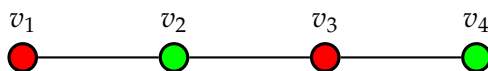
donner un ordre v_1, \dots, v_n aux sommets;

pour chaque v_i **faire**

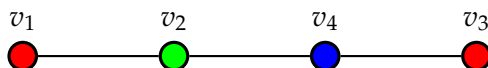
colorer v_i avec le plus petit numéro de couleur non utilisé par ses voisins

fin

Si l'on applique cet algorithme sur un P_4 ordonné comme suit, on obtient une 2-coloration, ce qui correspond à une solution optimale puisqu'un P_4 est un graphe biparti comportant des arêtes :



En revanche, si l'on exécute le même algorithme sur un P_4 ordonné de la manière suivante, la solution obtenue est une 3-coloration, et n'est donc pas optimale :



Evidemment, tous les graphes admettent un ordre tel que l'algorithme glouton donne une coloration optimale (puisque cette coloration existe). Chvátal [87] a alors posé la question suivante : *quels sont les graphes pour lesquels il existe un ordre parfait sur les sommets, c'est-à-dire un ordre tel que l'algorithme glouton donne une coloration optimale non seulement pour le graphe, mais également pour chacun de ses sous-graphes induits ?*, et a prouvé que la seule obstruction à un ordre parfait est le P_4 décrit ci-dessus. Autrement dit :

Proposition 1.7 (Chvátal (1984) [87]) *Un graphe G est **parfaitement ordonnable** (perfectly orderable graph) (i.e. il admet un ordre parfait) si et seulement s'il existe un ordre $<$ sur les sommets tel que G ne contienne pas de P_4 a - b - c - d avec $a < b$ et $d < c$.*

Chvátal a également démontré que les graphes parfaitement ordonnables sont parfaits. La réciproque est fautive : certains graphes parfaits (notamment, les graphes cordaux) sont parfaitement ordonnables, mais il en existe qui ne le sont pas, comme le graphe de la figure 1.13).

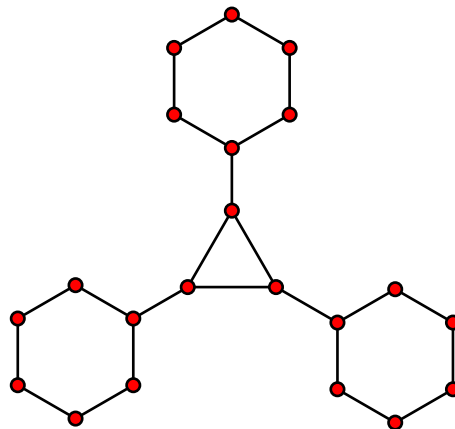


FIGURE 1.13 – Un graphe parfait, mais non parfaitement ordonnable.

1.2 THÉORIE DE LA COMPLEXITÉ

La théorie de la complexité est une branche des mathématiques et de l'informatique ayant pour cadre l'étude de la *difficulté* intrinsèque des problèmes algorithmiques, et qui vise à classer ces problèmes en fonction de cette difficulté. Ici, les mots « complexité » et « difficulté » ne se rapportent pas à la mise au point d'un algorithme de résolution, ou aux concepts avancés auxquels il peut faire appel (comme une structure de données élaborée), mais plutôt à la quantité de ressources à utiliser pour résoudre le problème. En ce qui nous concerne, les ressources consistent en le temps que met un algorithme à résoudre le problème (c'est sa complexité *temporelle*) et l'espace mémoire qu'il utilise au cours de son exécution (sa complexité *spatiale*)¹² ; mais il existe d'autres mesures de complexité, comme

¹². Il faut distinguer la théorie de la complexité de l'*analyse des algorithmes* : celle-ci a pour objet l'analyse de la quantité de ressources utilisées par un algorithme particulier

le nombre de portes logiques à utiliser pour la réalisation d'un circuit, ou encore la quantité d'information à transmettre dans le cadre de la théorie de la complexité de la communication. Le principal atout de la théorie de la complexité est que ces grandeurs sont exprimées indépendamment de tout dispositif physique concret ; au contraire, elle est basée sur un modèle de calcul abstrait, généralement une *machine de Turing*, ce qui permet de comparer facilement l'efficacité de deux algorithmes en s'affranchissant de considérations telles que la vitesse du processeur. On est quasiment sûrs aujourd'hui que certains problèmes nécessitent, pour leur résolution, des algorithmes dont le temps de calcul est bien supérieur à celui d'autres problèmes, et c'est en ce sens que l'on dira qu'ils sont « plus difficiles ».

1.2.1 Concepts de base

Définition 1.32 (Premières définitions)

Un **problème** est une question générale possédant des paramètres dont la valeur n'est pas connue.

Une **instance** d'un problème est obtenue en affectant une valeur à chacun de ses paramètres.

La **taille** d'une instance désigne généralement la quantité de cases mémoires nécessaires pour décrire les paramètres.

Exemple 1.2 Le **problème du voyageur de commerce** (ou TSP pour Traveling salesman problem) consiste, étant donné un ensemble de villes séparées par des distances connues, à trouver le plus court chemin qui relie toutes les villes, en ne passant qu'une seule fois par chaque ville. Une instance du TSP est donc un ensemble de n points (représentant les villes) définis chacun par un couple de coordonnées et la taille de cette instance est $2n + 1$ (il faut une case mémoire pour chaque coordonnée des n points et une autre pour stocker l'entier n).

Nous donnons maintenant des définitions plus formelles des notions abordées dans le paragraphe précédent. On commence par définir la notion de complexité et la notation « grand \mathcal{O} », puis les classes de complexité \mathcal{P} et \mathcal{NP} qui en découlent. Enfin nous présentons rapidement le concept d'algorithme d'approximation.

Problèmes de décision et d'optimisation

Définition 1.33 Un **problème de décision** est un problème auquel la réponse est oui ou non.

Définition 1.34 Un **problème d'optimisation** consiste à déterminer la meilleure solution parmi toutes les solutions réalisables.

pour résoudre un problème, alors que la première pose la question plus générale de déterminer quels algorithmes peuvent résoudre le problème en question.

En particulier, étant donné un ensemble fini $E = \{e_1, \dots, e_n\}$ pour lequel chaque élément possède un **poids** $c(e_i)$, et \mathcal{S} une famille de sous-ensembles de E , un problème d'**optimisation combinatoire** consiste à déterminer un élément $S \in \mathcal{S}$ de poids optimal (i.e. minimal ou maximal suivant le problème; le poids $c(S)$ de S est égal à $\sum_{e_i \in S} c(e_i)$). \mathcal{S} est l'**ensemble des solutions** du problème.

Complexité en temps et en espace

Définition 1.35 La **complexité temporelle** d'un algorithme correspond au nombre d'instructions élémentaires (opérations arithmétiques, comparaison, affectation...) effectuées au cours de son exécution.

Définition 1.36 La **complexité spatiale** d'un algorithme correspond au nombre de cases mémoires occupées par les données manipulées par l'algorithme au cours de son exécution.

En général, le temps d'exécution dépend de la taille de l'instance du problème considéré; en particulier, plus une instance est grande, plus le problème demandera de temps pour être résolu. Par exemple, si l'on considère un algorithme de tri d'un tableau d'entiers, le nombre d'instructions et l'espace occupé ne seront pas les mêmes si l'on travaille sur un tableau de 10 entiers ou sur un tableau de 10 000 entiers. On exprime donc le temps (ou toute autre mesure de complexité) en fonction de la *taille d'une instance générale* du problème considéré, souvent notée n . En algorithmique des graphes, en fonction du problème traité, on choisit généralement le nombre n de sommets, ou le nombre m d'arêtes du graphe. Mais même sur des instances de même taille, une complexité peut varier d'une instance à une autre: par exemple, rechercher une valeur dans un tableau demande plus de temps dans un tableau dont les éléments sont désordonnés que dans le même tableau trié. Aussi définit-on généralement une complexité en considérant la *pire* instance possible parmi toutes les instances de taille n , c'est-à-dire celle demandant le plus de ressources. Dans la suite, nous ne nous intéresserons qu'à la complexité *temporelle* des problèmes étudiés.

Notation « grand \mathcal{O} »

La notation **grand \mathcal{O}** , dite aussi *symbole de Landau*, décrit le comportement asymptotique d'une fonction, exprimé à l'aide d'une autre fonction généralement plus simple. Plus formellement, nous dirons que :

Définition 1.37 (Notation grand \mathcal{O}) $f(n) = \mathcal{O}(g(n))$ (« $f(n)$ est en grand \mathcal{O} de $g(n)$ ») quand $n \rightarrow \infty$ si et seulement si $\exists M > 0, n_0 \in \mathbb{R}$ tels que $\forall n \geq n_0, |f(n)| \leq M|g(n)|$.

Intuitivement, ceci signifie qu'à partir de n_0 et à un facteur constant près, f ne croît pas plus rapidement que g .

Exemple 1.3 Soit $f(n) = 6n^4 - 2n^3 + 5$. Choisissons $n_0 = 1$. Alors pour tout $n \geq n_0$, on a

$$\begin{aligned} |6n^4 - 2n^3 + 5| &\leq 6n^4 + |2n^3| + 5 \\ &\leq 6n^4 + 2n^4 + 5n^4 \\ &= 13n^4 \\ &= 13|n^4| \end{aligned}$$

Ainsi, en prenant $M = 13$, on a $f(n) = \mathcal{O}(n^4)$. Autrement dit, à un facteur constant près, $f(n)$ ne croît pas plus rapidement que n^4 .

Il est facile de voir qu'un polynôme $P(n)$ de degré k est toujours en $\mathcal{O}(n^k)$.

Classes de complexité \mathcal{P} et \mathcal{NP}

Définition 1.38 Un problème de décision P est dans la **classe \mathcal{P}** si, pour chacune de ses instances, dont la taille est notée n , il existe un réel positif k tel qu'il peut être résolu par un algorithme de complexité temporelle $\mathcal{O}(n^k)$, c'est-à-dire qu'il peut être décidé en temps polynomial.

Les problèmes de la classe \mathcal{P} sont dits *faciles*. Ce sont ceux que l'on sait résoudre *efficacement*.

Définition 1.39 Un problème de décision est dans la **classe \mathcal{NP}** ¹³ si l'on peut vérifier en temps polynomial qu'une solution pour une instance donnée est valide (ce que l'on appelle un *certificat du oui*).

Exemple 1.4 Considérons la version « décision » du problème du stable : étant donné un graphe G et un entier positif k , existe-t-il un stable de taille au moins k ? Ce problème est clairement dans \mathcal{NP} : si l'on dispose d'un ensemble S de sommets, on peut vérifier en temps polynomial que $|S| \geq k$ et que S est stable (par exemple en examinant la liste d'adjacence de chaque sommet de S).

Intuitivement, les problèmes de la classe \mathcal{NP} sont ceux que l'on peut résoudre en énumérant **l'ensemble des solutions possibles** (méthode « brutale ») et en les testant à l'aide d'un algorithme polynomial. Naturellement, si on peut résoudre un problème avec un algorithme polynomial, on peut aussi vérifier en temps polynomial que la solution fournie est bien une solution ; par conséquent $\mathcal{P} \subset \mathcal{NP}$.

Problèmes \mathcal{NP} -complets

Une **réduction** est une transformation d'un problème en un autre ; ceci permet de capturer la notion informelle de « problème au moins aussi difficile qu'un autre problème ». Plus précisément, si un problème X peut

¹³. Pour *Non-déterministe Polynomial* et non « Non-Polynomial ».

être résolu en utilisant un algorithme permettant de résoudre un problème Y , c'est que X n'est pas plus difficile que Y ; on dit alors que X se réduit à Y . Par exemple, le problème consistant à élever un nombre au carré se réduit au problème plus général de multiplication de deux nombres (ici, aucune transformation n'est nécessaire). Une réduction est **polynomiale** lorsque le processus de transformation peut se faire en temps polynomial.

Définition 1.40 Un problème X est **difficile** pour une classe de problèmes \mathcal{C} , ou **\mathcal{C} -difficile**, si tout problème de \mathcal{C} se réduit à X .

Autrement dit, il n'existe pas de problème de \mathcal{C} plus difficile que X , puisque tout algorithme résolvant X résout aussi n'importe quel problème de \mathcal{C} . En particulier, les problèmes difficiles pour la classe \mathcal{NP} forment la classe de problèmes **\mathcal{NP} -difficiles**. Lorsque, de plus, X appartient lui aussi à la classe \mathcal{C} , on dit qu'il est **complet** pour \mathcal{C} ; ceci signifie que X est l'un des problèmes les plus difficiles de \mathcal{C} (il peut en effet y avoir plusieurs problèmes de même difficulté).

Définition 1.41 (Problème \mathcal{NP} -complet) Un problème de décision est **\mathcal{NP} -complet** s'il vérifie les deux conditions suivantes :

- il appartient à la classe \mathcal{NP} ,
- tous les problèmes de la classe \mathcal{NP} se ramènent à celui-ci via une réduction polynomiale.

Tout problème d'optimisation combinatoire dont le problème de décision associé est \mathcal{NP} -complet est \mathcal{NP} -difficile. Le problème du stable maximum est \mathcal{NP} -difficile (nous redonnons la preuve à la section 2.1.3).

1.3 THÉORIE POLYÉDRALE

Dans cette dernière section, nous rappelons les définitions de base de la théorie polyédrale¹⁴. Notre but n'est aucunement ici de rentrer dans des détails techniques, mais juste de redonner au lecteur les notions nécessaires à la compréhension d'expressions que nous rencontrerons par la suite, telles que *problème de séparation* ou encore *polytope des stables*.

1.3.1 Concepts de base

Soit x un vecteur de \mathbb{R}^n . On dit que x est **combinaison linéaire** des points $x_1, \dots, x_k \in \mathbb{R}^n$ s'il existe k scalaires $\lambda_1, \dots, \lambda_k$ tels que $x = \sum_{i=1}^k \lambda_i x_i$. Si, de plus, $\sum_{i=1}^k \lambda_i = 1$ alors on dit que x est une **combinaison affine de ces points**, et une **combinaison convexe** si en outre $\lambda_i \geq 0 \forall i \in \{1, \dots, k\}$.

¹⁴. *Polyédral* est la transcription de l'anglais *polyhedral*. Il serait sans doute plus correct de parler de théorie *polyédrique*.

Définition 1.42 Soit S un ensemble non vide et fini de points de \mathbb{R}^n . L'*enveloppe convexe* (convex hull) des points de S , notée $\text{conv}(S)$, est l'ensemble des points de \mathbb{R}^n qui peuvent s'écrire comme combinaison convexe de points de S .

Un point qui ne peut pas s'écrire comme combinaison convexe d'autres points exclusivement est appelé **point extrême**. Il est facile de voir que les points extrêmes de $\text{conv}(S)$ sont des points de S .

Définition 1.43 Un *polytope* est l'enveloppe convexe d'un nombre fini de points de \mathbb{R}^n .

Les points extrêmes de S sont donc les sommets du polytope associé.

Un **polyèdre** P est un sous-ensemble de \mathbb{R}^n engendré par l'intersection d'un nombre fini de demi-espaces de \mathbb{R}^n . De manière équivalente, P est l'ensemble des solutions d'un système d'inégalités linéaires :

$$P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$$

où A est une matrice à m lignes et n colonnes et b un vecteur à m composantes. On dit que le système $Ax \leq b$ caractérise P .

Théorème 1.4 (Minkowski (1896), Steinitz (1916), Weyl (1935)) *Un polytope est un polyèdre borné et réciproquement.*

1.3.2 Résolution des programmes linéaires

Un **programme linéaire** est un problème de la forme

$$\max\{c^T x \mid Ax \leq b\}^{15}$$

où A est une matrice $m \times n$ de réels, et b et c sont des vecteurs donnés de \mathbb{R}^n ; le vecteur c est le vecteur des *poids* (ou des *coûts*) et la fonction $c^T x$ est appelée **fonction objectif**. Résoudre un programme linéaire consiste donc à trouver un point $x \in \mathbb{R}^n$ de poids maximal dans le polyèdre P caractérisé par $Ax \leq b$. Or, si P est un polytope, on a le résultat suivant :

Théorème 1.5 *Un programme linéaire borné admet une solution optimale en un sommet du polytope associé.*

Ceci implique un algorithme simple pour résoudre un programme linéaire : il suffit d'évaluer la fonction objectif en chacun des sommets du polytope et de retenir celui qui donne l'optimum. Cependant, un

¹⁵. Bien sûr, on peut remplacer *max* par *min*, mais tout problème de minimisation se ramène à un problème de maximisation.

polytope associé à un programme linéaire à n variables et m contraintes peut avoir jusqu'à $\binom{m}{n}$, soit un nombre potentiellement exponentiel, de sommets ! L'algorithme du **simplexe** élaboré par Dantzig [108] en 1947¹⁶, visite « intelligemment » les sommets de sorte à augmenter la valeur de l'optimum courant à chaque itération, ce qui explique son efficacité en pratique. Néanmoins, Klee et Minty [232] ont montré qu'il existe des programmes linéaires pour lesquels le simplexe doit visiter presque tous les sommets, et ainsi prouvé qu'il n'est pas polynomial.

Le premier algorithme polynomial pour résoudre des programmes linéaires fut mis au point en 1979 par Khachiyan [229], qui a eu l'idée d'adapter la **méthode de l'ellipsoïde** introduite par Shor [315, 316] pour minimiser des fonctions convexes. Sa complexité originale était en $\mathcal{O}(n^6 L^2)$, abaissée depuis à $\mathcal{O}(n^4 L^2)$, où L désigne le nombre de bits nécessaires pour coder les données (A , b et c). Bien que plus rapide que le simplexe sur les problèmes de Klee et Minty, il n'a jamais pu s'imposer en raison de sa piètre efficacité sur les problèmes « réels » et de sa grande instabilité numérique ; il n'en existe d'ailleurs aucune implémentation connue. Nous verrons au paragraphe suivant qu'il est cependant d'une importance théorique capitale.

En 1984, Karmarkar [225], des laboratoires AT&T, a publié un nouvel algorithme, de complexité $\mathcal{O}(n^{3.5} L^2)$. Il s'agit d'une méthode **de points intérieurs** : contrairement au simplexe qui se déplace de sommet en sommet sur la « bordure » du polytope, la méthode de Karmarkar « plonge » à l'intérieur pour aller au plus vite vers le sommet optimal.

Les solveurs disponibles actuellement sont extrêmement puissants et disposent généralement de nombreuses méthodes d'optimisation (prétraitement, coupes, génération de colonnes...) dont la description dépasse le cadre de cette thèse.

1.3.3 Approche polyédrale pour la résolution d'un problème d'optimisation combinatoire

Considérons le segment L joignant deux points u et v de \mathbb{R}^n . Pour tout vecteur $c \in \mathbb{R}^n$ donné, le problème

$$\max\{c^T x \mid x \in L\}$$

est trivial à résoudre : soit u soit v est une solution optimale. En effet, supposons par exemple que $c^T u \geq c^T v$ et soit $x \in L$. Comme x appartient à $\text{conv}(\{u, v\})$, il existe $0 \leq \lambda \leq 1$ tel que $x = \lambda u + (1 - \lambda)v$. Par conséquent,

$$c^T x = \lambda c^T u + (1 - \lambda)c^T v \leq \lambda c^T u + (1 - \lambda)c^T u = c^T u$$

16. Cet algorithme fut largement employé durant le blocus de Berlin afin d'optimiser le ravitaillement par pont aérien. La revue *Computing in Science and Engineering* l'a cité parmi les dix algorithmes les plus importants du XX^e siècle. Le mot *simplexe* désigne la généralisation d'un triangle et d'un tétraèdre aux dimensions supérieures, et aurait été suggéré par Motzkin (cf. http://en.wikipedia.org/wiki/Simplex_algorithm).

et donc u est une solution optimale. Ceci se généralise à un nombre quelconque de points : si x est combinaison convexe de points v_1, \dots, v_k alors

$$c^T x \leq \max\{c^T v_i\} (i \in 1, \dots, k)$$

Autrement dit :

Proposition 1.8 *Étant donné $S \subseteq \mathbb{R}^n$ un ensemble fini et $c \in \mathbb{R}^n$,*

$$\max\{c^T x \mid x \in S\} = \max\{c^T x \mid x \in \text{conv}(S)\}$$

A première vue, cette proposition peut sembler de peu d'intérêt puisque l'on remplace un problème sur un ensemble fini (S) par un problème sur un ensemble *infini* ($\text{conv}(S)$). Mais le polytope $\text{conv}(S)$ est aussi un polyèdre, et en tant que tel on sait qu'il peut être caractérisé par un système linéaire. **Si on arrive à donner cette caractérisation, on se ramène alors à la résolution d'un programme linéaire.**

Cette approche, dite *approche polyédrale*, a été initiée par Edmonds [120] dans le cadre de ses travaux sur le problème du couplage maximum (cf. section 2.1.2 pour les définitions concernant les couplages). Dans un graphe $G = (V, E)$, pour chaque couplage M on note $v_M \in \{0, 1\}^E$ son *vecteur caractéristique* ou *vecteur d'incidence* :

$$v_M(e) = \begin{cases} 1 & \text{si } e \in M \\ 0 & \text{si } e \notin M \end{cases}$$

pour chaque arête $e \in E$. L'ensemble $\text{conv}(v_M \mid M \text{ est un couplage})$ est un polytope de \mathbb{R}^n appelé **polytope des couplages** de G , et c'est précisément de ce polytope dont Edmonds a donné la première description complète sous forme d'inégalités linéaires.

Cependant, il ne suffit pas de connaître une telle caractérisation pour en déduire un algorithme polynomial. En effet, le programme linéaire obtenu est susceptible de contenir un nombre exponentiel de contraintes (c'est d'ailleurs le cas pour le polytope des couplages), et il n'est donc *a priori* pas possible de le résoudre tel quel en temps polynomial. Cependant, une description partielle du polyèdre peut s'avérer suffisante pour y parvenir. Grötschel, Lovász et Schrijver ont effectivement démontré en 1981 [178] qu'avec la méthode de l'ellipsoïde, l'optimisation sur un polyèdre ne dépend pas du nombre de contraintes du problème, mais du **problème de séparation** associé.

Définition 1.44 *Le problème de séparation associé à un polyèdre P de dimension n consiste à vérifier, pour tout $x \in \mathbb{R}^n$, si $x \in P$ et sinon, à déterminer un hyperplan séparant x de P , c'est-à-dire une inégalité vérifiée par tout point de P mais non satisfaite par x .*

Ce problème n'est pas du tout trivial ; il est même \mathcal{NP} -difficile dans de nombreux cas. Mais lorsque l'on peut le résoudre en temps polynomial, il s'en suit que l'optimisation sur le polyèdre peut elle-même être effectuée en temps polynomial. C'est le **théorème de séparation / optimisation** :

Théorème 1.6 (Grötschel, Lovász et Schrijver (1981)) *On peut optimiser sur un polyèdre en temps polynomial si et seulement si on peut résoudre le problème de séparation en temps polynomial sur ce polyèdre.*

Nous verrons à la section 2.1.4 que ce théorème a des conséquences fondamentales sur la résolution de certains problèmes d'optimisation dans la classe des graphes parfaits.

CONCLUSION DU CHAPITRE

Dans ces quelques pages, nous avons redonné les notions de base de la théorie des graphes, de la théorie de la complexité et de la théorie polyédrale. Nous disposons à présent de tous les outils nécessaires pour aborder sereinement les prochains chapitres ; en particulier, le chapitre suivant traite en profondeur du problème au cœur de cette thèse : le problème du stable maximum dans les graphes sans P_5 .

PROBLÈME DU STABLE MAXIMUM : TECHNIQUES DE RÉSOLUTION ET APPLICATIONS AUX GRAPHES SANS P_5

SOMMAIRE

2.1	LE PROBLÈME DU STABLE MAXIMUM	37
2.1.1	Énoncé et applications	37
2.1.2	Problèmes connexes	38
2.1.3	Complexité	42
2.1.4	Quelques exemples de classes MIS-faciles	44
2.1.5	Théorèmes d'Alekseev	49
2.2	LA CLASSE DES GRAPHES SANS P_5	52
2.2.1	Propriétés des graphes sans P_5	52
2.2.2	Trouver une clique dominante dans un graphe sans P_5	54
2.2.3	Propriétés des graphes sans P_5 bipartis	57
2.3	LES GRAPHES AUGMENTANTS	59
2.3.1	Notion de chaîne S -augmentante	59
2.3.2	Méthode des graphes augmentants	60
2.3.3	Caractérisation des graphes augmentants	63
2.3.4	Comment trouver un graphe augmentant en temps polynomial ?	64
2.3.5	Application aux graphes sans P_5	65
2.4	LA DÉCOMPOSITION MODULAIRE	72
2.4.1	Premières définitions et propriétés	72
2.4.2	Théorème de décomposition modulaire	75
2.4.3	Arbre de décomposition modulaire	75
2.4.4	Application aux cographes	77
2.4.5	Graphes sans P_5 de largeur de clique bornée	80
2.4.6	Application aux graphes sans P_5	82
2.5	LA STRUCTION	86
2.5.1	Fonctions pseudo-booléennes, posiformes et graphe de conflit	86

2.5.2	La struction	89
2.5.3	Construction directe	90
2.5.4	Complexité	91
2.5.5	D'autres types de transformations	91
2.5.6	Application aux graphes sans P_5	96
2.6	LA DÉCOMPOSITION PAR CLIQUES D'ARTICULATION	98
2.6.1	Présentation	98
2.6.2	Calcul d'un stable de poids maximum	100
2.6.3	Application aux graphes sans P_5	101
2.7	GÉNÉALOGIE DES GRAPHES SANS P_5	102
	CONCLUSION	107

LE problème consistant à déterminer un stable de cardinalité maximale dans un graphe est un problème classique d'optimisation combinatoire, ayant de très nombreuses applications pratiques (il intervient par exemple en vision par ordinateur, en biologie moléculaire, ou encore dans des problématiques d'ordonnement).

Dans la première partie de ce chapitre, nous commençons par rappeler les deux formes du problème (décision et optimisation), et nous redonnons la démonstration de sa \mathcal{NP} -difficulté ; nous exposerons également un certain nombre de problèmes connexes. La seconde partie est consacrée à la présentation de la classe des graphes sans P_5 . Nous justifierons l'étude de cette classe au regard du problème du stable maximum et rappellerons certaines propriétés de ces graphes, notamment en terme de domination ; nous insisterons plus particulièrement sur les graphes sans P_5 bipartis, qui sont au cœur du problème étudié au chapitre 4, et dont les intéressantes propriétés interviendront à de nombreuses reprises dans la résolution du problème du stable maximum dans les graphes sans P_5 k -colorables présenté au chapitre 3. Le caractère \mathcal{NP} -difficile du problème du stable maximum a amené les chercheurs à développer un certain nombre de techniques pour le résoudre efficacement dans les classes de graphes pour lesquelles ceci est possible. Nous présentons les quatre principales : la technique des graphes augmentants, la décomposition modulaire, la struction et la décomposition par clique-séparateurs ; pour chacune d'elles nous donnons un état de l'art des algorithmes polynomiaux qu'elle a permis d'obtenir sur des sous-classes de graphes sans P_5 .

2.1 LE PROBLÈME DU STABLE MAXIMUM

2.1.1 Énoncé et applications

Nous avons vu au chapitre précédent qu'un ensemble de sommets est stable s'ils sont deux à deux non adjacents. Il est très facile de trouver un ensemble stable S dans un graphe G : il suffit de choisir arbitrairement un premier sommet v que l'on ajoute à S , de supprimer dans G v ainsi que tous ses voisins et de recommencer sur le sous-graphe ainsi obtenu. En exécutant cet algorithme glouton (dont on montre facilement qu'il s'exécute en temps $O(m)$) jusqu'à ce que le graphe ne contienne plus de sommets, on obtient un stable **maximal**, c'est-à-dire qui n'est contenu dans aucun autre ensemble stable :

Algorithme 2: Stable maximal

Entrées : Un graphe $G = (V, E)$
Sorties : Un stable maximal S de G

$W \leftarrow V;$
 $S \leftarrow \emptyset;$
tant que $W \neq \emptyset$ **faire**
 | choisir un sommet w de W ;
 | $S \leftarrow (S \cup \{w\});$
 | $W \leftarrow (W \setminus N[w]);$
fin

Puisque l'on sait calculer facilement un stable, une question vient assez naturellement : étant donné un graphe G et un entier positif k , peut-on déterminer si G contient un stable de cardinalité au moins k ? Comme nous l'avons vu au chapitre précédent, il s'agit là d'un problème de décision, auquel la réponse est soit « oui » soit « non ». Le problème d'optimisation correspondant, qui consiste à trouver dans un graphe un stable de **cardinalité** maximum, est appelé **problème du stable maximum**¹ (**Maximum Independent Set Problem**²), que nous abrègerons souvent par « MIS » dans la suite :

PROBLÈME DU STABLE MAXIMUM (MIS)
(MAXIMUM INDEPENDENT SET (GT20))

INSTANCE : Un graphe $G = (V, E)$
OBJECTIF : Déterminer un stable maximum de G

Rappelons que la cardinalité d'un stable maximum de G est notée $\alpha(G)$. Certains auteurs définissent MIS sous une forme plus faible, comme le problème consistant à calculer seulement $\alpha(G)$. Bien que les deux problèmes soient de même difficulté, nous préférons la formulation donnée

1. Le français est assez flou sur l'usage des adjectifs *maximal* et *maximum*, ce dernier étant d'ailleurs renseigné d'abord comme un substantif dans les dictionnaires. Dans toute la suite de cette thèse, l'adjectif *maximal* (resp. *minimal*) sera employé pour signifier *maximal* (resp. *minimal*) par inclusion ; *maximum* (resp. *minimum*) signifiera de cardinalité maximum (resp. minimum).

2. On rencontre aussi parfois *Vertex Packing* mais cet usage semble s'atténuer ; certains le réservent à la version pondérée du problème.

ci-dessus, porteuse de plus d'informations.

Ce problème a de nombreuses applications pratiques [34] : il intervient en théorie de l'information (cf. section A.1.5), en vision par ordinateur, dans l'allocation des registres d'un processeur, en biologie moléculaire, dans l'attribution des fréquences radio ou encore dans les problèmes de planification. Plus généralement, le problème du stable maximum apparaît dès que l'on cherche à utiliser au mieux un ensemble de ressources sans créer de conflit entre elles.

Exemple 2.1 *Monsieur Hulot se rend pour une semaine dans un centre de vacances proposant diverses activités de loisir et sportives. Afin de satisfaire l'ensemble de la clientèle, certaines activités ont lieu en même temps, et il n'est donc pas possible de participer à toutes. Soucieux de rentabiliser son séjour, Monsieur Hulot désire participer à un maximum d'activités.*

La théorie des graphes est un excellent moyen de modéliser cette situation : on associe à chaque activité le sommet d'un graphe, et on relie deux sommets si les activités correspondantes sont « incompatibles », c'est-à-dire s'il est impossible de participer aux deux, pour des contraintes d'horaires. Il est alors facile de voir qu'un ensemble d'activités « compatibles » constitue un stable. Ainsi, le problème de Monsieur Hulot se ramène à la recherche d'un stable maximum dans un graphe.

2.1.2 Problèmes connexes

Il existe un certain nombre de problèmes d'optimisation étroitement liés au problème du stable maximum :

Le problème de la clique maximum

Il s'agit ici de déterminer une clique de plus grande taille dans un graphe G :

PROBLÈME DE LA CLIQUE MAXIMUM (MAXIMUM CLIQUE (GT19))
INSTANCE : Un graphe $G = (V, E)$
OBJECTIF : Déterminer une clique maximum de G

Évidemment, comme une clique dans un graphe G est un stable dans son complémentaire \overline{G} et réciproquement, ce problème revient à chercher un stable maximum dans \overline{G} . Rappelons que la cardinalité d'une clique maximum de G est notée $\omega(G)$, et qu'on a donc $\omega(G) = \alpha(\overline{G})$.

Exemple 2.2 *Plutôt que d'établir le graphe d'incompatibilité des activités comme ci-dessus, Monsieur Hulot peut préférer le graphe de « compatibilité » des activités : autrement dit, il choisit de relier deux sommets si et seulement s'ils correspondent à deux activités n'ayant pas lieu en même temps. Son problème se ramène alors à celui de la clique maximum.*

Le problème du stable de poids maximum

Il s'agit d'une généralisation du problème du stable maximum ; chaque sommet du graphe possède un *poids*, et l'on cherche le stable de poids maximum, noté $\alpha_w(G)$ (le poids d'un stable étant égal à la somme des poids des sommets le composant) :

PROBLÈME DU STABLE DE POIDS MAXIMUM (MAXIMUM WEIGHT INDEPENDENT SET)
INSTANCE : Un graphe $G = (V, E)$ et une fonction $f : V \rightarrow \mathbb{R}$
OBJECTIF : Déterminer un stable de poids maximum dans G

Naturellement, ce problème est au moins aussi difficile que celui du stable maximum. Il nous arrivera fréquemment de le désigner par MWIS.

Exemple 2.3 *Supposons à présent que Monsieur Hulot émette des préférences sur certaines activités ; ceci se traduit sur le graphe en affectant des poids aux sommets. La solution au problème correspond alors à un stable de poids maximum.*

Le problème du dominant stable minimum

Un **(ensemble) dominant** (*dominating set*) est un sous-ensemble $D \subseteq V$ tel que tout sommet de $V \setminus D$ ait au moins un voisin dans D . Par exemple, V est un dominant (trivial) ; aussi le problème consiste-t-il à déterminer un plus petit dominant dans un graphe :

PROBLÈME DU DOMINANT MINIMUM (MINIMUM DOMINATING SET (GT2))
INSTANCE : Un graphe $G = (V, E)$
OBJECTIF : Déterminer un dominant minimum de G

On désigne par $\gamma(G)$ la taille d'un dominant minimum, appelée **nombre de domination** (*domination number*) de G . Le lien avec le problème du stable maximum est illustré par les propositions suivantes :

Proposition 2.1 *Un stable S est maximal si et seulement s'il est dominant.*

Démonstration. Soit S un stable.

\Rightarrow Si S n'est pas dominant, il existe un sommet v qui n'a aucun voisin dans S ; donc $S \cup \{v\}$ est un stable, et S n'est donc pas maximal.

\Leftarrow Si S n'est pas maximal, il existe un sommet v tel que $S \cup \{v\}$ est stable ; donc v n'a aucun voisin dans S , et S n'est donc pas dominant. □

Par conséquent, tout stable maximal est aussi dominant. On peut même montrer que :

Proposition 2.2 *Tout stable maximal est un dominant minimal.*

Démonstration. Soit S un stable maximal. D'après la proposition précédente, S est un ensemble dominant. Supposons qu'il ne soit pas dominant minimal; il existe donc $v \in S$ tel que $S \setminus \{v\}$ est également dominant. En particulier, v est dominé par $S \setminus \{v\}$, donc adjacent à un sommet de $S \setminus \{v\}$, ce qui contredit l'hypothèse que S est stable. Par conséquent, S est bien dominant minimal. \square

Bien entendu, la réciproque n'est pas vraie puisqu'un dominant n'est pas forcément stable (par exemple, dans un P_4 , les deux sommets centraux constituent un dominant minimal mais pas un stable).

On note à présent $i(G)$ la cardinalité **minimum** d'un stable maximal. D'après ce qui précède, on a :

Proposition 2.3 Pour tout graphe G , $\gamma(G) \leq i(G) \leq \alpha(G)$.

Ainsi, alors que le problème du stable maximum consiste à déterminer le plus grand des stables maximaux, le problème du dominant stable minimum consiste à trouver le plus petit (ce qui explique qu'il soit aussi connu dans la littérature comme **MINIMUM MAXIMAL INDEPENDENT SET**), que nous noterons **MIDS** (ou **MWIDS** dans sa version pondérée) :

PROBLÈME DU DOMINANT STABLE MINIMUM
(MINIMUM INDEPENDENT DOMINATING SET)
INSTANCE : Un graphe $G = (V, E)$
OBJECTIF : Déterminer un dominant stable minimum de G

Pour plus de précisions sur la domination dans les graphes, on pourra se reporter à [193] et [194].

Le problème du transversal minimum

Un **transversal** (*vertex cover*) est un ensemble de sommets contenant au moins une extrémité de chaque arête. Une nouvelle fois, V est un exemple trivial de transversal. Le problème consistant à trouver un transversal minimum est un problème d'optimisation classique, apparaissant dans les 21 problèmes de Karp [226]³ :

PROBLÈME DU TRANSVERSAL MINIMUM
(VERTEX COVER (GT1))
INSTANCE : Un graphe $G = (V, E)$
OBJECTIF : Déterminer un transversal minimum de G

On désigne par $\tau(G)$ la cardinalité d'un transversal minimum de G . Par définition, si $T \subseteq V$ est un transversal, $V \setminus T$ est un stable, et réciproquement. On en déduit immédiatement la relation suivante :

3. Les 21 problèmes de Karp sont une liste de problèmes classiques de combinatoire et de théorie des graphes pour lesquels Richard Karp a montré en 1972 qu'ils sont tous \mathcal{NP} -complets.

Théorème 2.1 (Gallai (1959) [146]) *Pour tout graphe G d'ordre n ,*

$$\alpha(G) + \tau(G) = n.$$

Ainsi, si l'on est en mesure de déterminer un transversal minimum de G , on obtient simultanément un stable maximum de ce graphe.

Le problème du couplage maximum

Un **couplage** (*matching*) est un ensemble d'arêtes deux à deux non adjacentes ; autrement dit, un couplage est aux arêtes d'un graphe ce qu'un stable est aux sommets. Il s'agit d'une des notions fondamentales en théorie des graphes, à mettre en relation avec d'autres concepts théoriques, en particulier ceux de flot et de facteur (un couplage parfait (voir ci-dessous) n'est rien d'autre qu'un 1-facteur (cf. Annexe)) ; et qui présente également un intérêt pratique si l'on considère le problème classique de l'affectation optimale ou l'identification de certains composés en chimie. D'un point de vue combinatoire, le problème consiste à trouver un couplage maximum :

PROBLÈME DU COUPLAGE MAXIMUM

(MAXIMUM MATCHING)

INSTANCE : Un graphe $G = (V, E)$

OBJECTIF : Déterminer un couplage maximum de G

Les sommets incidents à une arête du couplage sont dits **saturés**, et les autres **insaturés**. Un couplage est **parfait** (*perfect matching*) si tous les sommets sont saturés. On note généralement $\nu(G)$ la taille d'un couplage maximum de G . Par définition du graphe adjoint $L(G)$, un couplage dans G est un stable dans $L(G)$, et on a donc l'égalité suivante :

$$\nu(G) = \alpha(L(G)). \quad (2.1)$$

De plus, comme un sommet ne peut couvrir plus d'une arête d'un couplage, on a la relation de dualité faible suivante :

$$\tau(G) \geq \nu(G). \quad (2.2)$$

Cette inégalité peut être stricte (il suffit de considérer le triangle K_3), mais König [233] a démontré en 1931 qu'on a toujours l'égalité quand on considère des graphes bipartis. Ce classique résultat *min-max*, fut généralisé indépendamment la même année par Egerváry [123] aux graphes pondérés, si bien qu'il est désormais d'usage de l'appeler *théorème de König-Egerváry* :

Théorème 2.2 (König (1931), Egerváry (1931)) *Pour tout graphe biparti, la cardinalité d'un couplage maximum est égale à la cardinalité d'un transversal minimum :*

$$\nu(G) = \tau(G).$$

En fait, le théorème de König-Egerváry est équivalent à plusieurs autres résultats importants en combinatoire, parmi lesquels le théorème de Frobenius [141], (exprimé à l'origine en termes de matrices), et le théorème de Hall [183] (exprimé en termes de représentants distincts d'un ensemble), qui tous deux caractérisent l'existence d'un couplage parfait dans un graphe biparti :

Théorème 2.3 (Frobenius (1917)) *Un graphe biparti $G = (V, E)$ a un couplage parfait si et seulement si tout transversal est de cardinalité au moins $\frac{1}{2}|V|$.*

Théorème 2.4 (Hall (1935)) *Un graphe biparti $G = (V, E)$, où $V = S \cup T$ (et $|S| = |T|$), a un couplage parfait si et seulement si $|X| \leq |N(X)|$ pour tout $X \subseteq S$.*

Généralisation de la notion de stable

Il existe plusieurs généralisations de la notion d'ensemble stable, dont nous donnons une très brève présentation.

Considérons par exemple un ensemble de sommets de degré maximum k . Lorsque $k = 0$, nous retrouvons la notion de stable. Lorsque $k = 1$, on parle d'**ensemble de dissociation** (*dissociation set*). Yannakakis [349] a prouvé qu'il est \mathcal{NP} -difficile de trouver un ensemble de dissociation maximum dans un graphe biparti. Il a été montré récemment que le problème reste difficile dans les graphes adjoints planaires et dans les graphes planaires bipartis, et qu'il est également difficile de trouver un ensemble de dissociation maximal de taille minimum dans un graphe quelconque [292].

Plus récemment, Jagota, Narasimhan et Šoltés [219] ont défini pour tout entier k positif la notion d'**ensemble k -isolé** (*k -insulated set*) comme un ensemble de sommets S tel que tout sommet de S a au plus k voisins dans S et tout sommet de $V \setminus S$ a au moins $k + 1$ voisins dans S . Les ensembles 0-isolés sont donc les stables maximaux.

2.1.3 Complexité

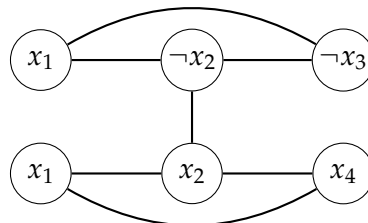
Revenons à présent à notre question initiale : peut-on calculer efficacement un stable maximum dans un graphe ? Un algorithme simple, puisqu'un stable maximum est aussi maximal, consiste à énumérer tous les stables maximaux du graphe et à en retenir un qui soit de cardinalité maximum. Le problème est qu'on est alors confronté à un phénomène d'explosion combinatoire, car un graphe peut avoir un nombre exponentiel (en fonction de son ordre) de stables maximaux : Moon et Moser [280] ont en effet montré qu'un graphe d'ordre n peut contenir jusqu'à $3^{n/3}$ cliques maximales, et par conséquent (en passant au complémentaire), qu'un graphe d'ordre n peut contenir jusqu'à $3^{n/3}$ stables maximaux. Il est d'ailleurs

facile de construire un graphe atteignant cette borne : il suffit de considérer l'union disjointe de $\frac{n}{3}$ cliques d'ordre 3 (des *triangles*). De plus il est possible de lister tous les stables maximaux en temps $O(3^{n/3})$, en utilisant par exemple l'algorithme de Bron-Kerbosch [56], et par conséquent, on peut calculer un stable maximum en temps $O(3^{n/3}) \approx O(1.4422^n)$. Ce problème a bénéficié d'une large attention dans la littérature, et d'autres algorithmes ont été proposés, voir par exemple Johnson et al. [222] et Lawler et al. [240]; l'algorithme récursif de Tarjan et Trojanowski [328] permet de détecter quand certains appels conduisent à une moins bonne solution que la solution courante, ce qui permet de ramener la complexité à $O(2^{n/3}) \approx O(1.2599^n)$. Ce résultat a ensuite été plusieurs fois amélioré, et l'algorithme le plus efficace connu aujourd'hui est celui de Robson [305], qui s'exécute en $O(2^{0.249n}) \approx O(1.1888^n)$, une complexité qui reste donc exponentielle.

En fait, si, comme tout porte à le croire, $P \neq \mathcal{NP}$, on ne pourra jamais trouver d'algorithme polynomial pour résoudre le problème du stable maximum dans un graphe général, car l'on sait depuis longtemps que ce problème est \mathcal{NP} -difficile (le problème équivalent de la **clique maximum** fait partie des 21 problèmes de Karp évoqués précédemment). Nous redonnons ici la démonstration de ce résultat :

Théorème 2.5 (Karp (1972) [226]) *Le problème du stable maximum est \mathcal{NP} -difficile.*

Démonstration. Pour montrer que *MIS* est \mathcal{NP} -difficile, il suffit de prouver que le problème de décision associé est \mathcal{NP} -complet. Un moyen simple d'y parvenir est d'effectuer une réduction à partir du problème 3-SAT : on associe à chaque littéral de la formule booléenne un sommet d'un graphe (en le dupliquant éventuellement autant de fois qu'il apparaît dans ladite formule) et on relie ensuite entre deux sommets s'ils correspondent respectivement à un littéral et à sa négation, ou s'ils apparaissent dans la même clause. Par exemple, la figure suivante donne le graphe associé à la formule $(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee x_2 \vee x_4)$:



Le graphe ainsi construit contient un ensemble stable de taille au moins k (k étant le nombre de clauses) si et seulement si la formule booléenne est satisfaisable. En effet :

⇐ si l'on connaît une affectation de valeurs aux variables telle que la formule est vraie, on choisit dans chaque clause un littéral rendu vrai par cette affectation (et on peut le faire, puisque la formule est vraie).

L'ensemble constitué des sommets correspondants dans le graphe est bien stable (puisque'il ne contient qu'un sommet de chaque clause et qu'aucune affectation ne rend vrais à la fois un littéral et sa négation) et est de taille k car la formule contient k clauses.

⇒ réciproquement, supposons que nous ayons un stable S de taille k ou plus. Pour les raisons que l'on vient d'évoquer, il ne peut contenir deux sommets correspondant à des littéraux opposés ou appartenant à une même clause, et comme la formule contient k clauses, chaque clause doit contenir un littéral correspondant à un sommet de S . Par conséquent, en donnant la valeur *vrai* aux littéraux correspondant aux sommets de S , la formule est satisfaite.

Il est facile de voir que cette réduction peut être effectuée en temps polynomial. Par conséquent, le problème du stable maximum est \mathcal{NP} -difficile. \square

Corollaire 2.1 *Les problèmes de la clique maximum, du stable de poids maximum, du dominant stable minimum et du transversal minimum sont eux aussi \mathcal{NP} -difficiles.*

Notons que le problème du dominant stable minimum est plus difficile que celui du stable maximum, dans le sens où il est \mathcal{NP} -difficile même dans les graphes bipartis [116], classe dans laquelle MIS peut être résolu en temps polynomial. Signalons également que MIS reste \mathcal{NP} -difficile dans certaines classes soumises à de fortes restrictions, comme les graphes sans triangle [298] (voir [52] pour une preuve de l'argument utilisé) ou les graphes planaires [151, 152], même de degré maximum 3 [149], et même cubiques [269]. De plus, Håstad [192] a montré qu'à moins que $\mathcal{P} = \mathcal{NP}$, il n'est pas possible d'approximer la valeur de $\alpha(G)$ à un facteur $n^{1-\epsilon}$ près.

Notation : afin d'éviter les phrases trop lourdes, nous dirons qu'une classe de graphes est Π -facile par rapport à un problème Π s'il existe un algorithme polynomial permettant de résoudre Π dans cette classe ; et nous dirons qu'elle est Π -difficile si Π est \mathcal{NP} -difficile dans cette classe.

En particulier, nous rencontrerons à de très nombreuses reprises les expressions *MIS-facile* et *MIS-difficile*. La section suivante présente d'ailleurs quelques exemples de classes MIS-faciles.

2.1.4 Quelques exemples de classes MIS-faciles

Malgré la \mathcal{NP} -difficulté du problème dans le cas général et dans les cas évoqués ci-dessus, il existe de nombreuses classes de graphes pour lesquelles on sait trouver un stable maximum de manière exacte en temps polynomial.

Graphes ayant un nombre polynomialement borné de stables maximaux

Ainsi que nous l'avons vu au paragraphe précédent, un graphe G d'ordre n peut avoir un nombre élevé de stables maximaux (jusqu'à $3^{n/3}$). Mais ce n'est pas le cas de tous les graphes ; il est en effet possible de montrer que certains n'en contiennent qu'un nombre polynomial. C'est par exemple le cas des graphes sans $2K_2$ (l'union disjointe de deux K_2), qui, comme l'a prouvé Farber [130], possèdent au plus $\binom{n}{2}$ stables maximaux (notons au passage que puisqu'un P_5 contient un $2K_2$, la classe des graphes sans $2K_2$ est incluse dans la classe des graphes sans P_5 ; nous y reviendrons plus longuement dans la suite).

Or Tsukiyama, Ide, Ariyoshi et Shirakawa [333] ont montré qu'il est possible de lister tous les stables maximaux d'un graphe en un temps polynomial, $O(mn)$, **par stable généré**⁴. Chiba et Nishizeki [75] ont amélioré ce résultat, en prouvant que l'on peut faire la même chose en temps $O(ma(G))$ (où $a(G)$ désigne l'**arboricité** (*arboricity*) de G , c'est-à-dire le nombre minimal de forêts nécessaires pour couvrir les arêtes de G), et plus récemment, Makino et Uno [270] ont fourni un autre algorithme, de complexité $O(M(n))$ par stable (où $M(n)$ désigne le temps nécessaire pour multiplier deux matrices carrées de taille $n \times n$, soit actuellement $O(n^{2,376})$, la complexité de l'algorithme de Coppersmith-Winograd [96]).

En appliquant un de ces algorithmes sur une classe de graphes ayant un nombre de stables maximaux polynomialement borné, on obtient immédiatement un algorithme polynomial permettant de résoudre le problème du stable maximum pour cette classe. Un tel algorithme, dont le temps d'exécution dépend non seulement de la taille de l'entrée mais également de la taille de la sortie, est qualifié d'*output-sensitive* par les anglophones.

Graphes bipartis

Les résultats énoncés dans la section 2.1.2 montrent que dans un graphe biparti, le problème du stable maximum se ramène à celui du couplage maximum, puisqu'on a alors

$$\nu(G) + \alpha(G) = n.$$

L'obtention d'un couplage dans un biparti a été décrite par Kőnig dans la démonstration du théorème de Kőnig-Egerváry (théorème 2.2) et repose sur le concept de **chaîne augmentante**, introduit 40 ans plus tôt par Petersen [296] :

Définition 2.1 (Chaîne augmentante) *Etant donné un couplage M d'un graphe $G = (V, E)$,*

- une **chaîne M -alternée** (alternating path) *est une chaîne (non nécessairement induite) dont les arêtes appartiennent alternativement à M et à $E \setminus M$;*
- une **chaîne M -augmentante** (augmenting path) *est une chaîne alternée dont les extrémités sont M -insaturées.*

4. En fait, leur algorithme est une variante d'un « vieil » algorithme de 1959 développé par Paull et Unger [295], qui génère tous les stables maximaux en temps $\mathcal{O}(n^2)$ par stable généré, mais qui construit des structures de données de taille exponentielle.

Le qualificatif *augmentante* vient ici de ce qu'il suffit d'inverser les arêtes de M et celles de $E \setminus M$ d'une telle chaîne pour obtenir un couplage de cardinalité supérieure. Petersen avait déjà observé que cette condition est également nécessaire, un résultat redémontré et reformulé de façon plus moderne en 1957 par Berge [23], ce qui explique qu'on le rencontre parfois sous le nom de *lemme de Berge* :

Lemme 2.1 (Petersen (1891), König (1931), Berge (1957), Norman et Rabin [290] (1959)) *Un couplage est maximum si et seulement s'il n'existe aucune chaîne augmentante.*

La méthode suivante, permettant de trouver une chaîne augmentante dans un graphe biparti, est essentiellement due à König [233] :

Algorithme 3: Algorithme du couplage maximum dans un graphe biparti

Entrées : Un graphe biparti $G = (V, E)$ et un couplage M

Sorties : Un couplage M' tel que $|M'| > |M|$ (s'il en existe un)

Soient A et B les classes de la bipartition de V . Orienter chaque arête $e = \{a, b\}$ (où $a \in A, b \in B$) de la manière suivante :

- si $e \in M$, orienter e de b vers a ;
- si $e \notin M$, orienter e de a vers b ;

Soient A_M et B_M les ensembles de sommets M -insaturés de A et B respectivement. Alors une chaîne M -augmentante (s'il en existe une) peut être trouvée en cherchant une chaîne orientée de A_M à B_M .

Comme on peut trouver une chaîne orientée de A_M à B_M en temps $\mathcal{O}(m)$, l'algorithme précédent permet de trouver un couplage maximum dans un graphe biparti en temps $\mathcal{O}(nm)$. Il est même possible d'y parvenir en temps $\mathcal{O}(\sqrt{nm})$, ainsi que l'ont montré Hopcroft et Karp [215, 216] (le facteur \sqrt{n} pouvant même être réduit à $\sqrt{v(G)}$ [311, Théorème 16.5]).

Dans un graphe biparti pondéré, la recherche d'un stable de poids maximum peut être menée à bien en utilisant un algorithme de flot maximum, par exemple l'algorithme d'Edmonds-Karp (de complexité $\mathcal{O}(nm^2)$) [122], surpassé depuis par des algorithmes de complexité moindre que $\mathcal{O}(n^3)$ (cf. [311]).

Théorème 2.6 *Dans un graphe biparti, on peut calculer un stable maximum en temps $\mathcal{O}(\sqrt{nm})$ et un stable de poids maximum en temps $\mathcal{O}(n^3)$.*

Graphes adjoints

Bien qu'il soit simple de détecter une chaîne augmentante dans un graphe biparti, le problème est beaucoup moins trivial dans un graphe général. Puisque ce sont alors les cycles impairs qui posent problème, Jack Edmonds [121] eut le premier l'idée, en 1965, de *contracter* certains de ces cycles impairs, qu'il appelle *blossoms* ; il réussit ainsi à mettre au point un algorithme polynomial permettant de trouver les chaînes augmentantes

dans un graphe quelconque, et donc de calculer un couplage maximum en temps polynomial dans n'importe quel graphe⁵. Ceci implique l'existence d'un algorithme polynomial pour le problème du stable maximum dans la classe des graphes adjoints; en effet, comme nous l'avons vu précédemment, un stable dans un graphe adjoint $L(G)$ correspond à un couplage dans le graphe original G . Or, Roussopoulos [307] et Lehot [243] ont montré de manière indépendante qu'il est possible de reconstruire un graphe à partir de son graphe adjoint en temps linéaire. En appliquant l'algorithme d'Edmonds au graphe G ainsi reconstruit, on obtient ainsi simultanément un stable maximum de $L(G)$. D'abord restreint aux graphes non pondérés, Edmonds a rapidement étendu son algorithme aux graphes pondérés [120]; dans les deux cas, les algorithmes s'exécutent en temps $\mathcal{O}(n^2m)$. En ce qui concerne la version non pondérée, un algorithme en temps $\mathcal{O}(n^3)$ a été publié en 1969 par Balinski [20], un autre en $\mathcal{O}(n^{5/2})$ par Even et Kariv [128] en 1975; un algorithme en $\mathcal{O}(\sqrt{nm})$ fut annoncé par Micali et Vazirani en 1980 [277], mais avec une démonstration partielle; une preuve complète fut donnée par la suite par Blum [29], Vazirani [338, 339] et Gabow et Tarjan [145]. Un algorithme en $\mathcal{O}(n^3)$ pour le couplage pondéré a également été développé par Gabow [143, 144]; un algorithme de même complexité est donné dans les ouvrages de Lawler [239] et de Schrijver [311]. Ces complexités ont été plusieurs fois améliorées depuis, et nous invitons le lecteur à se reporter à cette dernière référence pour plus de précisions sur les algorithmes de couplages.

Théorème 2.7 *Dans un graphe adjoint, on peut trouver un stable de cardinalité maximum en temps $\mathcal{O}(\sqrt{nm})$ et un stable de poids maximum en temps $\mathcal{O}(n^3)$.*

Graphes sans griffe

Ainsi que cela a été signalé au chapitre précédent, la classe des graphes adjoints est généralisée par celle des graphes sans griffe. C'était la motivation initiale pour l'étude de cette classe, et au moins deux autres découvertes dans les années 70 ont contribué à renforcer l'attention des chercheurs sur ces graphes : d'une part, Las Vergnas [238] et Sumner [322] ont montré que tout graphe sans griffe d'ordre pair admet un couplage parfait; et d'autre part, Parthasarathy et Ravindra [294] ont prouvé que la conjecture forte des graphes parfaits était vérifiée pour les graphes de Berge sans griffe.

En 1980, Minty [279] et Sbihi [309, 310] ont, indépendamment, montré que l'on peut calculer un stable maximum en temps polynomial dans la classe des graphes sans griffe. L'algorithme de Sbihi est une extension de la technique de contraction des blossoms utilisée dans l'algorithme d'Edmonds pour trouver un couplage maximum; l'algorithme de Minty réduit le problème à celui du couplage maximum dans un graphe auxiliaire appelé **graphe d'Edmonds** (l'un pouvant être obtenu à partir de l'autre en

⁵. Selon Lovász et Plummer [251], l'algorithme d'Edmonds est l'un des algorithmes combinatoires les plus utilisés.

temps polynomial). Dans les deux cas, le principe consiste à déterminer des chaînes augmentantes *de sommets*, une idée qui sera par la suite étendue au concept de **graphe augmentant**. Notons qu'alors que l'algorithme de Sbihi (de complexité $\mathcal{O}(n^3)$) se limite au cas de graphes non pondérés, celui de Minty (de complexité $\mathcal{O}(n^7)$ ⁶) permet de calculer un stable de poids maximum (bien que la version originale contienne une erreur qui n'a été corrigée qu'en 2001 par Nakamura et Tamura [289]). Quelques années plus tard, Lovász et Plummer [251] ont proposé un algorithme consistant en une réduction plus immédiate que celle de Minty du problème du stable maximum (donc pour la version non pondérée seulement) vers le problème du couplage maximum (de complexité $\mathcal{O}(n^4)$), et assez proche de la méthode dite de *struction*. Les graphes augmentants seront étudiés plus en détail dans la section 2.3 et la *struction* dans la section 2.5. Enfin, Oriolo, Pietropaoli et Stauffer [291] ont présenté en 2008 un nouvel algorithme pour le calcul d'un stable de poids maximum dans les graphes sans griffe, basé sur des décompositions et de complexité $\mathcal{O}(n^6)$ ⁷.

Théorème 2.8 Dans un graphe sans griffe, on peut trouver un stable de cardinalité maximum en temps $\mathcal{O}(n^3)$ et un stable de poids maximum en temps $\mathcal{O}(n^6)$.

En 1997, Faudree, Flandrin et Ryjáček [132] ont réalisé une synthèse très complète des résultats connus alors au sujet des graphes sans griffe, à laquelle nous invitons le lecteur à se reporter pour plus d'informations. Plus récemment, Chudnovsky et Seymour ont fourni un théorème de décomposition structurelle complète de l'ensemble des graphes sans griffe, à travers une série d'articles [79, 80, 81, 82, 83, 84, 85] dont on peut retrouver l'essentiel dans [78].

Graphes parfaits

De nombreuses autres classes, historiquement très étudiées, disposent d'algorithmes polynomiaux, voire linéaires, permettant de résoudre le problème du stable (de poids) maximum :

- les **graphes cordaux** (qui contiennent notamment les **graphes à seuil**⁸ (*threshold graphs*), introduits par Chvátal et Hammer [88, 89]) : un algorithme pour MIS a été donné par Gavril [153], et Frank a résolu MWIS [139];
- les **graphes de comparabilité**⁹ (*comparability graphs*) (qui contiennent les bipartis, les **graphes de permutation**¹⁰, les cographes

6. La complexité exacte de l'algorithme de Minty est rarement explicitée. Dans [254], Lozin et Milanič affirment qu'il s'exécute en temps $\mathcal{O}(n^7)$. Oriolo, Pietropaoli et Stauffer [291] émettent des doutes sur cette valeur de l'exposant et annoncent $\mathcal{O}(n^6)$.

7. Les auteurs espèrent pouvoir diminuer cette complexité par une analyse plus fine de leur algorithme.

8. Un graphe est à seuil si tout sommet v possède un poids $w(v)$ et s'il existe un réel S (le *seuil*) tel que (u, v) est une arête si et seulement si $w(u) + w(v) \geq S$.

9. Un graphe de comparabilité est un graphe dont les sommets sont les éléments d'un ensemble partiellement ordonné, reliés par une arête si et seulement s'ils sont comparables.

10. Étant donnée une permutation des entiers $1, 2, \dots, n$, un graphe de permutation a

(graphes sans P_4) et également les graphes à seuil) (voir par exemple [166]);

- les **graphes de co-comparabilité** (complémentaires des graphes de comparabilité, et qui contiennent les graphes d'intervalles et les co-graphes) [273].

Tous ces graphes sont des cas particuliers de graphes *parfaits*, évoqués au chapitre 1. En 1975, Chvátal [86] a décrit le polytope des stables des graphes parfaits. Puis, en 1981, Grötschel, Lovász et Schrijver [178, 179] ont montré les conséquences en optimisation combinatoire de la méthode de l'ellipsoïde. En particulier, ils ont introduit un algorithme qui résout en temps polynomial le problème de séparation associé au polytope des stables, ce qui a pour conséquence immédiate le théorème suivant :

Théorème 2.9 *Dans un graphe parfait, on peut trouver un stable de cardinalité maximum et un stable de poids maximum en temps polynomial*¹¹.

Comme l'on ne dispose pas, à ce jour, d'un algorithme *combinatoire* pour le problème du stable maximum dans les graphes parfaits, les algorithmes décrits dans les références qui précèdent conservent toute leur importance.

2.1.5 Théorèmes d'Alekseev

Dans les deux sections précédentes, nous avons vu que le problème du stable maximum est \mathcal{NP} -difficile dans un graphe en général, mais qu'il est possible de le résoudre en temps polynomial dans de nombreuses classes de graphes particulières.

En 1983, Alekseev [3] a énoncé un théorème (que nous désignerons dans la suite par *théorème d'Alekseev*) qui a généralisé un grand nombre de résultats connus, et qui s'apparente à une frontière entre polynomialité et \mathcal{NP} -difficulté pour MIS, puisqu'il fournit une condition suffisante pour que ce problème soit difficile dans une classe de graphes définie par une famille de sous-graphes induits interdits :

Théorème 2.10 (Alekseev (1983)) *Soit \mathcal{C} une classe de graphes définie par une famille finie \mathcal{F} de sous-graphes induits interdits. Si chaque graphe de \mathcal{F} a au moins une composante connexe non isomorphe au graphe $S_{i,j,k}$ (figure 2.1), alors le problème du stable (de poids) maximum est \mathcal{NP} -difficile dans la classe \mathcal{C} .*

Dans le cas le plus simple où \mathcal{F} est réduite à un seul graphe H , on en déduit immédiatement que MIS est \mathcal{NP} -difficile dans les classes des

pour sommets ces entiers et deux sommets sont reliés par une arête si et seulement si les entiers correspondants sont dans l'ordre inverse dans la permutation.

¹¹. Nous n'avons pas été en mesure de trouver une source écrite dans laquelle figure la complexité de l'algorithme de l'ellipsoïde appliqué aux graphes parfaits. Plusieurs sources orales ont évoqué une complexité de l'ordre de $\mathcal{O}(n^8)$ au moins.

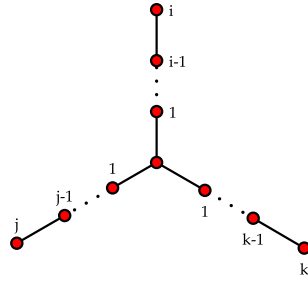


FIGURE 2.1 – Le graphe $S_{i,j,k}$, homéomorphe à la griffe (i, j, k peuvent être nuls).

graphes sans triangle, sans carré ou encore sans C_5 puisque dans ces trois cas H n'est pas homéomorphe¹² à une griffe. En notant \mathcal{S} la classe des graphes dont toutes les composantes sont des arbres à au plus 3 feuilles, on peut reformuler ce théorème ainsi :

Théorème 2.10' Soit \mathcal{C} une classe de graphes définie par une famille finie de sous-graphes induits interdits. Si $\mathcal{S} \subseteq \mathcal{C}$, alors \mathcal{C} est MIS-difficile.

La preuve repose sur une opération appelée *séparation de sommet* (cf. section 2.5.5). Notons que personne, à ce jour, n'a montré que cette condition est *nécessaire* à la \mathcal{NP} -difficulté de MIS en général (c'est-à-dire que MIS serait polynomial dans une classe si au moins un élément de \mathcal{S} en est exclu) ; néanmoins Alekseev et Korobitsyn [9] ont prouvé que c'est le cas si \mathcal{C} est **fortement héréditaire**, *i.e.* close par suppression d'arête (autrement dit, tout sous-graphe de \mathcal{C} , même non induit, appartient à \mathcal{C})¹³ :

Théorème (Alekseev et Korobitsyn (1992)) Soit \mathcal{C} une classe de graphes **fortement héréditaire**. Si $\mathcal{S} \not\subseteq \mathcal{C}$, alors \mathcal{C} est MIS-facile.

Le théorème 2.10 nous incite donc à considérer les classes de graphes qui ont au moins un sous-graphe interdit appartenant à \mathcal{S} , et en particulier, celles qui n'en ont qu'un seul. Examinons les cas où ce graphe (appelons le H) est connexe, pour les premières valeurs de $i + j + k$:

- si $i + j + k = 1$, un graphe ne contenant pas H comme sous-graphe induit est réduit à un graphe sans arête, donc on peut résoudre MIS de manière immédiate ;
- si $i + j + k = 2$, H est isomorphe au graphe $P_3 (= S_{0,1,1})$; or un graphe sans P_3 est un graphe pour lequel chaque composante connexe est une clique. Ici encore, on résout donc MIS de manière triviale, en choisissant un sommet dans chaque clique ;
- si $i + j + k = 3$, deux cas sont possibles (à isomorphisme près) : H est soit une griffe ($= S_{1,1,1}$), soit un $P_4 (= S_{0,1,2})$. Nous avons évoqué précédemment les algorithmes de Minty et de Sbihi qui permettent de

12. Informellement, deux graphes sont **homéomorphes** si l'un peut être obtenu à partir de l'autre en remplaçant des arêtes par des chaînes.

13. Alekseev les appelle *classes fortement héréditaires* dans [6] ; depuis, dans les autres articles sur le sujet, elles sont appelées classes *monotones* [8].

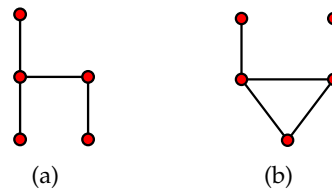


FIGURE 2.2 – (a) La chaise et (b) le taureau

résoudre MIS en temps polynomial dans la classe des graphes sans griffe. En ce qui concerne les graphes sans P_4 (ou cographes), comme ce sont des graphes de comparabilité, on peut leur appliquer l'algorithme cité plus haut. Nous verrons cependant dans la section 2.4 une autre technique, la décomposition modulaire, particulièrement élégante pour résoudre le problème du stable maximum cette classe.

Intéressons-nous de plus près au cas $i + j + k = 4$. À nouveau, deux configurations sont possibles pour H : soit c'est une **chaise**¹⁴ ($= S_{1,1,2}$) (*chair*, parfois appelée aussi *fork* en anglais) (figure 2.2(a)), soit c'est un P_5 ($= S_{0,2,2}$). On voit immédiatement que la classe des graphes sans chaise généralise celle des graphes sans griffe. La complexité de MIS dans cette classe est longtemps restée une question ouverte (quelques résultats partiels étaient cependant connus, par exemple pour la classe des graphes (chaise, taureau¹⁵)-libres [112]), jusqu'à ce qu'Alekseev [5, 7] réussisse en 1999 à étendre la méthode des graphes augmentants utilisée pour les graphes sans griffe à la classe des graphes sans chaise (cf. section 2.3). Cependant cette technique ne fonctionne que dans la version non-pondérée du problème. Pour la version pondérée, ce sont Lozin et Milanič [253, 254] qui ont décrit le premier algorithme polynomial en 2006, utilisant la décomposition modulaire¹⁶.

Théorème 2.11 (Alekseev (1999), Lozin et Milanič (2006)) *Dans un graphe sans chaise, on peut calculer un stable (de poids) maximum en temps $\mathcal{O}(n^7)$.*

En revanche, on ignore toujours s'il existe un tel algorithme pour la classe des graphes sans P_5 . De plus, tout graphe $S_{i,j,k}$ avec $i + j + k \geq 6$ contient un P_5 . Autrement dit :

Problème 2.1 *La classe des graphes sans P_5 constitue l'unique classe minimale définie par un unique sous-graphe induit connexe interdit pour laquelle on ignore le statut du problème du stable maximum.*

14. La première apparition du nom *chair* pour ce graphe semble être dans [112].

15. Un *taureau* (*bull*) est le graphe obtenu à partir d'un P_5 $\{a, b, c, d, e\}$ en reliant les sommets b et d (cf. figure 2.2(b)).

16. Notons que la construction proposée par Alekseev est relativement lourde : Lozin et Milanič estiment la complexité de l'algorithme après une analyse grossière à $\mathcal{O}(n^{10})$! Le leur s'exécute quant à lui en temps $\mathcal{O}(nC)$, où C est le temps nécessaire pour résoudre le problème du stable maximum dans les graphes sans griffe (soit *a priori* $\mathcal{O}(n^6)$ avec l'algorithme d'Oriolo, Pietropaoli et Stauffer [291]).

Voici donc la motivation pour l'étude de la stabilité dans les graphes sans P_5 . Intéressons-nous à présent de plus près aux propriétés de cette classe.

2.2 LA CLASSE DES GRAPHES SANS P_5

2.2.1 Propriétés des graphes sans P_5

Propriétés structurelles

Bien avant la démonstration du théorème 2.11, plusieurs chercheurs avaient exploré la classe des graphes sans P_5 , souvent dans le but de généraliser des propriétés existant dans des sous-classes, comme les cographes ou les graphes sans $2K_2$. Par exemple, Wolk [345, 346] avait montré que :

Théorème 2.12 (Wolk (1962)) *Un graphe est sans P_4 et sans C_4 si et seulement si chacun de ses sous-graphes induits connexes contient un sommet dominant.*¹⁷

En 1990, Bacsó et Tuza [18] et Cozzens et Kelleher [104] ont indépendamment généralisé ce théorème aux cliques dominantes de taille quelconque¹⁸ :

Théorème 2.13 (Bacsó et Tuza (1990), Cozzens et Kelleher (1990)) *Un graphe est sans P_5 et sans C_5 si et seulement si chacun de ses sous-graphes induits connexes contient une clique dominante.*

Une version plus forte de ce théorème a ensuite été donnée par Bacsó et Tuza [19] et indépendamment par Goddard et Henning [162] :

Théorème 2.14 (Bacsó et Tuza (2004), Goddard et Henning (2004)) *Un graphe est sans P_5 et sans C_5 si et seulement si dans chacun de ses sous-graphes induits connexes, le nombre de domination est égal à la taille d'une clique dominante minimum.*

On peut renforcer le théorème 2.13 si l'on exclut les *star-cutsets* ; un **star-cutset**¹⁹ est un ensemble de sommets S possédant un sommet dominant (de S) et tel que $G[V \setminus S]$ n'est pas connexe :

17. Les graphes (P_4, C_4) -libres sont connus dans la littérature sous les noms **graphes de comparabilité d'arbres** (*comparability graphs of trees*) [345, 346], **graphes à quasi-seuil** (*quasi-threshold graphs*) [259, 347] ou encore **graphes trivialement parfaits** (*trivially perfect graphs*) [165]. Ce sont les cographes qui sont aussi cordaux, ou, de manière équivalente, les cographes qui sont aussi d'intervalles [347].

18. Plus exactement, ce théorème est dû à Bacsó et Tuza, Cozzens et Kelleher ayant seulement prouvé qu'un graphe connexe sans P_5 et sans C_5 possède une clique dominante.

19. On pourrait dire *ensemble d'articulation étoilé*, mais le terme *star-cutset* semble beaucoup plus usité en français.

Théorème 2.15 (Bacsó et Tuza (1990)) *Soit G un graphe (P_5, C_5) -libre sans star-cutset. Alors tout sommet de G appartient à une arête dominante (ou bien G n'a pas d'arête et $V \leq 2$).*

Si l'on autorise le graphe à posséder des C_5 , on peut s'assurer quand même de l'existence d'une clique dominante en formulant certaines hypothèses supplémentaires :

Théorème 2.16 (Bacsó et Tuza (1990)) *Soit G un graphe connexe sans P_5 . S'il existe un sommet v qui n'est l'extrémité d'aucun P_4 de G , alors il existe une clique dominante contenant v .*

Corollaire 2.2 (Bacsó et Tuza (1990)) *Soit G un graphe connexe sans P_5 . Si G contient un sommet d'articulation, alors il contient une clique dominante.*

De plus, si l'on tolère d'autres graphes dominants que les cliques, on peut s'affranchir de ces hypothèses supplémentaires, comme le montre le théorème suivant :

Théorème 2.17 (Bacsó et Tuza (1990)) *Tout graphe connexe sans P_5 contient une clique dominante ou un P_3 dominant.*

qui est une généralisation d'un théorème d'El-Zahar et Erdős :

Théorème 2.18 (El-Zahar et Erdős (1985) [124]) *Tout graphe connexe sans $2K_2$ contient une clique dominante ou un P_3 dominant.*

Caractérisations

Les théorèmes précédents ne sont que des conditions *nécessaires* pour qu'un graphe soit sans P_5 . Il est cependant possible de caractériser pleinement ces graphes (c'est-à-dire d'énoncer des conditions *nécessaires et suffisantes*). Comme pour les propriétés structurelles ci-dessus, les caractérisations connues reposent sur la notion d'ensemble dominant, mais il nous faut auparavant introduire la notion plus générale d'ensemble *d-dominant*. Dans un graphe connexe G , la **distance** entre deux sommets est la longueur de la plus courte chaîne reliant ces sommets. On définit le **k -centre** $C_k(G)$ du graphe comme l'ensemble des sommets à distance au plus k de tout autre sommet, et on dit qu'un ensemble de sommets S est **d -dominant** si pour tout sommet x de G il existe un sommet de S à distance au plus d de x (la notion de domination définie précédemment correspond donc au cas $d = 1$).

Théorème 2.19 (Bacsó et Tuza (1990) [17]) *Un graphe G est sans P_{2k+1} ($k \geq 2$) si et seulement si pour chacun de ses sous-graphes induits connexes H , $C_k(H)$ est un ensemble $(k - 1)$ -dominant de H .*

Corollaire 2.3 *Un graphe G est sans P_5 si et seulement si pour chacun de ses sous-graphes induits connexes H , $C_2(H)$ est un ensemble dominant de H .*

Par la suite, Liu et Zhou [248] ont renforcé ce résultat en explicitant la nature que peut prendre le graphe $C_2(H)$:

Théorème 2.20 (Liu et Zhou (1994)) *Un graphe G est sans P_5 si et seulement si chacun de ses sous-graphes induits connexes contient un C_5 dominant ou une clique dominante.*

2.2.2 Trouver une clique dominante dans un graphe sans P_5

En général, il est \mathcal{NP} -difficile de trouver une clique dominante dans un graphe [150]. Dans [18], Bacsó et Tuza affirment que « toutes [leurs] preuves peuvent être utilisées pour obtenir des algorithmes polynomiaux », sans donner explicitement un algorithme permettant de déterminer une clique dominante d'un graphe (P_5, C_5) -libre ; cependant, une rapide analyse de la preuve du théorème 2.13 conduit effectivement à un algorithme de complexité $O(n^3)$ (voire même $O(n^{2,376})$ si l'on utilise l'algorithme de Coppersmith-Winograd pour calculer la matrice des voisins à distance 2, et ainsi déterminer un sommet ayant un voisinage à distance 2 maximal, ainsi que spécifié dans la preuve). Cozzens et Kelleher mentionnent pour leur part explicitement un algorithme, différent, de complexité $O(n^3)$. Toutefois, nous avons montré, dans une note non publiée [262], qu'en utilisant des structures de données adéquates, il est possible de diminuer la complexité à $O(m)$.

Le principe de l'algorithme est très simple : à chaque itération, on choisit un sommet ; s'il est dominé par la clique courante, on passe à l'itération suivante, sinon on essaie de construire une nouvelle clique dominante. Notons qu'il s'agit là d'un algorithme *robuste*, c'est-à-dire que s'il est exécuté sur un graphe contenant un P_5 ou un C_5 , il essaie quand même de déterminer une clique dominante, et s'il n'y parvient pas, il donne la liste des sommets induisant un P_5 ou un C_5 . Les sommets sont numérotés de 1 à n et traités selon un parcours en profondeur ; les structures de données utilisées sont les suivantes :

$newK$ (resp. $oldK$) est un tableau de booléens de longueur n : $newK[i]$ (resp. $oldK[i]$) est vrai si et seulement si le sommet i appartient à la clique dominante courante (resp. ancienne). Ces tableaux nous permettent de savoir si un sommet appartient à la clique dominante en temps $O(1)$;

nK et oK sont des listes qui contiennent seulement les sommets des cliques dominantes courante et ancienne ;

Q est une file d'attente dans laquelle on insère les sommets à examiner ;

inQ est un tableau de booléens de longueur n : $inQ[i]$ est vrai si et seulement si le sommet i est déjà dans Q ou a déjà été examiné ;

$father$ est un tableau d'entiers de longueur n : $father[i]$ contient le numéro d'un voisin de i déjà examiné (ou 0 s'il n'en existe pas encore) ;

dom est un tableau d'entiers de longueur n : $dom[i]$ contient le nombre de voisins de i qui appartiennent à K (ce qui permet de savoir en temps $\mathcal{O}(1)$ si un sommet est dominé ou non.

Les lignes commençant par deux slashes sont des commentaires.

Algorithme 4

Entrée : Un graphe connexe G possédant n sommets étiquetés de 1 à n

Sortie : Une clique dominante de G ou s'arrête en donnant un P_5 ou un C_5

```

1: newK ← [false, false, ..., false]
2: oldK ← [false, false, ..., false]
3: inQ ← [false, false, ..., false]
4: father ← [0, 0, ..., 0]
5: dom ← [0, 0, ..., 0]
6:
7: // On commence arbitrairement par le sommet 1
8: inQ[1] ← true
9: father[1] ← 1
10: Q ← 1
11:
12: tant que Q n'est pas vide faire
13:   Soit v l'élément en tête de Q
14:   si dom[v] > 0 alors
15:     // v est dominé, on ajoute ses voisins non explorés dans Q
16:     retirer v de Q
17:     pour chaque voisin w de v faire
18:       si inQ[w] = false alors
19:         father[w] ← v
20:         ajouter w en fin de Q
21:         inQ[w] ← true
22:     fin si
23:   fin pour
24: sinon
25:   // v n'est pas dominé, on essaie de construire une nouvelle clique dominante avec son père f
26:
27:   f ← father[v]
28:   dom[p] ← dom[p] + 1
29:   oK ← nK
30:   nK ← f
31:

```

```

32:   pour chaque sommet  $i$  dans  $oK$  faire
33:      $oldK[i] \leftarrow \mathbf{true}$ 
34:      $newK[i] \leftarrow \mathbf{false}$ 
35:   fin pour
36:    $newK[f] \leftarrow \mathbf{true}$ 
37:
38:   pour chaque voisin  $w$  de  $f$  faire
39:      $dom[w] \leftarrow dom[w] + 1$ 
40:     si  $oldK[i] = \mathbf{true}$  alors
41:       ajouter  $w$  dans  $nK$ 
42:        $newK[w] \leftarrow \mathbf{true}$ 
43:     fin si
44:   fin pour
45:
46:   pour chaque sommet  $i$  dans  $oK$  faire
47:     si  $newK[i] = \mathbf{false}$  alors
48:        $dom[i] \leftarrow dom[i] - 1$ 
49:       pour chaque neighbor  $j$  of  $i$  faire
50:          $dom[w] \leftarrow dom[w] - 1$ 
51:         si  $dom[j] = 0$  alors
52:           afficher " $G$  contient le  $C_5$  ou  $P_5$  suivant : " +
               $(j, i, father[i], f, v)$ 
53:           stop
54:         fin si
55:       fin pour
56:     fin si
57:      $oldK[i] \leftarrow \mathbf{false}$ 
58:   fin pour
59: fin si
60: fin tant que
61:
62: retourner  $newK$ 

```

Nous prouvons à présent le théorème suivant :

Théorème 2.21 (Maffray, Morel (2010)) *On peut déterminer une clique dominante d'un graphe (P_5, C_5) -libre en temps $\mathcal{O}(m)$ grâce à l'algorithme 4.*

Démonstration. La correction de cet algorithme est donnée dans [104]. Nous prouvons ici qu'ils s'exécute en temps $\mathcal{O}(m)$.

Tout d'abord, remarquons que lorsqu'un sommet f entre dans la clique dominante, le compteur $dom[w]$ de chaque voisin w de f augmente, et ceux qui n'ont pas encore été examinés sont ajoutés à la file d'attente Q (lignes 38 à 44). Ensuite, puisque leur compteur est strictement positif, ils sont successivement retirés de cette même file (lignes 14 à 23). Autrement dit lorsqu'un sommet entre dans la clique dominante, ses voisins sont ceux qui sont examinés immédiatement après lui.

Ceci implique qu'un sommet ne peut entrer dans la clique dominante qu'au plus une fois ; en effet, si un sommet v devait entrer une deuxième

fois dans la clique dominante, ce serait parce que l'un de ses voisins n'est pas dominé et n'a pas été examiné lorsque v était dans la clique, une contradiction.

Par conséquent, chaque sommet est considéré exactement une fois, et ainsi entre et sort de la file Q exactement une fois. Autrement dit, le cas v est dominé (lignes 14 à 23) est exécuté exactement une fois pour chaque sommet, et requiert $\mathcal{O}(d(v))$ opérations pour chaque sommet v ; cette étape nécessite donc $\mathcal{O}(m)$ opérations pour l'ensemble des sommets du graphe. Considérons maintenant la seconde partie de l'algorithme (lignes 24 à 59). $newK$ et $oldK$ sont des listes de sommets qui induisent une clique. Par construction, le nombre de sommets dans la clique dominante à une étape donnée de l'algorithme est au plus égal au degré du dernier sommet entré dans la clique, et nous savons qu'un sommet entre au plus une fois. Il s'ensuit que la première boucle **pour** (lignes 32 à 35) requiert un total de $\mathcal{O}(m)$ opérations; la seconde boucle **pour** (lignes 38 à 44) nécessite quant à elle clairement $\mathcal{O}(d(f))$, f étant le sommet qui entre dans la clique; cette étape nécessite donc elle aussi $\mathcal{O}(m)$ opérations pour la totalité du graphe. La dernière étape consiste en deux boucles **pour** imbriquées (lignes 48 à 49). Mais la boucle interne, qui nécessite clairement $\mathcal{O}(i)$ opérations, est exécutée seulement quand $newK[i]$ vaut **false**, c'est-à-dire quand le sommet i quitte la clique dominante. Puisqu'un sommet quitte la clique au plus une fois, cette étape ne requiert au total que $\mathcal{O}(m)$ opérations. Par conséquent, l'algorithme s'exécute en temps $\mathcal{O}(m)$. \square

Enfin, en ce qui concerne la classe plus générale des graphes sans P_5 , la preuve du théorème 2.20 donnée par Liu et Zhou n'est pas constructive. Mais récemment, van't Hof et Paulusma [336, 337] en ont donné une démonstration algorithmique, conduisant au corollaire suivant :

Corollaire 2.4 *On peut trouver une clique dominante ou un C_5 dominant d'un graphe sans P_5 en temps $\mathcal{O}(n^3)$.*

2.2.3 Propriétés des graphes sans P_5 bipartis

Il est facile de se convaincre que la classe des graphes *connexes* bipartis sans P_5 coïncide avec celle des graphes bipartis sans $2K_2$. Ces derniers sont connus dans la littérature sous les noms de **graphes de chaîne** (*chain graphs*) [349], ou encore de **graphes de différence** (*difference graphs*) [188]. La première dénomination vient de ce que l'on peut, de chaque côté de la bipartition, ordonner les sommets selon l'inclusion de leur voisinage, et ainsi former ce que l'on appelle une **chaîne** en théorie des ordres :

Proposition 2.4 *Un graphe biparti $G = (A, B; E)$ est sans $2K_2$ si et seulement si pour toute paire de sommets u et v de A , on a $N(u) \subseteq N(v)$ ou $N(v) \subseteq N(u)$ et il en va de même pour B .*

Le corollaire suivant découle directement de la proposition précédente :

Corollaire 2.5 *Tout graphe biparti connexe sans P_5 possède une arête dominante.*

La dénomination *graphe de différence* provient quant à elle de la propriété suivante :

Proposition 2.5 *Un graphe biparti $G = (A, B; E)$ est sans $2K_2$ si et seulement si on peut affecter des poids (réels) w_i aux sommets et s'il existe un réel positif t tel que $(i, j) \in E$ si et seulement si $|w(i) - w(j)| \geq t$.*

Les graphes de chaîne sont étroitement liés aux graphes à seuil (cf. section 2.1.4). Un graphe est à seuil si et seulement si son ensemble de sommets peut être partitionné en une clique C et un stable S tels que les sommets de S peuvent être ordonnés selon l'inclusion de leur voisinage. Ainsi, il est facile de voir qu'un graphe biparti $G = (A, B; E)$ est un graphe de chaîne si et seulement si le graphe obtenu à partir de G en reliant deux à deux les sommets de A est un graphe à seuil ; réciproquement, un graphe H dont l'ensemble de sommets peut être partitionné en une clique C et un stable S (un tel graphe est qualifié de **scindé** (*split graph*)) est un graphe à seuil si et seulement si le graphe biparti obtenu à partir de H en supprimant toutes les arêtes de C est un graphe de chaîne.

Une autre propriété des graphes de chaîne est particulièrement intéressante ici, car elle nous indique précisément le nombre de stables maximaux que peut posséder un tel graphe (voir [268]) :

Proposition 2.6 *Soit $G = (A, B; E)$ un graphe biparti sans P_5 connexe. Alors :*

- *il existe un entier $k \geq 1$, une partition A_1, \dots, A_k de A en k ensembles non vides et une partition B_1, \dots, B_k de B en k ensembles non vides telle qu'un sommet de A_i et un sommet de B_j sont adjacents si et seulement si $i + j \leq k + 1$;*
- *G a exactement $k + 1$ stables maximaux, qui sont A, B et $A_{i+1} \cup \dots \cup A_k \cup B_{k+1-i} \cup \dots \cup B_k$ pour chaque $i = 1, \dots, k - 1$.*

Avec les notations de la proposition 2.6, tout sommet de A_1 est complet à B^{20} , et tout sommet de B_1 est complet à A (autrement dit, toute arête dominante de G (voir la proposition 2.5) a l'une de ses extrémités dans A_1 et l'autre dans B_1). On en déduit que pour tout sommet v de $A_1 \cup B_1$, les non-voisins de v induisent un stable. Maffray a montré qu'une propriété similaire est vraie pour *tout* graphe connexe sans P_5 [260] ; nous y reviendrons au chapitre suivant.

D'un point de vue algorithmique, tester si un graphe biparti connexe donné $G = (A, B; E)$ est sans P_5 peut se faire comme suit.

²⁰. Pour rappel, A est complet à B lorsque tout sommet de A est adjacent à tout sommet de B .

Tant que A a un sommet a de degré $|B|$, alors enlever a de A (et mettre à jour le degré de chaque sommet de B). Tant que B a un sommet b de degré 0, alors enlever b de B .

Si cette procédure ne vide pas les ensembles A et B , alors déclarer que G n'est pas sans P_5 (ceci est correct en vertu de l'argument suivant : supposons que A et B ne soient pas vides à la fin de la procédure, et prenons un sommet $a \in A$ de degré maximum ; alors $d(a) < |B|$, donc il existe un sommet $b \in B \setminus N(a)$. La procédure implique que b a un voisin u dans A , et le choix de a implique qu'il existe un sommet $v \in B$ adjacent à a et pas à u . Alors $\{a, u, b, v\}$ induit un $2K_2$; autrement dit, G contient un P_5).

Sinon, déclarer que G est sans P_5 . Ceci est correct d'après la proposition précédente.

Cette procédure peut clairement être implémentée en temps linéaire. De plus, il est possible de calculer un stable de poids maximum en temps linéaire, simplement en examinant les $k + 1$ stables mentionnés à la fin de la proposition 2.6.

Dans les sections qui suivent, nous présentons un certain nombre de techniques développées pour résoudre le problème du stable, illustrées par des sous-classes des graphes sans P_5 dans lesquelles elles ont été utilisées avec succès.

2.3 LES GRAPHES AUGMENTANTS

La méthode des **graphes augmentants** tire son origine des travaux de Minty [279] et Sbihi [310] pour résoudre le problème du stable maximum dans les graphes sans griffe (c.f. section 2.1.4). Remarquant que dans de tels graphes, un sommet est adjacent à au plus deux sommets d'un stable, ils ont conçu des algorithmes capables de déterminer (en temps polynomial) des **chaînes S -augmentantes**, c'est-à-dire qui permettent d'augmenter la cardinalité d'un stable S , concept qui fut donc ensuite étendu à celui de graphe augmentant.

2.3.1 Notion de chaîne S -augmentante

Définition 2.2 Étant donné un graphe $G = (V, E)$ sans griffe, un stable S de G et une chaîne $P = v_0 - v_1 - \dots - v_k$,

- P est **S -alternée** (S -alternating path) si ses sommets appartiennent alternativement à S et $V \setminus S$,
- P est **S -augmentante** (S -augmenting path) si elle est S -alternée avec $v_0, v_k \notin S$ et $S \setminus \{v_1, v_3, \dots, v_{k-1}\} \cup \{v_0, v_2, \dots, v_k\}$ est un stable.

Il est évident qu'à une chaîne M -alternée associée à un couplage M d'un graphe G , correspond dans son graphe adjoint $L(G)$ une chaîne S -alternée associée au stable S constitué des sommets de $L(G)$ correspondant aux arêtes de M , et réciproquement. Autrement dit :

Proposition 2.7 *Un stable S d'un graphe adjoint est maximum si et seulement s'il n'existe pas de chaîne S -augmentante.*

Comme indiqué précédemment, l'algorithme du couplage maximum d'Edmonds permet donc de résoudre le problème du stable maximum dans les graphes adjoints. Les algorithmes de Minty et Sbihi reposent sur la généralisation de la propriété précédente aux graphes sans griffe :

Proposition 2.8 *Soient G un graphe sans griffe et S un stable maximal de G . S est maximum si et seulement s'il n'existe pas de chaîne S -augmentante.*

Démonstration.

\Rightarrow Résulte directement de la définition d'une chaîne S -augmentante.

\Leftarrow Soit S' un stable maximum de G . Considérons le sous-graphe de G induit par $S \Delta S'$ (où Δ désigne la différence symétrique) : tout sommet de ce sous-graphe a au plus deux voisins puisque G est sans griffe et que les deux stables S et S' sont maximaux. Autrement dit, les composantes connexes de $G[S \Delta S']$ sont soit des cycles pairs (puisque $G[S \Delta S']$ est biparti), soit des chaînes, et dans ce dernier cas forcément d'ordre pair puisque G ne contient pas de chaîne S -augmentante. Donc $|S| = |S'|$. \square

2.3.2 Méthode des graphes augmentants²¹

Pendant quinze ans, l'idée des chaînes S -augmentantes est restée relativement peu exploitée, et il faut attendre le milieu des années 1990 pour qu'elle soit étendue à des graphes quelconques. Cette approche s'est révélée être un outil efficace pour obtenir des algorithmes d'approximation pour le problème du stable maximum [184], pour calculer des bornes sur le nombre de stabilité [115], ou encore pour résoudre le problème en temps polynomial dans certaines classes de graphes, en particulier des classes de graphes sans P_5 , résultats que nous détaillerons ci-dessous.

Définition 2.3 *Un graphe S -augmentant (S -augmenting graph) pour un stable S est un graphe biparti $H = (B, R; E)$ où $B \subseteq S$ et $R \subseteq V \setminus S$ vérifient :*

1. $|R| > |B|$,
2. $N(R) \cap (S \setminus B) = \emptyset$ (i.e. les sommets de R ne sont pas adjacents à des sommets de S autres que ceux de B).

Clairement, si H est S -augmentant, S n'est pas maximum puisque l'ensemble $S' = (S \setminus B) \cup R$ est stable et vérifie $|S'| > |S|$. On dira que S' est obtenu par **H -augmentation** de S et on appellera **incrément** de H le nombre $|R| - |B| = |S'| - |S|$ (cf. figure 2.3).

Réciproquement, si S n'est pas maximum, alors il existe un stable S' tel

²¹. Les notions introduites dans cette section proviennent principalement de [11], [203], [255], [278] et [281].

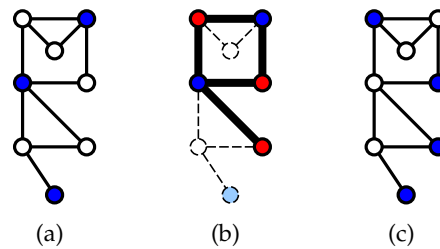


FIGURE 2.3 – Un exemple d'augmentation de stable. Le graphe de la figure (a) possède un stable de cardinalité 3 (sommets bleus) ; ce stable admet un graphe augmentant (représenté par les arêtes épaisses sur la figure (b)) ; les sommets de B sont les sommets bleus et les sommets de R sont les sommets rouges) ; la figure (c) montre le stable obtenu après augmentation.

que $|S'| > |S|$, et le sous-graphe induit par $S \triangle S'$ est S -augmentant. On vient donc de prouver le *théorème des graphes augmentants* :

Théorème 2.22 (Théorème des graphes augmentants) *Dans un graphe, un stable S est maximum si et seulement s'il n'existe pas de graphe S -augmentant.*

On déduit de ce théorème un algorithme général permettant de déterminer un stable maximum dans un graphe G : générer un premier stable S quelconque puis, tant qu'il existe un graphe S -augmentant H , appliquer la H -augmentation à S . Mais trouver un graphe augmentant est évidemment en général un problème \mathcal{NP} -difficile, puisque MIS est lui-même \mathcal{NP} -difficile. Ainsi, pour que cet algorithme soit polynomial dans une classe de graphes donnée, il faut :

- connaître la liste complète des graphes augmentants pour cette classe (cf. section 2.3.3),
- avoir un algorithme polynomial permettant de les détecter (cf. section 2.3.4).

Le premier point consiste donc à caractériser les graphes bipartis de la classe en question. Clairement, les graphes bipartis ne sont pas tous augmentants : ainsi, un cycle (pair, puisque le graphe est biparti) n'est jamais un graphe augmentant, car la condition 1 de la définition 2.3 n'est jamais vérifiée. On peut également retirer de la liste, sans perte de généralité, les graphes augmentants qui ne sont pas *minimaux* :

Définition 2.4 *Etant donné un stable S , un graphe S -augmentant est **minimal** s'il ne contient aucun sous-graphe propre induit S -augmentant.*

Il est facile de voir qu'il existe des graphes augmentants qui ne sont jamais minimaux : par exemple, si une griffe est S -augmentante alors chacun de ses sous-graphes induits $K_{1,2}$ est lui-même S -augmentant. La propriété qui suit caractérise un graphe augmentant minimal :

Proposition 2.9 $H = (B, R; E)$ est un graphe augmentant minimal si et seulement si :

1. H est connexe ;
2. $|B| = |R| - 1$;
3. $\forall X \subseteq B, |X| < |N(X)|$.

Démonstration.

\Rightarrow La condition 1 est évidente. Si $|B| < |R| - 1$, $H[B \cup (R \setminus \{v\})], \forall v \in B$ est un sous-graphe propre de H qui est aussi augmentant, ce qui contredit la minimalité de H et suffit à prouver la condition 2. Pour montrer la condition 3, supposons qu'il existe $X \subseteq B$ tel que $|X| \geq |N(X)|$. Alors, les sommets de $(B \setminus X) \cup (R \setminus N(X))$ induisent un sous-graphe propre de H qui est augmentant (car $|B| < |R|$).

\Leftarrow Soit $H = (B, R; E)$ un graphe augmentant satisfaisant les conditions 1–3. Supposons que H ne soit pas minimal : il contient donc un sous-graphe propre $H' = (B', R'; E')$ qui est aussi augmentant. Alors $B' \subset B$ (si B était égal à B' , comme $|R'| \geq |B'| + 1$ et $|B| + 1 = |R|$, on aurait $|R'| \geq |R|$ d'où $R' = R$ et donc $H' = H$). Par conséquent, l'ensemble $X = B \setminus B'$ n'est pas vide. De plus, puisque H' est augmentant, $N(R') \cap X = \emptyset$ i.e. $N(X) \subseteq R \setminus R'$, d'où $|N(X)| \leq |R \setminus R'| = |R| - |R'| \leq |B| + 1 - (|B'| + 1) = |B \setminus B'| = |X|$, contredisant ainsi la condition 3. \square

Une autre notion qui peut être utile dans certains cas est celle de graphe augmentant *maximum* :

Définition 2.5 Un graphe augmentant est *maximum* si son incrément est supérieur à celui de tout autre graphe augmentant pour un stable S .

L'importance de cette notion peut être perçue la proposition suivante :

Proposition 2.10 Soient S un stable et H un graphe S -augmentant. Le stable obtenu par H -augmentation est maximum si et seulement si H est un graphe S -augmentant maximum.

Nous terminons cette section en présentant un concept similaire à celui des graphes augmentants, mais pour le problème du stable maximal de cardinalité *minimale* (i.e. du dominant stable minimum, cf. section 2.1.2). On définit pour ce problème le concept de graphe *diminuant* :

Définition 2.6 Un graphe biparti $H = (B, R; E)$ est *S -diminuant* pour un stable maximal S si :

- 1' $|B| < |R|$,
- 2' $N(B) \cap (S \setminus R) = \emptyset$,
- 3' $(S \setminus R) \cup B$ est un stable maximal de G .

Naturellement, si H est S -diminuant, S n'est pas minimum puisque l'ensemble $B \cup (S \setminus R)$ est de cardinalité strictement inférieure à celle de S . Réciproquement, si S n'est pas minimum, soit S' un stable maximal de cardinalité inférieure; le sous-graphe induit par $S \triangle S'$ est clairement S -diminuant. On vient de prouver le théorème des graphes diminuants :

Théorème 2.23 (des graphes diminuants) *Dans un graphe, un stable maximal S est minimum si et seulement s'il n'existe pas de graphe S -diminuant.*

La condition supplémentaire $3'$ de la définition, qui exprime que S est aussi un dominant (proposition 2.1.2), rend la détection d'un graphe diminuant plus difficile que celle d'un graphe augmentant; le problème du stable maximal minimum est d'ailleurs \mathcal{NP} -difficile dans les graphes sans griffe. Boliac et Lozin [33] sont néanmoins parvenus, en utilisant les graphes diminuants, à établir un certain nombre de propriétés et à résoudre le problème en temps polynomial dans une sous-classe des graphes sans griffe :

Proposition 2.11 (Boliac, Lozin (2003)) *Tout graphe diminuant minimal d'un graphe sans griffe est une forêt.*

Proposition 2.12 (Boliac, Lozin (2003)) *Tout graphe diminuant minimal d'un graphe (P_6, griffe) -libre est connexe.*

Théorème 2.24 (Boliac, Lozin (2003)) *Étant donné un graphe (P_6, griffe) -libre d'ordre n , on peut trouver un stable maximal minimum en temps $\mathcal{O}(n^3)$.*

2.3.3 Caractérisation des graphes augmentants

Pour résoudre MIS dans une classe donnée grâce à la méthode des graphes augmentants, la première étape est de décrire les graphes bipartis de ladite classe. Voici quelques exemples pour lesquels les graphes augmentants sont parfaitement connus (hormis la première, nous réservons les classes de graphes sans P_5 pour la section 2.3.5) :

- graphes sans P_4 ($S_{0,1,2}$)²² : c'est un exercice simple que de montrer que tout biparti connexe de cette classe est complet, et par conséquent, que tout graphe augmentant minimal sans P_4 est de la forme $K_{n,n+1}$. Les graphes augmentants ne sont cependant pas forcément la meilleure approche pour cette classe; la décomposition modulaire (cf. section 2.4) s'y révèle particulièrement efficace et permet d'obtenir des algorithmes linéaires.

²² Pour les classes définies par un seul sous-graphe induit interdit, nous redonnons l'expression de ce graphe sous la forme $S_{i,j,k}$ (cf. théorème 2.10).

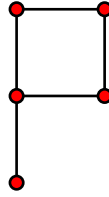


FIGURE 2.4 – Un drapeau.

- *graphes sans griffe* ($S_{1,1,1}$) : tout graphe augmentant minimal de cette classe est une chaîne de longueur paire (cf. la démonstration de la proposition 2.8) ;
- *graphes sans chaise* ($S_{1,1,2}$) : les graphes bipartis connexes sans chaise ont été caractérisés par Alekseev [5] : ce sont les graphes G tels que soit $\Delta(G) \leq 2$ (ils sont donc sans griffe, et par conséquent ce sont des chaînes d'après le point précédent) soit $\Delta(\tilde{G})^{23} \leq 1$ (c'est-à-dire que chaque sommet de G a au plus un non-voisin dans l'autre partie de la bipartition ; Alekseev les appelle des *complexes*). Ainsi, un graphe augmentant minimal sans chaise est soit une chaîne de longueur paire, soit un complexe ;
- *graphes* (P_6, C_4)-*libres* : Mosca a prouvé dans [283] que tout graphe augmentant minimal de cette classe est un *arbre augmentant simple*, c'est-à-dire un graphe obtenu à partir d'une étoile $K_{1,k}$ en insérant un sommet sur chaque arête ;
- *graphes* ($P_7, \text{drapeau}^{24}$)-*libres* : Alekseev et Lozin [11] ont montré que tout graphe augmentant minimal ($P_7, \text{drapeau}$)-libre est soit un biparti complet $K_{n,n+1}$, soit une *plante* (i.e. un graphe obtenu à partir d'un arbre augmentant simple en reliant un sommet v à tous les sommets de degré 1, et un sommet w à v (remarque : cette classe généralise celle des graphes (P_6, C_4)-libres, et celle des graphes ($P_5, \text{drapeau}$)-libres).

Notons que pour ces cinq classes, il est également possible de résoudre MWIS en temps polynomial. Cela a déjà été signalé pour la deuxième et la troisième ; la décomposition modulaire permet de résoudre le problème en temps linéaire dans les graphes sans P_4 ; Brandstädt et Hoàng [42, 43] ont obtenu un algorithme polynomial (de complexité $\mathcal{O}(n^3m)$) pour les graphes (P_6, C_4)-libres en combinant la décomposition modulaire et la décomposition par clique-séparateurs (cf. section 2.6) ; Mosca [286] a par la suite appliqué cette même méthode aux graphes ($P_7, \text{drapeau}$)-libres, obtenant ainsi un algorithme de complexité $\mathcal{O}(n^9m)$.

2.3.4 Comment trouver un graphe augmentant en temps polynomial ?

Connaître la liste des graphes augmentants minimaux d'une classe donnée ne suffit pas à résoudre MIS en temps polynomial. Encore faut-il être capable de les détecter en temps polynomial (ce qui n'est pas toujours possible (à moins que $\mathcal{P} = \mathcal{NP}$!) puisque trouver un stable de cardinalité supérieure à k pour un k donné est déjà \mathcal{NP} -complet en général).

23. \tilde{G} désigne le biparti complémentaire de G , i.e. le graphe $\tilde{G} = (V_1, V_2, (V_1 \times V_2) \setminus E)$.

24. Un **drapeau** (*banner*, parfois appelé P) s'obtient à partir d'un $P_5 \{a, b, c, d, e\}$ en reliant les sommets b et e (cf. figure 2.4). Notons que le problème du stable maximum est \mathcal{NP} -difficile dans les graphes sans drapeau en vertu du théorème 2.10.

Ainsi, bien que l'on connaisse parfaitement la structure des graphes bipartis connexes sans P_5 (cf. section 2.2.3), on ne dispose à ce jour d'aucun algorithme polynomial permettant de les détecter. D'un autre côté, nous savons trouver une chaîne augmentante dans un graphe sans griffe, grâce aux algorithmes de Sbihi et de de Minty. Ce résultat a été généralisé à deux surclasses :

- Gerber, Hertz et Lozin ont prouvé dans [157] qu'en modifiant légèrement l'algorithme de Minty, on peut déterminer les chaînes augmentantes dans la classe des graphes sans $S_{1,2,3}$;
- Hertz, Lozin et Schindl [204] ont montré que le problème qui consiste à trouver des chaînes augmentantes dans les graphes $(S_{1,2,i}$, drapeau)-libres (pour $i \geq 1$ fixé), peut se réduire de manière polynomiale au problème de recherche des chaînes augmentantes dans les graphes sans griffe.

Hertz et Lozin [203] décrivent une méthode pour trouver les graphes augmentants qui sont bipartis complets, comme c'est le cas pour les graphes sans P_4 . Cette approche a été appliquée pour la première fois aux graphes sans chaise par Alekseev [5], et aux graphes sans P_5 ni drapeau [252]. Elle a ensuite été étendue à la classe entière des graphes sans drapeau [11] et à la classe entière des graphes sans P_5 , pour laquelle Boliac et Lozin ont prouvé que MIS est polynomialement équivalent au problème de la recherche de graphes augmentants qui contiennent un P_4 [32]. Hertz et Lozin ont étendu cette idée à la classe des graphes sans drapeau₂ (graphe obtenu en connectant un sommet au sommet de degré 1 d'un drapeau), qui contient toutes les classes que l'on vient de citer.

2.3.5 Application aux graphes sans P_5

Nous allons à présent donner quelques sous-classes de graphes sans P_5 pour lesquelles les graphes augmentants ont permis d'obtenir un algorithme polynomial pour résoudre MIS (voire MWIS).

Mosca a été l'un des premiers à adopter cette approche dans les graphes sans P_5 [282]. Il a montré que dans cette classe, stables maximal et maximum sont liés par une relation plus forte que le théorème des graphes augmentants (théorème 2.22). Pour décrire cette relation, nous avons besoin de quelques définitions préliminaires.

Définition 2.7 Soient S un stable d'un graphe $G = (V, E)$, et $v \in V \setminus S$. On note $H(v, S)$ l'ensemble $\{w \in V \setminus (S \cup N[v]) \mid N_S(w) \subseteq N_S(v)\}$.

Autrement dit, $\forall v \in V$, $H(v, S)$ désigne l'ensemble des non-voisins $w \in V \setminus S$ de v vérifiant $N_S(w) \subseteq N_S(v)$. Clairement, si S est maximal, on a $N_S(w) \neq \emptyset, \forall w \in H(v, S)$ (Sinon, on pourrait ajouter w à S). La figure 2.5 illustre cette définition ; sur ce graphe, on a $H(v, S) = \{u_1, u_2\}$ et $H(w, S) = \{u_1, u_3\}$.

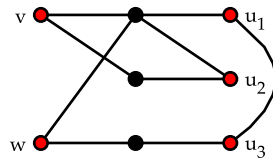


FIGURE 2.5 – Illustration de la définition 2.7. S est constitué des sommets noirs, et on a $H(v, S) = \{u_1, u_2\}$ et $H(w, S) = \{u_1, u_3\}$

Définition 2.8 Soient S un stable d'un graphe $G = (V, E)$, et $v \in V \setminus S$. On dit que v est un sommet S -augmentant si $G[H(v, S)]$ contient un stable S_v tel que $|S_v| \geq |N_S(v)|$. Si de plus $|N_S(v)| = 1$, on parlera de sommet S -augmentant simple.

Par exemple, sur la figure 2.5, v est S -augmentant alors que w ne l'est pas. Notons que d'après la définition précédente, le graphe biparti induit par $S \cup (\{v\} \cup S_v)$ est augmentant.

Proposition 2.13 Soit S un stable maximal d'un graphe $G = (V, E)$. Si S admet un sommet augmentant, il n'est pas maximum.

Démonstration. En effet, avec les notations de la définition précédente, l'ensemble $(S \setminus N_S(v)) \cup \{v\} \cup S_v$ est stable et a plus d'éléments que S . \square

Mosca a prouvé que **si G est sans P_5 , la condition précédente est aussi nécessaire :**

Proposition 2.14 Soit S un stable maximal d'un graphe $G = (V, E)$ sans P_5 . Si aucun sommet de $V \setminus S$ n'est S -augmentant, S est maximum.

Démonstration. (Par contraposée.) Supposons que S ne soit pas maximum. D'après le théorème 2.22, G contient un graphe S -augmentant $B = (B, N; E)$; de plus, d'après la condition (2) de la définition 2.3 et le corollaire 2.5, il existe un sommet $v \in N$ tel que $N_S(v) = N_B(v) = B$. En notant $T = N \setminus \{v\}$, on a clairement $T \subseteq H(v, S)$ (et T est stable); et comme $|N| > |B|$, $|T| \geq |N_S(v)|$. Ainsi, v est S -augmentant. \square

Ainsi, pour savoir si un stable maximal S d'un graphe $G = (V, E)$ sans P_5 est maximum, il suffit de tester si chaque sommet de $V \setminus S$ est S -augmentant; mais ceci implique de déterminer un stable de $H(v, S)$ de cardinalité $\geq |N_S(v)|$, ce qui est en général \mathcal{NP} -difficile. Mosca a montré que si l'on sait le faire en temps polynomial, on peut en déduire un algorithme polynomial pour calculer un stable maximum dans G :

Théorème 2.25 (Mosca (1997)) Grâce à l'algorithme Alpha (algorithme 5), on sait calculer un stable maximum en temps $\max\{\mathcal{O}(n^{k+2}), \mathcal{O}(n^4)\}$ dans tout graphe G d'ordre n , dès lors que l'on sait calculer un stable maximum de $G[H(v, S)]$ en temps $\mathcal{O}(n^k)$ pour tout stable maximal S de G et tout $v \in V \setminus S$.

Algorithme 5: Algorithme Alpha**Entrées :** Un graphe sans P_5 $G = (V, E)$ **Sorties :** Un stable maximum S de G $S \leftarrow \emptyset;$ **tant que** *il existe un sommet v non adjacent à S ou S -augmentant* **faire** **si** v n'est pas adjacent à S **alors** $S \leftarrow S \cup \{v\};$ **sinon** // v est S -augmentant $S \leftarrow (S \setminus N_S(v)) \cup \{v\} \cup S_v;$ (où S_v est un stable de $H(v, S)$ tel que $|S_v| \geq |N_S(v)|$) **fin****fin****retourner** $S;$

Les propositions 2.13 et 2.14 s'étendent facilement à la version pondérée du problème du stable, mais la preuve originale du théorème 2.25 faisait que ce n'était pas le cas pour ce dernier. Mosca [284] a réussi plus tard à obtenir cette généralisation, et simultanément à en améliorer la complexité :

Théorème 2.26 (Mosca (2004)) *on sait calculer un stable de poids maximum dans un graphe G sans P_5 pondéré d'ordre n en temps $\max\{\mathcal{O}(n^{k+1}), \mathcal{O}(n^3)\}$, dès lors que l'on sait calculer un stable de poids maximum de $G[H(v, S)]$ pour tout stable maximal S de G et tout $v \in V \setminus S$ en temps $\mathcal{O}(n^k)$.*

Les trois premiers résultats qui suivent ont été énoncés à l'origine avec le théorème 2.25, donc dans le cas non pondéré; nous avons simplement mis à jour les complexités en fonction du théorème 2.26. Nous verrons ensuite un théorème très fort qu'a donné Mosca et qui généralise ces trois résultats à MWIS.

Graphes $(P_5, K_{1,m})$ -libres (m fixé)

Si $m < 3$, la solution est triviale; si $m = 3$, G est sans griffe et l'on peut bien sûr appliquer l'un des algorithmes mentionnés précédemment pour cette classe (mais nous verrons qu'il en existe des plus efficaces pour des sur-classes des graphes (P_5, griffe) -libres).

Théorème 2.27 (Mosca (1997, 2004)) *On sait calculer un stable maximum d'un graphe $(P_5, K_{1,m})$ -libre ($m > 3$) d'ordre n en temps $\mathcal{O}(n^m)$.*

Démonstration. En reprenant les notations précédentes, on doit vérifier si $G[H(v, S)]$ contient un stable avec au moins $N_S(v)$ sommets. Or, comme G est sans $K_{1,m}$, $|N_S(v)| < m$. Donc on peut tester si v est augmentant en temps $\mathcal{O}(n^{m-1})$ et par conséquent, d'après le théorème 2.26, calculer un stable maximum de G en temps $\mathcal{O}(n^m)$. \square

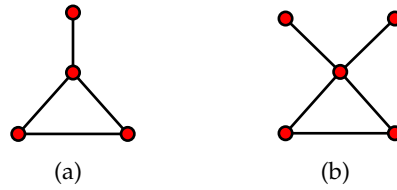


FIGURE 2.6 – (a) La patte et (b) le grillon.

Graphes $(P_5, \text{grillon})$ -libres

Une **patte**²⁵ *paw* et un **grillon** *cricket* sont les graphes obtenus respectivement à partir d'un $K_{1,3}$ et d'un $K_{1,4}$ en ajoutant une arête entre deux sommets de degré 1 (figure 2.6); par conséquent, tout graphe sans patte est aussi sans grillon. Par ailleurs, le théorème d'Alekseev (théorème 2.10) prouve immédiatement que le problème du stable maximum est \mathcal{NP} -difficile dans la classe des graphes sans patte et dans la classe des graphes sans grillon. Arbib et Mosca [14] ont prouvé qu'on peut résoudre MIS en $\mathcal{O}(n^2)$ dans les graphes (P_5, patte) -libres, résultat que Mosca [282] a donc étendu à la classe des graphes $(P_5, \text{grillon})$ -libres.

Théorème 2.28 (Mosca (1997)) *On sait calculer un stable maximum d'un graphe $(P_5, \text{grillon})$ -libre d'ordre n en temps $\mathcal{O}(n^3)$.*

Démonstration. Mosca a tout d'abord noté le fait suivant :

Lemme 2.2 *Soient S un stable maximal d'un graphe $(P_5, \text{grillon})$ -libre, et $v \in V \setminus S$. Le graphe $G[H(v, S)]$ est sans P_4 .*

Or on peut déterminer un stable maximum dans un graphe sans P_4 en temps $\mathcal{O}(m)$, par exemple en utilisant la décomposition modulaire. Le théorème 2.26 conduit alors au résultat. \square

Graphes $(P_5, K_{2,3})$ -libres

Enfin, Mosca a donné un résultat semblable pour la classe des graphes $(P_5, K_{2,3})$ -libres :

Théorème 2.29 (Mosca (1997)) *On sait calculer un stable maximum d'un graphe $(P_5, K_{2,3})$ -libre d'ordre n en temps $\mathcal{O}(n^3)$.*

Démonstration. Pour prouver ce théorème, Mosca commence par démontrer le lemme suivant :

Lemme 2.3 *Soient S un stable maximal d'un graphe $(P_5, K_{2,3})$ -libre, $v \in V \setminus S$ et T un stable de $G[H(v, S)]$. Si G ne contient pas de sommet S -augmentant, alors $|T| \leq 2$.*

On en déduit qu'on peut calculer un stable maximum de $G[H(v, S)]$ en temps $\mathcal{O}(n^2)$. Le théorème 2.26 permet alors de conclure. \square

Ce dernier résultat a été amélioré par Brandstädt et Lozin [49] qui, par une approche basée sur la *réduction de voisinage* (cf. section 2.5.6), ont

25. On lit parfois *patte d'oie*.

donné algorithmique en $\mathcal{O}(mn)$. La solution au problème pondéré n'a été apportée que très récemment par Mosca [285], qui, en utilisant la décomposition modulaire, a donné un algorithme de complexité $\mathcal{O}(n^2m)$.

Nous donnons à présent une liste d'autres classes pour lesquelles un algorithme polynomial est connu pour MIS.

Graphes $(P_5, \text{drapeau, chaise})$ -libres

Lozin [252] a démontré que tous les bipartis de cette classe sont soit des bipartis complets, soit un P_4 . Il s'ensuit que les graphes augmentants $(P_5, \text{drapeau, chaise})$ -libres sont bipartis complets. Ce résultat a permis ensuite à Lozin de résoudre MIS dans la classe des graphes $(P_5, \text{drapeau})$ -libres, en utilisant une technique proche de la structure (cf. section 2.5.6).

Graphes (P_5, D_m) -libres (m fixé)

Un D_m est obtenu à partir d'un mK_2 auquel on ajoute un sommet universel. Pour $m = 1$, il s'agit donc d'un triangle, pour $m = 2$, d'un papillon, et pour $m \leq 3$, un D_m contient un grillon. De manière générale, les graphes (P_5, D_m) -libres généralisent aussi évidemment les graphes $(P_5, K_{1,m})$ -libres. Gerber et Lozin [155, 159] ont montré que dans un graphe (P_5, D_m) -libre, $G[H(v, S)]$ est sans $(m+1)K_2$. Or il est bien connu que les graphes sans mK_2 ont au plus $\mathcal{O}(n^{2m-2})$ stables maximaux [4, 131]; il est donc possible de trouver un stable maximum en $\mathcal{O}(n^{2m+1})$ (cf. section 2.1.4)²⁶. En combinant ces résultats avec le théorème 2.26, on en déduit qu'on peut résoudre MIS en temps $\mathcal{O}(n^{2m+4})$.

Graphes $(P_5, K_{3,3} - e)$ -libres

Gerber, Hertz et Schindl [158] ont donné une caractérisation alternative des graphes augmentants sans P_5 : à tout n -uplet d'entiers (d_1, d_2, \dots, d_n) tel que $d_1 \geq d_2 \geq \dots \geq d_n$, ils associent le graphe de chaîne $B_n(d_1, \dots, d_n) = (V_1, V_2; E)$ avec $V_1 = \{a_1, \dots, a_n\}$, $V_2 = \{b_1, \dots, b_{d_1}\}$ et $E = \{\{a_i, b_j\} \mid j \leq d_i\}$ ($\{a_1, b_1\}$ est donc une arête dominante). Par exemple, $B_2(3, 1)$ est isomorphe à la chaise, $B_2(3, 2)$ est isomorphe au drapeau et $B_2(3, 3)$ est isomorphe au biparti complet $K_{2,3}$; en particulier $B_n(d, \dots, d)$ est isomorphe au biparti complet $K_{n,d}$.

Proposition 2.15 (Gerber, Hertz et Schindl (2004)) *Un graphe augmentant connexe est sans P_5 si et seulement s'il est isomorphe à un graphe $B_n(d_1, \dots, d_n)$ avec $n < d_1$ et $d_n > 0$.*

Le lemme suivant fournit une information supplémentaire importante sur les graphes augmentants sans P_5 :

²⁶. Dans [45], les auteurs affirment que MIS est résolu en $\mathcal{O}(n^{m-2})$ dans [4]. Nous n'avons pas pu le vérifier; la plupart des articles disent simplement que le nombre de stables maximaux est borné par un polynôme.

Lemme 2.4 Soient S un stable d'un graphe sans P_5 et $B_n(d_1, \dots, d_n)$ un graphe S -augmentant. Si G ne contient pas de $K_{1,2}$ S -augmentant, alors $n > 1$ et $d_2 \geq d_1 - 1$.

Gerber, Hertz et Schindl ont ainsi pu caractériser les graphes augmentants des graphes sans P_5 ni $K_{3,3} - e$ (i.e. un $K_{3,3}$ auquel on a retiré une arête), et ainsi généraliser à la fois les graphes $(P_5, \text{drapeau})$ -libres et les graphes $(P_5, K_{2,3})$ -libres :

Théorème 2.30 (Gerber, Hertz et Schindl (2004)) Soit S un stable d'un graphe sans $(P_5, K_{3,3} - e)$ -libre. Si G ne contient pas de $K_{1,2}$ S -augmentant, alors tout graphe S -augmentant connexe minimal est soit un biparti complet $K_{n,d}$ soit un $B_n(d, d - 1, \dots, d - 1)$ avec $1 < n < d$.

Le graphe $B_n(d, d - 1, \dots, d - 1)$ s'obtient en connectant un nouveau sommet à un sommet de degré $d - 1$ d'un biparti $K_{n,d-1}$; il est noté plus simplement $K_{n,d-1}^+$. L'article de Gerber, Hertz et Schindl se termine sur le fait qu'il n'existe pas d'algorithme polynomial pour détecter ces graphes augmentants. La solution a été apportée récemment par Lozin et Mosca [257] en utilisant les arguments présentés dans la classe qui suit. Notons que si l'on sait, ainsi, résoudre MIS dans un graphe $(P_5, K_{3,3} - e)$ -libre, on ne dispose pas, à notre connaissance, d'algorithme pour la version pondérée.

Graphes $(P_5, K_{2,m} - e)$ -libres (m fixé)

A partir d'un graphe G et d'un stable maximal S , Boliac et Lozin [32] construisent un graphe auxiliaire Γ dans lequel un stable de poids maximum correspond à un graphe sans P_4 S -augmentant maximum dans G . Or, si G est sans P_5 , Γ a une structure très particulière (il est notamment parfait, mais aussi sans $K_{2,3}$ ni drapeau) et il est donc possible de trouver un stable de poids maximum en temps polynomial dans ce graphe. Par conséquent :

Théorème 2.31 (Boliac et Lozin (2003)) Le problème du stable maximum dans les graphes sans P_5 est polynomialement équivalent au problème de la recherche de graphes augmentants contenant un P_4 .

Ils se sont ensuite intéressés à la classe des graphes $(P_5, K_{2,m} - e)$ -libres. Pour $m \leq 3$, les résultats étaient déjà connus ($K_{2,3} - e$ est isomorphe au drapeau) ; pour $m \geq 4$, cette classe généralise donc les graphes $(P_5, K_{2,3})$ -libres ainsi que les graphes $(P_5, \text{drapeau})$ -libres. Boliac et Lozin ont montré que dans cette classe, les graphes augmentants minimaux sans P_4 et le graphe Γ ont des propriétés particulières qui rendent le problème du stable de poids maximum résoluble en temps polynomial, en temps $\mathcal{O}(n^{2m-2})$.

D'autres familles infinies et généralisations

Mosca [284] a formulé un théorème très fort, dont la preuve repose sur le théorème 2.26, qui permet de généraliser indéfiniment tous les résultats connus sur MWIS dans les graphes sans P_5 à de nouvelles classes de graphes sans P_5 . Ces classes sont construites de la façon suivante : à partir d'un graphe F , on définit le graphe $F^{(h)}$ en ajoutant $h + 1$ sommets v, s_1, \dots, s_h tels que $\{s_1, \dots, s_h\}$ est stable et complet à V_F , et v est adjacent seulement à $\{s_1, \dots, s_h\}$ (par exemple, $K_2^{(1)}$ est isomorphe à la patte). Le théorème de Mosca s'énonce alors ainsi :

Théorème 2.32 (Mosca (2004)) *Soit F un graphe. Si l'on sait résoudre MWIS en temps $\mathcal{O}(n^d)$ dans les graphes (P_5, F) -libres, alors on peut le résoudre en temps $\mathcal{O}(n^{d+2})$ dans les graphes $(P_5, F^{(1)})$ -libres.*

Par exemple, nous verrons par la suite qu'il est possible de résoudre MWIS en $\mathcal{O}(n + m)$ dans la classe des graphes $(P_5, \text{diamant}^{27})$ -libres. Pour les graphes denses (i.e. ayant beaucoup d'arêtes), $m = \mathcal{O}(n^2)$. Le théorème précédent nous garantit donc que pour tout $p \geq 4$ fixé, on peut calculer un stable de poids maximum dans un graphe (P_4, K_p) -libre en temps $\mathcal{O}(n^{2p-4})$ et dans un graphe $(P_4, K_p - e)$ -libre en temps $\mathcal{O}(n^{2p-6})$.

Remarque 2.1

Ainsi, on peut déterminer un stable maximum et un stable de poids maximum en temps $\mathcal{O}(n^4)$ dans les graphes (P_5, K_4) -libres, qui contiennent évidemment les graphes P_5 3-colorables ; nous verrons cependant au chapitre 3 que pour cette dernière classe, nous avons réussi à exhiber un algorithme qui résout MWIS en temps $\mathcal{O}(n + m)$.

Bien que l'on soit assuré d'avoir un algorithme polynomial, nous voyons que la complexité de celui-ci augmente rapidement ; de plus, elle dépend de la complexité des classes définies par de plus petits graphes interdits. Il est donc important de connaître la complexité du meilleur algorithme résolvant M(W)IS dans une classe, même si celle-ci est généralisé par une autre pour laquelle on dispose encore d'un algorithme polynomial. Mosca a d'ailleurs montré que si l'on se restreint à MIS, on peut abaisser un peu la complexité donnée dans le théorème précédent :

Théorème 2.33 (Mosca (2004)) *Soit F un graphe. Si l'on sait résoudre MIS en temps $\mathcal{O}(n^d)$ dans les graphes (P_5, F) -libres, alors on peut le résoudre en temps $\mathcal{O}(n^{d+h+2})$ dans les graphes $(P_5, F^{(h)})$ -libres ($h \geq 2$).*

Si F est stable, de cardinalité $k - 1$, on peut trouver un stable maximum dans un graphe (P_5, F) -libre en temps $\mathcal{O}(n^{k-2})$ (par exemple en regardant de manière exhaustive tous les ensembles de $k - 2$ sommets). Or, $F^{(h)}$ est alors le graphe biparti complet $K_{h,k}$. On peut donc calculer un stable maximum dans tout biparti complet $K_{n,m}$ en temps $\mathcal{O}(n^{n+m})$.

27. cf. figure 2.11 (d)

Nous terminons cette section avec un autre type de généralisation infinie, similaire, proposée par Gerber, Hertz et Schindl [158] : à partir d'un graphe H et d'un entier $t \geq 0$, on construit le graphe $A(t, H)$ en ajoutant une clique $K = \{k_1, \dots, k_t\}$ et un stable $S = \{s_1, \dots, s_t\}$ tels que K est complet à H et k_i est adjacent à l_j si et seulement si $i \leq j$. Notons que pour $t = 1$, on retrouve le graphe $F^{(1)}$ défini par Mosca ; or, par construction, $A(t, H) = A(1, A(t - 1, H))$, d'où :

Théorème 2.34 (Mosca ; Gerber, Hertz et Schindl (2004)) *Soit H un graphe. Si l'on sait résoudre MWIS en temps polynomial dans les graphes (P_5, H) -libres, alors on peut le résoudre en temps polynomial dans les graphes $(P_5, A(t, H))$ -libres pour tout $t \geq 0$.*

2.4 LA DÉCOMPOSITION MODULAIRE

La décomposition modulaire se présente comme l'un des outils majeurs de la théorie des graphes. L'idée naturelle de toute décomposition de graphe est de regrouper des sommets (ou des sous-graphes) qui possèdent des propriétés similaires. Dans le cas présent, nous nous intéressons aux *modules* d'un graphe, que l'on peut voir comme une généralisation des composantes connexes. Mais au contraire de ces dernières, un module peut être un sous-ensemble propre d'un autre module, ce qui conduit à une décomposition arborescente, plutôt qu'une simple partition. Si la décomposition modulaire a permis de mettre au point des algorithmes très efficaces pour reconnaître certaines classes de graphes, tels les cographes [98], pour trouver une orientation transitive dans un graphe de comparabilité [272, 273], ou encore pour améliorer le tracé de graphes [293], nous verrons qu'elle a été également très fructueuse pour résoudre le problème du stable maximum dans plusieurs classes de graphes sans P_5 .

2.4.1 Premières définitions et propriétés

Module

On s'accorde généralement à attribuer la paternité de la décomposition modulaire à Gallai, qu'il a décrite dans [148]²⁸ alors qu'il étudiait les graphes de comparabilité (cf. section 2.1.4). Cependant l'idée de module a été redécouverte indépendamment à différentes époques et dans des domaines différents. On trouvait ainsi les appellations *closed sets* (traduction anglaise de l'allemand), *stable sets* (traduction anglaise du russe), *clumps*, *partitive subsets* [320], *clans*, *autonomous sets*, *homogenous sets* ... avant que le terme *module* ne s'impose.

Définition 2.9 *On appelle **module** (module) d'un graphe $G = (V, E)$ un ensemble de sommets $M \subseteq V$ tel que $\forall x, y \in M, N(x) \setminus M = N(y) \setminus M$.*

²⁸. Cet article, publié en 1967, a été rédigé en allemand. Il a fallu attendre plus de trente ans pour qu'une traduction en anglais soit effectuée, par Maffray et Preissmann [266], dans une version légèrement modernisée.

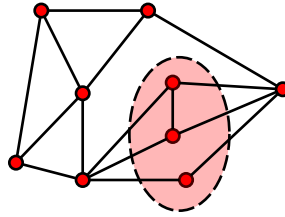


FIGURE 2.7 – Exemple de module dans un graphe, constitué des sommets à l'intérieur de l'ellipse.

Autrement dit, un module M est un ensemble de sommets ayant le même voisinage à l'extérieur de M . On a la définition équivalente suivante :

Définition 2.10 Un ensemble $M \subseteq V$ de sommets est un module d'un graphe $G = (V, E)$ si pour tout $v \notin M$, $M \subseteq N(v)$ ou $M \cap N(v) = \emptyset$.

c'est-à-dire que tout sommet $w \notin M$ est soit complet, soit anticomplet à M . La figure 2.7 donne un exemple de module. Deux sommets d'un même module sont dits **jumeaux** (vrais jumeaux s'ils sont adjacents, faux jumeaux dans le cas contraire).

Théorème 2.35 Tout graphe possède $\mathcal{O}(n)$ modules.

Propriétés des modules

Les propriétés suivantes découlent directement de la définition d'un module :

Proposition 2.16 Soit M un module de $G = (V, E)$. Alors :

- M est aussi un module de \overline{G} ;
- si M' est aussi un module, alors $M \cap M'$ est un module ; de plus, si $M \cap M' \neq \emptyset$, $M \cup M'$ est un module ;
- $\forall V' \subseteq V$ tel que $M \cap V' \neq \emptyset$, $M \cap V'$ est un module de $G[V']$;
- $V' \subseteq M$ est un module de G si et seulement si c'est un module de $G(M)$;
- si M' est un module disjoint de M , M' est soit complet soit anticomplet à M .

Contraction et substitution

Tous les sommets d'un module se comportent de la même façon vis-à-vis de l'extérieur. On peut donc se représenter un module comme un unique « gros sommet ». Il est d'ailleurs facile de voir que si M est un module d'un graphe $G = (V, E)$, tous les graphes $G[V \setminus (M \setminus \{v\})]$ ($v \in M$) sont isomorphes.

Définition 2.11 L'opération consistant à remplacer un module M d'un graphe G par un unique sommet $v \in M$ est appelée **contraction** de M . Le graphe résultant est noté G_M^v .

L'opération inverse est appelée *substitution* :

Définition 2.12 Étant donnés deux graphes $G = (V_G, E_G)$ et $H = (V_H, E_H)$, et $v \in V_G$, la **substitution** de v par H est le graphe noté G_v^H dont :

- les sommets sont les éléments de $(V_G \setminus \{v\}) \cup V_H$;
- les arêtes sont les éléments de $(E_G \setminus \{v, w\} | w \in N(v)) \cup E_H \cup \{\{u, w\} : u \in V_H \text{ et } w \in N(v)\}$

Évidemment, V_H est un module de G_v^H , et si M est un module de G , alors $G = (G_M^v)^{G[M]}$.

Proposition 2.17 Si v_1, \dots, v_k sont des sommets de G , distincts deux à deux, et H_1, \dots, H_k sont des graphes, alors l'ordre de substitution des v_i par les H_i n'a pas d'importance, et on note $G_{v_1 \dots v_k}^{H_1 \dots H_k}$ le graphe obtenu.

Module trivial, graphe premier, module fort et graphe quotient

Remarquons que, comme signalé dans l'introduction de cette section, les composantes connexes de G et de \bar{G} sont des modules. En fait, par définition tout sommet est lui-même un module.

Définition 2.13 Dans un graphe $G = (V, E)$, l'ensemble vide, l'ensemble V et chacun des singletons constitués des éléments de V sont appelés **modules triviaux** de G . Un **ensemble homogène** (homogeneous set) est un module non trivial.

Définition 2.14 Un graphe est dit **premier** (prime graph) s'il ne contient que des modules triviaux.

Par exemple, un P_5 est premier.

Définition 2.15 Un module non trivial M est **fort** (strong module) dans un graphe G s'il ne chevauche aucun autre module, i.e. \forall module M' de G , on a soit $M \cap M' = \emptyset$, soit $M \subseteq M'$, soit $M' \subseteq M$.

Selon cette définition, les modules triviaux sont forts. Les modules forts *maximaux* (i.e. non contenus dans un autre module fort) sont particulièrement intéressants :

Proposition 2.18 *Les modules forts maximaux (pour l'inclusion) d'un graphe $G = (V, E)$ induisent une partition unique de V , appelée **partition modulaire maximale**.*

Définition 2.16 *Soient G un graphe, et M_1, \dots, M_k ses modules forts maximaux. Le **graphe quotient** de G est le graphe $G_{M_1 \dots M_k}^{m_1 \dots m_k}$ ($m_i \in M_i, 1 \leq i \leq k$).*

Naturellement, un graphe quotient est également premier (s'il ne l'était pas, cela signifierait que l'un des M_i n'était pas maximal).

2.4.2 Théorème de décomposition modulaire

Rappelons que si un graphe G n'est pas connexe, son complémentaire \bar{G} est connexe.

Théorème 2.36 (Gallai (1967)) *Soit $G = (V, E)$ un graphe. Une et une seule des propriétés suivantes est vérifiée :*

- V ne contient qu'un seul sommet ;
- G n'est pas connexe ; ses modules forts maximaux sont ses composantes connexes, le graphe quotient associé est un stable, et l'on parle de partition de type parallèle ;
- \bar{G} n'est pas connexe ; les modules forts maximaux de G sont les composantes connexes de \bar{G} , le graphe quotient associé est une clique, et l'on parle de partition de type série ;
- si G et \bar{G} sont connexes, les modules forts maximaux de G sont ses modules maximaux ; le graphe quotient associé est premier, et l'on parle de partition de type premier.

Naturellement, un module étant un sous-graphe, il peut à son tour être partitionné en « sous-modules » et ainsi de suite. La **décomposition modulaire** est l'application récursive de ce théorème, jusqu'à ce qu'il n'y ait que des graphes à un seul sommet.

2.4.3 Arbre de décomposition modulaire

On déduit immédiatement du théorème précédent une (unique) structure hiérarchique arborescente des modules forts d'un graphe, appelée **arbre de décomposition modulaire**²⁹.

Chaque sommet³⁰ de l'arbre un graphe à décomposer, et est étiqueté selon le type de partition donné par le théorème de décomposition modulaire. Ainsi, la racine est associée au graphe tout entier, et les feuilles correspondent aux sommets du graphe. Afin de fixer les idées, nous allons détailler les étapes de la décomposition modulaire sur le graphe suivant :

29. En toute rigueur, on devrait plutôt parler d'*arborescence de décomposition modulaire*, puisque l'on distingue un sommet particulier et que l'on met en évidence une certaine orientation dans la structure.

30. On parle plus souvent de *nœud*, lorsqu'il s'agit d'arbres.

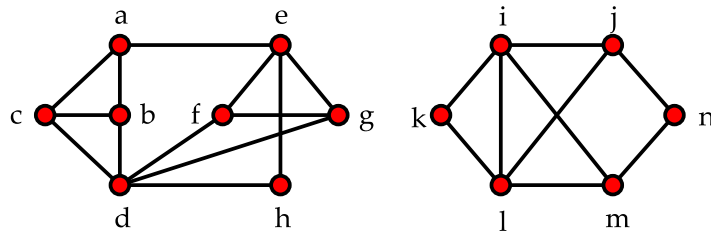


FIGURE 2.8 – Un graphe à décomposer.

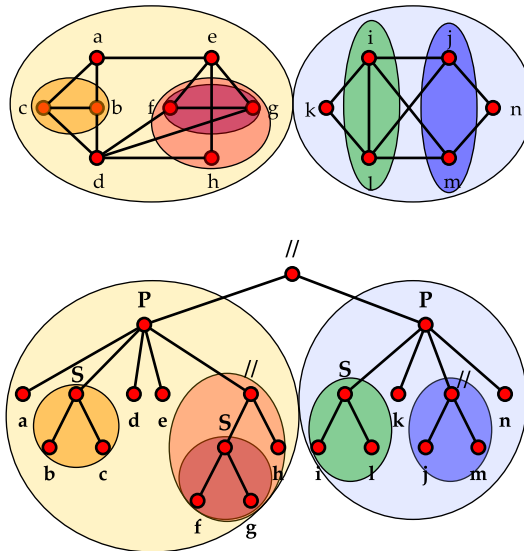


FIGURE 2.9 – Décomposition modulaire du graphe de la figure 2.8 et arbre correspondant.

Ce graphe n'est pas connexe. La racine de l'arbre est donc étiquetée *parallèle*, et a deux fils correspondant aux deux composantes connexes $\{a, b, c, d, e, f, g, h\}$ et $\{i, j, k, l, m, n\}$.

Considérons tout d'abord le sous-graphe induit par $V_1 = \{a, b, c, d, e, f, g, h\}$. Il est connexe, ainsi que son complémentaire. Ses modules maximaux différents de V_1 sont $\{a\}$, $\{b, c\}$, $\{d\}$, $\{e\}$ et $\{f, g, h\}$. Le graphe quotient associé est un C_5 , qui est premier, et le nœud correspondant est lui aussi étiqueté *premier* et a donc cinq fils. Trois d'entre eux seront des feuilles (associées aux sommets a , d , et e); un autre est associé au sous-graphe induit par $\{b, c\}$, dont le complémentaire n'est pas connexe et qui est donc étiqueté *série*; le dernier est étiqueté *parallèle* (il correspond au sous-graphe induit par $\{f, g, h\}$, qui n'est pas connexe). Enfin, le sous-graphe induit par $\{f, g\}$ est décomposé selon une partition *série*.

La deuxième composante connexe du graphe est décomposée de la même façon. Finalement, nous obtenons l'arbre de décomposition de la figure 2.9.

Le premier algorithme permettant d'obtenir l'arbre de décomposition modulaire d'un graphe a été publié en 1972 [220]. Beaucoup d'algorithmes ont ensuite été proposés, de complexité toujours meilleure, mais le plus souvent difficiles à mettre en œuvre et plutôt d'un intérêt théorique. Un algorithme linéaire, et relativement simple, a été proposé récemment par Tedder, Corneil, Habib et Paul [331]. Signalons que la décomposition modulaire d'un graphe orienté peut également être obtenue en temps linéaire [271].

2.4.4 Application aux cographes

Il a été plusieurs fois question jusqu'ici des *cographes*, un synonyme pour désigner les graphes sans P_4 , et qui constituent donc naturellement une sous-classe propre des graphes sans P_5 . Avant de montrer comment la décomposition modulaire permet de résoudre de manière particulièrement simple et élégante plusieurs problèmes d'optimisation (dont MIS) dans ces graphes, intéressons-nous à leurs propriétés.

Propriétés des cographes

Les premiers travaux relatifs aux graphes sans P_4 remontent aux années 1960 et sont liés aux graphes de comparabilité (Wolk [345, 346] a en effet montré que les graphes (P_4, C_4) -libres, qu'il appelle D-graphes³¹, sont les graphes de comparabilité des arborescences³²). Ils ont ensuite été redécouverts par de nombreux chercheurs dans des domaines variés des mathématiques, ce qui explique qu'on trouvait parfois les noms de *hereditary Dacey graphs* [321], *D*-graphes* [224] ou encore *graphes de 2-parité* [61]. Parmi les graphes sans P_4 , on trouve par exemple les bipartis, les bipartis complets, ou encore les graphes à seuil.

Au début des années 1970, Lerchs [244, 245] a introduit une classe de graphes \mathcal{C} définie récursivement à l'aide des trois règles suivantes :

1. un graphe à un seul sommet appartient à \mathcal{C} ,
2. si $G \in \mathcal{C}$, alors $\overline{G} \in \mathcal{C}$ (autocomplémentarité),
3. si $G \in \mathcal{C}$ et $H \in \mathcal{C}$, alors $G \cup H \in \mathcal{C}$.

Les graphes de la classe \mathcal{C} sont appelés **cographes** (pour *complement-reducible graphs*). Stewart [319] a étudié la représentation arborescente de ces graphes, et a développé un algorithme de reconnaissance en temps $\mathcal{O}(n^2)$. Corneil, Lerchs et Stewart [97] ont ensuite prouvé qu'un graphe est un cographe si et seulement s'il est sans P_4 , et ont donné d'autres caractérisations de cette classe :

Proposition 2.19 (Corneil, Lerchs et Stewart (1981)) *Soit G un graphe. Les propriétés suivantes sont équivalentes :*

- G est sans P_4 ,
- G est un cographe,
- tout sous-graphe induit de G a la propriété que tout stable maximal intersecte toute clique maximale en (exactement) un sommet,
- tout sous-graphe induit non trivial possède au moins deux sommets jumeaux [320],
- tout sous-graphe induit connexe de G a un complémentaire non connexe [245, 312].

31. Pour *diagonal graphs*, car ils possèdent la « propriété diagonale », à savoir que si un tel graphe contient quatre sommets a, b, c, d avec les arêtes $\{a, b\}, \{b, c\}, \{c, d\}$, il contient aussi l'arête $\{a, c\}$ ou l'arête $\{b, d\}$.

32. Dans [16], les auteurs affirment de manière erronée que « les graphes sans P_4 sont les graphes de comparabilité des arborescences » ; or, ce n'est pas le cas de C_4 .

Coarbre

En particulier, les deux dernières propriétés impliquent immédiatement que l'arbre de décomposition modulaire d'un cographe, appelé **coarbre** (*cotree*)³³, ne contient que des nœuds étiquetés *parallèle* ou *série* (on trouve parfois ces nœuds étiquetés respectivement 0 et 1) :

- un nœud parallèle correspond à l'union des sous-graphes définis par ses fils ;
- un nœud série correspond au *joint* des sous-graphes définis par ses fils (c'est-à-dire que l'on ajoute une arête entre chaque paire de sommets appartenant à des fils différents).

La figure ci-dessous donne un exemple de coarbre.

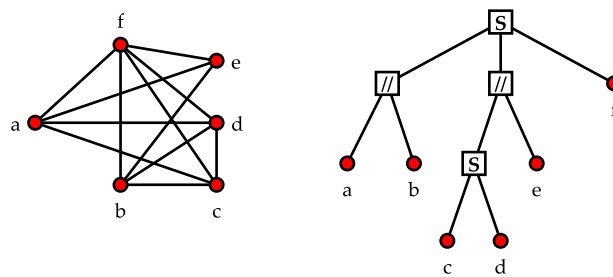


FIGURE 2.10 – Un cographe et son arbre.

Les propriétés suivantes sont faciles à établir :

Proposition 2.20

- Le coarbre d'un cographe connexe a pour racine un nœud série ;
- deux sommets sont adjacents si et seulement si leur plus proche ancêtre commun est un nœud série ;
- si on impose que les nœuds de tout chemin de la racine à une feuille soient étiquetés alternativement série et parallèle, le coarbre est unique [97].

Corneil, Perl et Stewart [98] ont montré qu'il est possible de reconnaître un cographe et de construire son coarbre en temps linéaire ($\mathcal{O}(n + m)$) en utilisant la décomposition modulaire. D'autres techniques ont été utilisées depuis, comme l'affinage de partition [182], l'algorithme de parcours en largeur lexicographique [53] ou la décomposition en splits [161].

Notons enfin qu'on peut construire le complémentaire d'un cographe en temps linéaire (ce qui peut demander $\mathcal{O}(n^2)$ pour un graphe quelconque) : par autocomplémentarité, le complémentaire d'un cographe G est lui-même un cographe, et son coarbre s'obtient simplement en inversant les étiquettes *parallèle* et *série* du coarbre de G .

33. Attention, dans un autre contexte, le coarbre peut également désigner le complémentaire d'un arbre couvrant d'un graphe !

Résolution de problèmes d'optimisation combinatoire à l'aide des coarbres

Lorsque l'on dispose du coarbre, plusieurs problèmes classiques en théorie des graphes peuvent être résolus par un simple parcours en profondeur. Par exemple (un p (resp. s) en indice signifie que le calcul se rapporte à un nœud parallèle (resp. série)) :

– *stable maximum* : par définition du coarbre, pour tout sous-graphe associé à un nœud *parallèle*, le stable maximum correspond à l'union des stables maximum calculés pour chacun des fils, et pour un nœud *série* il correspond simplement à un stable de cardinalité maximale parmi les stables calculés pour ses fils. Autrement dit, en commençant par attribuer la valeur 1 à chaque feuille, $\alpha_p = \sum \alpha_i$ et $\alpha_s = \max\{\alpha_i\}$ où les α_i sont les valeurs des fils du nœud considéré. On en déduit évidemment immédiatement un transversal minimum. Il est également très facile d'énumérer tous les stables maximaux, en multipliant les valeurs des fils d'un nœud parallèle et en les sommant pour un nœud série.

Exemple 2.4 En appliquant ce principe au graphe G de la figure 2.10, on trouve quatre stables maximaux : $\{a, b\}$, $\{c, e\}$, $\{d, e\}$ et $\{f\}$. Les stables maximum sont $\{a, b\}$, $\{c, e\}$ et $\{d, e\}$, d'où $\alpha(G) = 2$, et $\tau(G) = 4$.

– *clique maximum* : il suffit d'inverser les opérations décrites pour le calcul du stable maximum : $\omega_p = \max\{\omega_i\}$ et $\omega_s = \sum \omega_i$.

Exemple 2.5 Pour graphe G de la figure 2.10, on trouve deux cliques maximum : $\{a, c, d, f\}$, et $\{b, c, d, f\}$ et $\omega(G) = 4$.

– *coloration* : comme un nœud parallèle regroupe des composantes connexes, il est facile de voir que $\chi_p = \max\{\chi_i\}$; de même, $\chi_s = \sum \chi_i$: on retrouve les mêmes formules que pour le calcul de la clique maximum, ce qui n'a rien d'un hasard puisque les cograves sont parfaits. Pour obtenir une coloration, on initialise un compteur à la valeur 1 et on effectue un parcours en profondeur du coarbre; quand une feuille est visitée, elle reçoit la valeur du compteur et lorsqu'on rencontre un nœud série en remontant vers la racine, on incrémente le compteur.

Exemple 2.6 On peut colorer le graphe de la figure 2.10 avec 4 couleurs : les sommets a et b ont la couleur 1, puis on remonte à la racine, qui est un nœud série, ce qui a pour effet d'incrémenter le compteur. Le sommet c est donc colorié avec la couleur 2, puis le compteur est de nouveau incrémenté; le sommet d reçoit alors la couleur 3, ainsi que le sommet e . La racine est de nouveau visitée, et on termine en attribuant la couleur 4 au sommet f .

– *dominant minimum* : on a facilement $\gamma_p = \sum \gamma_i$; pour les nœuds série, il suffit de voir qu'un dominant dans un des fils domine également les autres fils, d'où $\gamma_s = \min \gamma_i$.

Exemple 2.7 Pour le graphe G de la figure 2.10, on trouve bien sûr $\gamma(G) = 1$ puisque le sommet f est dominant.

– *cycle hamiltonien* : le problème du **cycle hamiltonien** (ou de l'*hamiltonicité*

d'un graphe) consiste à savoir s'il est possible, en longeant les arêtes d'un graphe, de visiter tous les sommets exactement une fois. Il s'agit d'un problème \mathcal{NP} -difficile, même dans les graphes planaires cubiques [151, 152]. Une condition *nécessaire* pour qu'un graphe $G = (V, E)$ soit hamiltonien est que $c(G[V \setminus S]) \leq |S|$, où $S \subset V$ ($S \neq \emptyset$) et $c(G)$ désigne le nombre de composantes connexes de G . Jung [224] a appelé **nombre de dispersion** de G (*scattering number*) le nombre $s(G) = \max\{c(G[V \setminus S]) - |S|\}$. Par conséquent, si G est hamiltonien, $s(G) \leq 0$. Jung a montré que si G est un cografe d'ordre au moins 3, cette condition est aussi suffisante. Il suffit donc de calculer le nombre de dispersion d'un cografe pour tester son hamiltonicité; Corneil, Lerchs et Stewart [97] ont donné les formules correspondantes : attribuer à chaque feuille une valeur $a_i = -1$ et $b_i = 1$, et pour les nœuds internes appliquer les formules $a_p = \sum \max\{a_i, 1\}$, $b_p = \sum b_i$ et $a_s = \max\{a_i - \sum_{j \neq i} b_j\}$, $b_s = \sum b_i$ (autrement dit, a_i représente le nombre de dispersion et b_i le nombre de sommets dans le graphe associé au nœud).

Exemple 2.8 Pour le graphe G de la figure 2.10, on trouve finalement un nombre de dispersion égal à -1 . Il correspond à l'ensemble $\{a, b, f\}$, de cardinal 3 dont la suppression laisse deux composantes connexes : $\{c, d\}$ et $\{e\}$, ce qui prouve que le graphe est hamiltonien.

Ces formules ne permettent pas d'obtenir un chemin ou un cycle hamiltonien de manière aussi immédiate que l'on résolvait les autres problèmes, mais Lin, Olariu et Pruesse [247] ont donné un autre algorithme, constructif cette fois, qui s'exécute aussi en temps linéaire.

-isomorphisme : d'après la proposition 2.20, tout cografe possède une représentation unique sous forme de coarbre « normalisé ». L'isomorphisme de cografes se ramène donc (en temps linéaire) au problème de l'isomorphisme d'arbres, pour lequel il existe un algorithme linéaire [217, 218] (qui s'applique en fait à n'importe quel graphe planaire).

Naturellement, les versions pondérées du stable maximum, de la clique maximum, de la coloration minimum et du dominant minimum peuvent être résolues en temps linéaire dans les cografes, en apportant des modifications très simples aux algorithmes précédents. Yu et Yang [350] ont donné un algorithme permettant de calculer un couplage maximum en temps linéaire; bien qu'on sache qu'il est possible de trouver un couplage de poids maximum ou un cycle hamiltonien de poids maximum en temps polynomial dans les cografes, on ignore encore s'il est possible d'y parvenir en temps linéaire. Enfin, tous les problèmes habituellement difficiles à résoudre ne sont pas forcément plus simples dans les cografes : ainsi le problème de l'isomorphisme de sous-graphe reste-t-il \mathcal{NP} -difficile, que le sous-graphe soit induit [105] ou non [1].

2.4.5 Graphes sans P_5 de largeur de clique bornée

Courcelle, Engelfriet et Rozenberg [101] ont défini la notion de **largeur de clique** *clique width* d'un graphe étiquetés G , comme le nombre minimum

d'étiquettes nécessaires à la génération de G à partir des trois opérations suivantes :

- union disjointe,
- ajout d'arêtes entre tous les sommets étiquetés par i et tous les sommets étiquetés par j pour $i \neq j$,
- remplacement de l'étiquette i par l'étiquette j .

Par exemple, la largeur de clique des cographes est au plus 2 [103]. Une k -expression d'un graphe G de largeur de clique k décrit la génération récursive de G par l'application répétée des trois opérations, en utilisant seulement un ensemble d'au plus k étiquettes. La largeur de clique est intimement liée à la décomposition modulaire (cf. [103]) :

Proposition 2.21 *La largeur de clique d'un graphe est égale au maximum de la largeur de clique de ses sous-graphes premiers, et la largeur de clique de \overline{G} est au plus la largeur de clique de G .*

Pour reprendre l'exemple précédent, une 2-expression d'un cographe peut être trouvée en temps linéaire en utilisant le coarbre. L'intérêt pour le concept de largeur de clique est devenu considérable lorsque Courcelle, Makowsky et Rotics [102] ont montré le théorème suivant :

Théorème 2.37 (Courcelle, Makowsky, Rotics (2000)) *Tout problème algorithmique exprimable dans une certaine logique monadique du second ordre appelée $\text{LinEMSOL}(\tau_{1,L})$ peut être résolu (de manière constructive) en temps $\mathcal{O}(f(n, m))$ pour toute classe de graphes de largeur de clique bornée par un entier k pour laquelle une k -expression peut être construite en temps $\mathcal{O}(f(n, m))$.*

car il se trouve que les problèmes du stable de poids maximum, de la clique de poids maximum et du dominant de poids minimum (entre autres) sont justement exprimables dans $\text{LinEMSOL}(\tau_{1,L})$.

De plus, de nombreuses classes de graphes sans P_5 ont une structure qui conduit à une largeur de clique bornée, et à des algorithmes qui donnent une k expression en temps linéaire :

- graphes (P_5 , gemme³⁴, co-gemme³⁵)-libres [39],
- graphes (P_5 , gemme, drapeau)-libres [39],
- graphes (P_5 , gemme, taureau)-libres [39],
- graphes (P_5 , gemme, chaise)-libres [39],
- graphes (P_5 , gemme, cerf-volant)-libres [37],
- graphes (P_5 , maison, co-gemme)-libres [39],
- graphes (P_5 , maison, chaise)-libres [137],
- graphes (P_5 , maison, taureau)-libres [136],
- graphes (P_5 , drapeau, cerf-volant)-libres [51],
- graphes (P_5 , taureau, cerf-volant)-libres [41],

34. cf. figure 2.11 (c)

35. Co-gemme désigne le complémentaire d'une gemme.

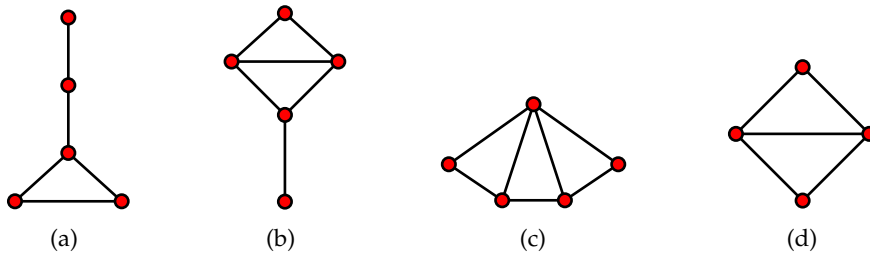


FIGURE 2.11 – (a) La 2-patte (b) le cerf-volant (c) la gemme (d) le diamant

– graphes (P_5 , maison, cerf-volant)-libres (graphes P_4 -épars)[51],

Attention toutefois : ce n'est pas parce qu'on peut résoudre MIS en temps polynomial dans une classe qu'elle est de largeur de clique bornée. Par exemple, la largeur de clique des graphes scindés, des bipartis ou des graphes (P_5 , taureau)-libres n'est pas bornée.

2.4.6 Application aux graphes sans P_5

Dans sa thèse de doctorat portant sur les graphes parfaits, Hoàng [208] a défini la classe des graphes P_4 -épars (P_4 -sparse graphs) : ce sont les graphes pour lesquels tout sous-graphe induit d'ordre 5 contient au plus un P_4 induit. Jamison et Olariu [221] ont montré qu'un graphe est P_4 -épars si et seulement s'il est (C_5 , P_5 , maison, drapeau, 2-patte³⁶, chaise, cerf-volant³⁷)-libre (cf. figure 2.11) ; il s'agit donc d'une sur-classe des co-graphes, qui a elle même été étendue ensuite de plusieurs manières :

Graphes (P_5 , taureau)-libres et graphes (P_5 , maison)-libres

Fouquet [136] a utilisé la décomposition modulaire pour étudier la structure des graphes (P_5 , maison, taureau)-libres. Il a notamment montré que tout graphe de cette classe d'ordre au moins 6 satisfait l'une des conditions suivantes :

- G contient un ensemble homogène,
- G contient un sommet universel,
- G est biparti,
- \overline{G} est biparti.

Naturellement, les graphes (P_5 , maison, taureau)-libres ont ensuite été généralisés d'une part à la classe des graphes (P_5 , taureau)-libres, et d'autre part à la classe des graphes (P_5 , maison)-libres.

Dans la première, il est possible de dériver de cette un algorithme polynomial pour MIS à partir d'une caractérisation par De Simone [110] des graphes premiers (chaise, taureau)-libres qui contiennent le complémentaire d'un diamant. Quant à la classe des graphes (P_5 , maison)-libres, elle a été amplement étudiée par Giakoumakis et Rusu [160], pour laquelle ils ont donné un algorithme de complexité $\mathcal{O}(n(n+m))$ qui parcourt l'arbre de décomposition modulaire pour calculer un stable de poids maximum (et donc, par autocomplémentarité, aussi une clique de poids maximum).

36. Une 2-patte désigne le complémentaire d'un drapeau.

37. Le cerf-volant est le complémentaire de la chaise.

Cet algorithme distingue deux cas, selon qu'un graphe (P_5, maison) -libre contient un C_5 ou non :

- s'il ne contient pas de C_5 , Chvátal, Hoàng, Mahadev et de Werra [90] ont montré qu'il est possible de trouver un stable maximum et une coloration minimum (et les complémentaires) en $\mathcal{O}(n + m)$;
- s'il contient un C_5 , ce C_5 est lui-même contenu dans une **bouée** (*buoy*), c'est-à-dire un graphe obtenu en remplaçant chaque sommet d'un C_5 par un ensemble non vide, et cette bouée est soit constituée de tous les sommets, soit forme un ensemble homogène (cf. [138]).

Il n'est pas inintéressant de signaler que les graphes $(P_5, C_5, \text{maison})$ -libres sont *parfaitement ordonnables* (et donc parfaits, cf. section 1.1.8), comme l'ont montré Chvátal, Hoàng, Mahadev et de Werra [90] (puis plus simplement Hoàng, Maffray, Olariu et Preissmann [212]), car bien que Hoàng [210] ait donné un algorithme de complexité $\mathcal{O}(nm)$ pour trouver une clique de poids maximum dans cette classe, la question d'un algorithme *combinatoire* pour MIS est encore aujourd'hui ouverte. Par ailleurs, des extensions des graphes (P_5, maison) -libres ont été données par Zverovich et Zverovich [353] pour MIS.

Graphes semi- P_4 -épars

Fouquet et Giakoumakis [137] ont défini une autre généralisation des graphes P_4 -épars : ce sont les **graphes semi- P_4 -épars**, ou graphes $(P_5, \text{maison}, \text{cerf-volant})$ -libres. En utilisant la décomposition modulaire, ils ont proposé un algorithme de reconnaissance linéaire, et montré qu'avec des modifications mineures, il est possible de réutiliser les algorithmes linéaires de Chvátal, Hoàng, Mahadev et de Werra mentionnés ci-dessus pour résoudre les problèmes d'optimisation classiques ; enfin, ils ont justifié que tous ces résultats sont applicables aux complémentaires des graphes semi- P_4 -épars.

En fait, les graphes $(P_5, \text{maison}, \text{taureau})$ -libres et les graphes semi- P_4 -épars ont une largeur de clique bornée, donc on peut résoudre MWIS en temps *linéaire* dans ces classes. Ni les graphes $(P_5, \text{taureau})$ -libres, ni les graphes (P_5, maison) -libres n'ont cette propriété, donc les algorithmes mentionnés pour ces classes conservent toute leur importance.

Graphes $(P_5, \text{diamant})$ -libres

Arbib et Mosca [15] ont été les premiers à donner un algorithme polynomial, de complexité $\mathcal{O}(n^2)$, pour résoudre MWIS dans les graphes ne contenant ni P_5 ni diamant (cf. section 2.5.6). Un peu plus tard, Brandstädt [37] a prouvé à l'aide de la décomposition modulaire que le problème peut être résolu en temps linéaire ($\mathcal{O}(n + m)$), de deux manières différentes : soit en utilisant son algorithme, qui exploite l'arbre de décomposition modulaire, soit parce qu'il a prouvé que les graphes $(P_5, \text{diamant})$ -libres ont une largeur de clique bornée et qu'une k -expression peut être trouvée en temps linéaire (la notion de k -expression sera elle aussi explicitée à la section 2.7).

Graphes (P_5 , gemme)-libres

Les graphes P_4 -épars et les graphes (P_5 , diamant)-libres sont généralisés par les graphes (P_5 , gemme)-libres, pour lesquels le théorème 2.32, appliqué aux cographes, donne un algorithme de complexité $\mathcal{O}(n^4)$. Brandstädt et Kratsch [44] ont donné une description complète de leur structure, montrant entre autres que leurs ensembles homogènes sont justement des cographes, et ont également prouvé qu'ils ont une largeur de clique d'au plus 5 (cf. [46] pour une preuve détaillée). Bodlaender et al. [30, 31] ont utilisé la caractérisation de ces graphes et la décomposition modulaire pour élaborer un algorithme qui résout en temps linéaire les problèmes du stable, de la clique, de la coloration et de la couverture par cliques, y compris dans leurs versions pondérées, ainsi qu'un algorithme de reconnaissance quadratique. Notons que comme il est possible de reconstruire une 5-expression en temps polynomial, le problème du dominant minimum peut aussi être résolu en temps polynomial. Enfin, De Simone et Mosca [111] ont caractérisé le polytope des stables et le polytope des cliques des graphes (P_5 , gemme)-libres. Les résultats de [31] entraînent que les problèmes de séparation associés peuvent être résolus en temps polynomial.

Graphes (P_5 , cerf-volant)-libres

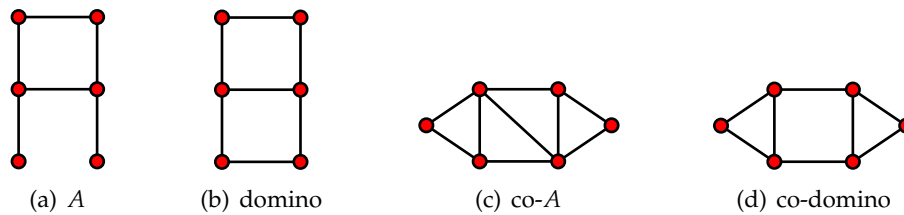
Les graphes (P_5 , cerf-volant)-libres généralisent les graphes semi- P_4 -épars et les graphes (P_5 , diamant)-libres. Brandstädt et Mosca [50] ont donné un algorithme robuste qui trouve une solution à MWIS en $\mathcal{O}(nm)$; il n'est donc pas nécessaire de savoir si le graphe donné en entrée est bien (P_5 , cerf-volant)-libre ou non. Cet algorithme est basé sur le fait que les graphes premiers (P_5 , cerf-volant)-libres qui contiennent un $2K_2$ ont une structure très simple; et pour ceux qui sont sans $2K_2$, un résultat de Farber [130] dit qu'ils possèdent au plus $\mathcal{O}(n^2)$ stables maximaux.

Graphes (P_5 , 2-patte)-libres

Les graphes (P_5 , 2-patte)-libres généralisent les graphes sans $2K_2$ et les graphes P_4 -épars. Brandstädt et Mosca [50] ont montré que les graphes premiers (P_5 , 2-patte)-libres sont sans $2K_2$. Le même argument que pour la classe précédente conduit là aussi à un algorithme polynomial.

Graphes (P_5 , chaise)-libres

La complexité élevée de l'algorithme d'Alekseev pour résoudre MIS dans les graphes sans chaise, conjuguée au fait qu'on ne disposait pas d'algorithme pour MWIS dans cette classe jusqu'à il y a peu (algorithme de Lozin et Milanič), a incité les chercheurs à développer des méthodes plus efficaces, au moins dans des sous-classes. C'était la démarche de Brandstädt, Le et Ridder [47], qui ont obtenu avec succès, entre autres, un algorithme qui trouve un stable de poids maximum en temps $\mathcal{O}(nm)$ dans un graphe (P_5 , drapeau)-libre. De plus, alors que l'algorithme d'Alekseev ne fonctionne pas si le graphe d'entrée contient une chaise, le leur est robuste à ces graphes.

FIGURE 2.12 – Les graphes A et domino, et leur complémentaire.

Graphes $(P_5, \text{co-}A)$ -libres

Puisque l'on sait résoudre le problème du stable maximum dans la classe des graphes sans $2K_2$, nous avons exploré durant la deuxième année de cette thèse la piste des graphes sans P_5 qui contiennent un $2K_2$, et avons réussi à expliciter leur structure [261] :

Théorème 2.38 (Maffray, Morel (2009)) *L'ensemble des sommets de tout graphe sans P_5 qui contient au moins un $2K_2$ peut être partitionné en quatre ensembles :*

- $B = B_1 \cup B_2 \cup \dots \cup B_k (k \geq 2)$, tel que les B_i sont connexes, d'ordre au moins 2, disjoints deux à deux, anticomplets à $B \setminus B_i$ et tel que B est maximal ;
- $A = A_1 \cup A_2 \cup \dots \cup A_k$, tel que A_i est complet à $B \setminus B_i$, et a un voisin et un non-voisin dans B_i ;
- T l'ensemble des sommets complets à B ;
- I l'ensemble des sommets anticomplets à B .

De plus, I est stable.

Démonstration. Soient $G = (V, E)$ un graphe, B un ensemble satisfaisant les hypothèses du théorème (en particulier, il est maximal) et $v \in V \setminus B$. Supposons que v ne soit ni complet, ni anticomplet à B (sinon, par définition, on peut l'inclure dans T ou I). Alors il existe au plus un sous-ensemble B_i contenant un voisin et un non-voisin de v (s'il en existait un deuxième, on aurait un P_5).

- si un tel B_i existe, alors soit v est complet à $B \setminus B_i$ et par définition on peut l'inclure dans A_i ; soit v est anticomplet à $B \setminus B_i$, et on peut alors l'inclure dans B_i , ce qui contredit l'hypothèse de maximalité de B .
- sinon, on peut partitionner B en deux ensembles B' et B'' tels que v est complet à B' et anticomplet à B'' ; on peut alors inclure v dans B' , ce qui contredit à nouveau l'hypothèse de maximalité de B .

De plus, si I contient deux sommets adjacents, ils sont inclus dans une composante connexe qui aurait dû être incluse dans B au départ, une nouvelle contradiction sur le fait que B soit maximal. \square

A présent, on appelle A le graphe obtenu à partir d'un P_6 $v_1-v_2-v_3-v_4-v_5-v_6$ en reliant les sommets v_2 et v_5 , **domino** le graphe obtenu en reliant ensuite les sommets v_1 et v_6 , et $\text{co-}A$ et co-domino leur complémentaire respectif (figure 2.12). Les graphes $(P_5, \text{co-}A, \text{co-domino})$ -libres sont intéressants à étudier, car ils généralisent à la fois les graphes $(P_5, 2\text{-patte})$ -libres et les graphes semi- P_4 -épars. Or il est facile de constater que :

Proposition 2.22 *Avec les notations du théorème précédent, tout graphe $(P_5, co-A, co-domino)$ -libre contient au plus un ensemble A_i non vide.*

Il s'ensuit que si $A_i \neq \emptyset$, alors $B \setminus B_i$ est homogène. On peut ainsi résumer comme suit :

Théorème 2.39 *Tout graphe $(P_5, co-A, co-domino)$ -libre est sans $2K_2$ ou contient un ensemble homogène.*

et on en déduit immédiatement un algorithme polynomial pour calculer un stable de poids maximum dans un graphe $(P_5, co-A, co-domino)$ -libre³⁸. Lozin et Mosca [257] ont récemment étendu ce résultat à la classe des graphes $(P_5, co-A)$ -libres. Ils ont en effet montré que tout graphe premier de cette classe est sans $K_2 + P_3$; or ils avaient prouvé quelques années avant dans [256] que :

Théorème 2.40 (Lozin et Mosca (2005)) *Si H est un graphe tel que MWIS peut être résolu en temps polynomial dans la classe des graphes (P_5, H) -libres, alors le problème a également une solution polynomiale dans la classe des graphes $(P_5, H + pK_2)$ -libres pour tout p fixé.*

Or, MWIS est évidemment résoluble en temps polynomial dans les graphes (P_5, P_3) -libres, d'où le résultat.

Les nombreux résultats exposés ci-dessus démontrent bien toute la puissance de la décomposition modulaire. Nous proposons de présenter un autre outil important au regard du problème du stable maximum : la struction.

2.5 LA STRUCTION

L'idée générale de la struction (pour STability RedUCTION) est de transformer un graphe G en un graphe G' tel que $\alpha(G') = \alpha(G) - 1$. L'origine de la méthode remonte à la toute fin des années 1970, lorsqu'un chercheur genevois, Christien Ebenegger, remarque des relations entre les graphes et certaines fonctions *pseudo-booléennes* (cf. [113]).

2.5.1 Fonctions pseudo-booléennes, posiformes et graphe de conflit

Définition 2.17 *Une fonction booléenne est une fonction $f : \mathbb{B}^n \rightarrow \mathbb{B}$, où $\mathbb{B} = \{0, 1\}$. Une fonction pseudo-booléenne est une fonction $f : \mathbb{B}^n \rightarrow \mathbb{R}$.*

³⁸. Ce résultat n'a pas été publié : nous nous sommes aperçus qu'il s'agit en fait de la formulation dans le complémentaire d'un théorème qu'avaient déjà énoncé Hoàng et Reed [213].

Les fonctions pseudo-booléennes ont été introduites par Hammer, Rosenberg et Rudeanu [189, 190] et jouent un rôle majeur dans la conception des circuits VLSI, en informatique, en statistique, en économie, en théorie des jeux coopératifs, en physique théorique, dans les réseaux de neurones, et bien sûr en optimisation combinatoire [36].

À toute variable booléenne $x_i \in \mathbb{B}$ est associée sa **négation** ou **complément** \bar{x}_i par la relation $\bar{x}_i = 1 - x_i$. Il est bien connu qu'une fonction pseudo-booléenne de $x = (x_1, \dots, x_n)$ possède une écriture unique sous forme d'un polynôme de x , mais admet de nombreuses *réécritures* si l'on fait intervenir les négations :

Définition 2.18 Une *réécriture* d'une fonction pseudo-booléenne f est une expression de f sous la forme

$$c + \sum_{i=1}^p w_i T_i$$

où c est une constante, $T_i = \prod_{j \in A_i} x_j \prod_{k \in B_i} \bar{x}_k$ et $A_i, B_i \subseteq \{1, \dots, n\}$ avec $\forall i \in \{1, \dots, p\} A_i \cap B_i = \emptyset$.

Si de plus tous les w_i sont strictement positifs, on dit que $\sum_{i=1}^p w_i T_i$ est une **posiforme**³⁹.

(la condition $A_i \cap B_i = \emptyset$ exprime simplement qu'aucun terme ne contient une variable et sa négation, puisqu'un tel terme est alors nul d'après les règles de l'algèbre booléenne).

Exemple 2.9 L'expression $2x_1 + 3x_2 + x_1x_2 + 8x_2x_3$ est une posiforme. On peut la réécrire par exemple sous les formes $2 - 2\bar{x}_1 + 4x_2 - \bar{x}_1x_2 + 8x_2x_3$ ou $3 + 3\bar{x}_1 - 3\bar{x}_2 - x_1\bar{x}_2 + 8x_3 - 8\bar{x}_2x_3$, en substituant respectivement x_1 par $1 - \bar{x}_1$ et x_2 par $1 - \bar{x}_2$.

À toute posiforme f , on peut associer un graphe $G_f = (V, E)$, appelé **graphe de conflit** de f , construit ainsi : à chaque terme $w_i T_i$, on associe un sommet v_i de poids w_i , et $E = \{\{v_i, v_j\} : (A_i \cap B_j) \cup (A_j \cap B_i) \neq \emptyset\}$ (autrement dit, il y a une arête entre v_i et v_j si et seulement si T_i et T_j sont en conflit, *i.e.* s'il existe k tel que x_k apparaît dans un terme et \bar{x}_k apparaît dans l'autre). Il est clair que maximiser f revient à trouver un stable de poids maximum dans G_f . Remarquons que par construction, tout sommet x_i de G_f appartient à un biparti (non nécessairement induit) dont les sommets correspondent d'un côté à tous les termes contenant x_i et d'un autre aux termes contenant \bar{x}_i .

Inversement, considérons à présent le problème du stable de poids

³⁹. Dans la littérature, il arrive qu'une posiforme désigne ce que nous appelons ici une réécriture.

maximum de $G = (V, E)$, écrit sous la forme d'un programme linéaire :

$$\begin{aligned} \text{maximiser} \quad & \alpha_w(G) = \sum_{i=1}^n w_i v_i \\ \text{s.c.} \quad & v_i + v_j \leq 1 \quad \forall \{v_i, v_j\} \in E \\ & v_i \in \{0, 1\} \quad \forall v_i \in V \end{aligned}$$

où w est le vecteur des poids des sommets (bien sûr, si toutes ses composantes sont égales, on retombe dans le cas du stable de cardinalité maximum), v est le vecteur caractéristique d'un stable (sa i -ème composante v_i vaut 1 si le sommet v_i appartient au stable considéré, 0 sinon) et la première contrainte exprime le fait qu'au plus une extrémité de chaque arête appartient au stable.

Hamor [191] a observé que ce problème se ramène à celui de la maximisation d'une posiforme, qu'il obtient en recouvrant les arêtes du graphe avec une famille de graphes bipartis complets (non nécessairement induits) $G_j = (V_j, W_j; E_j) (j = 1, \dots, q)$ auxquels sont associées des variables booléennes x_j , puis en posant

$$f(x_1, \dots, x_q) = \sum_{i=1}^{|V|} w_i T_i$$

où $T_i = \prod_{j \in A_i} x_j \prod_{k \in B_i} \bar{x}_k$, $A_i = \{j : v_i \in V_j\}$ et $B_i = \{k : v_i \in W_k\}$.

Exemple 2.10 Considérons le graphe G de la figure 2.13 (les nombres correspondent aux poids des sommets) dont on recouvre les arêtes avec les bipartis complets G_1 et G_2 (avec pour convention que les sommets rouges sont dans V_j et les sommets bleus dans W_j). Par exemple, le sommet b a un poids égal à 3, et apparaît dans les ensembles W_1 et W_2 : le terme correspondant est donc $3\bar{x}_1\bar{x}_2$. Ainsi, la posiforme associée à cette partition des arêtes du graphe est $x_1 + 3\bar{x}_1\bar{x}_2 + x_1 + 4\bar{x}_1x_2 + 2\bar{x}_2$; elle est maximum, de valeur 5, pour $x_1 = 0$ et $x_2 = 0$ (ce qui correspond au poids du stable $\{b, e\}$). Évidemment, son graphe de conflit est le graphe G .

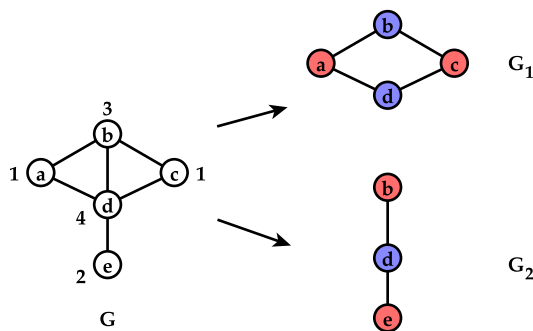


FIGURE 2.13 – Recouvrement des arêtes d'un graphe par des bipartis complets.

Notons que toutes les posiformes associées à G ont la même valeur optimale. Clairement, le maximum de f donne $\alpha_w(G)$: tout T_i ayant pour valeur 1 signifie que v_i appartient au stable de poids maximum, et par

construction deux sommets adjacents ne peuvent être simultanément dans un stable. Ainsi :

Proposition 2.23 *Le problème du stable de poids maximum est équivalent au problème de la maximisation d'une posiforme.*

2.5.2 La struction

On peut bien sûr recouvrir les arêtes d'un graphe par des bipartis de nombreuses façons différentes, et il existe donc beaucoup de posiformes différentes qui peuvent être maximisées pour trouver $\alpha_w(G)$. L'idée d'Ebenegger, et qui est exploitée par Ebenegger, Hammer et de Werra [119], est de choisir arbitrairement un sommet du graphe (le *centre de la struction*), et de recouvrir les arêtes par des étoiles (*i.e.* des bipartis $K_{1,k}$) centrées en chacun des autres sommets. Plus formellement :

- a) on choisit un sommet v_0 et soient v_1, \dots, v_p ses voisins (et donc v_{p+1}, \dots, v_{n-1} ses non-voisins) ;
- b) pour chaque sommet v_i ($i = 1, \dots, p$), on considère le graphe biparti $G_i = (V_i, W_i; E)$ où :
 - $V_i = \{v_i\}$,
 - $W_i = \{v_j \in N(v_i) | j > i\} \cup \{x\}$
- c) pour chaque sommet v_i ($i = p + 1, \dots, n - 1$), on considère le graphe biparti $G_i = (V_i, W_i; E)$ où :
 - $V_i = \{v_i\}$,
 - $W_i = \cup\{v_j \in N(v_i) | j > i\}$

Il est facile de voir qu'ainsi toutes les arêtes du graphe sont bien couvertes. La posiforme associée à cette décomposition est

$$f_G = w_0 \prod_{i=1}^p \bar{x}_i + \sum_{i=1}^{|V|-1} \left(w_i x_i \prod_{\substack{v_j \in N(v_i) \\ j < i}} \bar{x}_j \right)$$

Exemple 2.11 *La struction (centrée au sommet a) du graphe G de la figure 2.13 est donnée à la figure 2.14 (où v_i correspond au centre de G_i), et conduit à la posiforme $\bar{x}_1 \bar{x}_2 + 3x_1 + x_2 \bar{x}_1 + 4x_3 \bar{x}_1 \bar{x}_2 + 2x_4 \bar{x}_2$.*

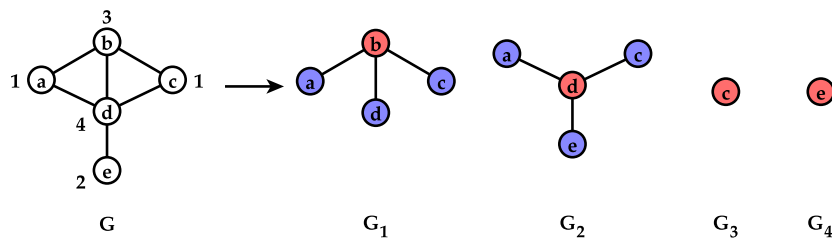


FIGURE 2.14 – Struction du graphe G.

L'avantage de cette décomposition, et donc de la posiforme f_G obtenue, est que l'on peut montrer qu'il est toujours possible (en utilisant de manière répétée l'égalité $\bar{x}_i = 1 - x_i$) de la réécrire sous la forme

$f_G = k + g$ où k est une constante positive et g une posiforme. Or, on peut évidemment construire le graphe de conflit G_g de g et il vient alors immédiatement que

$$\alpha(G_g) = \alpha(G_f) - k$$

ce qui explique l'appellation *stability reduction*. Dans le cas du stable de cardinalité maximum, k vaut 1.

2.5.3 Construction directe

Naturellement, il est possible d'obtenir le graphe G_g directement, sans passer par les posiformes. Par simplicité, nous nous restreignons au cas non pondéré, mais l'algorithme qui suit peut être étendu sans difficulté aux graphes pondérés.

Algorithme 6: Algorithme de struction

Entrées : Un graphe $G = (V, E)$

Sorties : Un graphe G' tel que $\alpha(G') = \alpha(G) - 1$

- 1 Choisir un sommet $v_0 \in V$;
 - 2 Supprimer v_0 ainsi que tous ses voisins v_1, \dots, v_p ;
 - 3 Ajouter l'ensemble de sommets $\{v_{i,j} : 1 \leq i < j \leq p, \{v_i, v_j\} \notin E\}$;
 - 4 Relier $v_{i,j}$ à v_k ($k > p$) si $v_k \in N(v_i) \cup N(v_j)$;
 - 5 Relier $v_{i,j}$ et $v_{k,l}$ si $i \neq k$ ou $\{v_j, v_l\} \in E$;
-

Les nouveaux sommets, créés à l'étape 3, peuvent être organisés par niveau (chaque niveau correspondant à l'indice i); l'étape 5 signifie alors qu'on relie deux sommets du même niveau s'ils étaient adjacents dans G , et qu'on relie tous les sommets appartenant à des niveaux différents. La figure 2.15 montre un exemple d'application de cet algorithme.

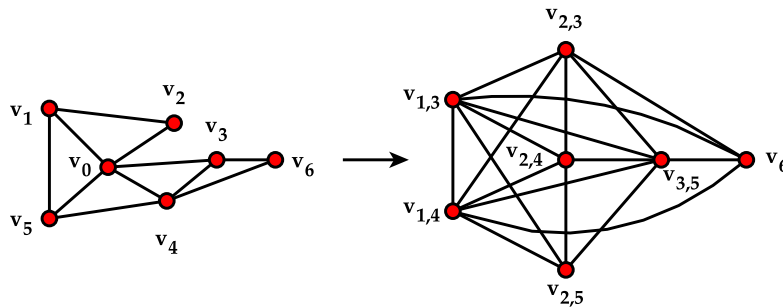


FIGURE 2.15 – Application de l'algorithme 6.

La justification de cet algorithme est donnée dans [119]. Nous redonnons quand même ici la preuve de sa correction.

Proposition 2.24 Le graphe G' obtenu à la fin de l'algorithme 6 vérifie $\alpha(G') = \alpha(G) - 1$.

Démonstration.

- a) Soit S un stable maximum de G . L'ensemble $S_0 = S \cap \{v_0, \dots, v_p\}$ n'est pas vide (sinon, S ne contiendrait que des non-voisins de v_0 , mais il ne serait alors pas maximum puisqu'on pourrait ajouter v_0).

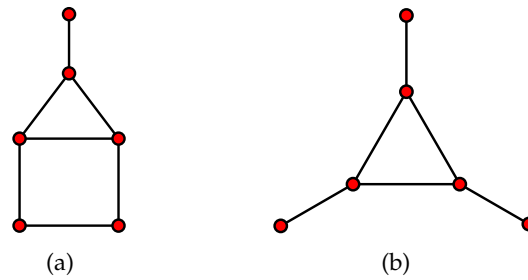


FIGURE 2.16 – (a) L'antenne et (b) le filet.

- Si $|S_0| = 1$, $S \setminus \{v_0, \dots, v_p\}$ est un stable de G' ;
 - sinon, soient $\{v_{i_1}, \dots, v_{i_r}\}$ les sommets de S_0 , avec $i_1 < \dots < i_r$; alors $S \setminus S_0 \cup \{v_{i_1, i_2}, v_{i_1, i_3}, \dots, v_{i_1, i_r}\}$ est un stable de G' .
- Dans les deux cas, le stable obtenu est de cardinalité $\alpha(G) - 1$ donc $\alpha(G') \geq \alpha(G) - 1$.

- b) Soit S un stable maximum de G' .
- Si S ne contient aucun nouveau sommet, $S \cup \{v_0\}$ est un stable de G ;
 - sinon, soient $\{v_{i, i_1}, \dots, v_{i, i_r}\}$ les nouveaux sommets appartenant à S (par construction, ils appartiennent tous au même niveau i); alors $S \setminus \{v_{i, i_1}, \dots, v_{i, i_r}\} \cup \{v_i, v_{i_1}, \dots, v_{i_r}\}$ est un stable de G .
- Dans les deux cas, le stable obtenu est de cardinalité $\alpha(G') + 1$ donc $\alpha(G') \leq \alpha(G) - 1$.

□

2.5.4 Complexité

En répétant l'exécution de cet algorithme sur le graphe G' , et ainsi de suite, on obtient un algorithme permettant de calculer $\alpha(G)$. La construction de G' se fait en $\mathcal{O}(n^2)$ (où $n = |V|$), et l'algorithme est exécuté au plus n fois. Le problème est que le nombre de sommets dans la suite des graphes G' peut croître de manière exponentielle, et la struction ne conduit donc pas toujours à un algorithme polynomial. Il arrive néanmoins que pour certaines classes de graphes, G' a toujours moins de sommets que G (c'est par exemple le cas des graphes **séries-parallèles**). On trouvera également dans [119] quelques « astuces » quant au choix du sommet jouant le rôle du centre de la struction.

2.5.5 D'autres types de transformations ⁴⁰

Puisque la méthode, dans le cas général, ne conduit pas nécessairement à un algorithme polynomial, d'autres transformations, qui préservent le nombre de stabilité d'un graphe ou le modifient par une constante, ont été développées en s'appuyant sur certaines propriétés structurelles.

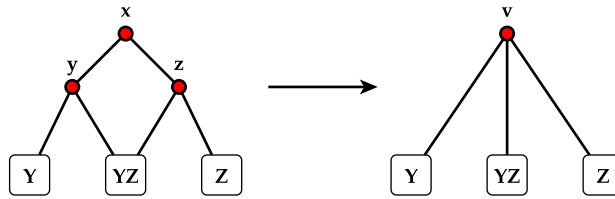


FIGURE 2.17 – Repliage de sommet.

Cas des graphes sans griffe

Hammer, Mahadev et de Werra se sont intéressés au cas des graphes sans griffe, afin de proposer une méthode alternative à l’algorithme des chaînes augmentantes. Un premier résultat concerne la sous-classe des **graphes CAN-libres** (où C désigne la griffe (*claw*), A l’antenne (*antenna*) et N le filet (*net*) (cf. figure 2.16)) [186], pour lesquels ils ont décrit une transformation polynomiale qui associe à tout graphe CAN-libre G un autre graphe CAN-libre G' ayant strictement moins de sommets que G et vérifiant $\alpha(G') = \alpha(G) - 1$, ce qui permet de calculer un stable maximum en temps polynomial (en $\mathcal{O}(n^3)$ dans le cas présent). Le procédé a été ensuite adapté aux graphes CN-libres [187] (étudiés précédemment pour leur propriétés hamiltoniennes [118, 175]), pour lesquels la struction générale ne conduit pas à un algorithme polynomial, et selon les auteurs, « il n’existe pas d’extension simple de cette struction spécialisée au cas des graphes pondérés ». Brandstädt et Dragan [38] ont montré qu’on peut, par une méthode totalement différente, calculer un stable maximum (ainsi qu’un dominant stable minimum) dans un graphe CN-libre en $\mathcal{O}(\delta m)$. Enfin, Hertz [199] a décrit une struction d’un graphe sans griffe en un autre graphe sans griffe, mais son algorithme fait appel à celui de Sbihi dans certains cas...

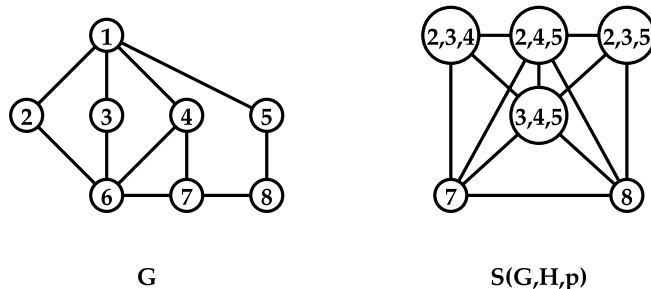
Le repliage de sommets [74]

Supposons que dans un graphe G un sommet x ait exactement deux voisins non adjacents y et z . On appelle G' le graphe obtenu en supprimant x , y et z et en introduisant un nouveau sommet v adjacent à tout autre voisin de y ou z dans G (cf. figure 2.17). Il n’est pas difficile de vérifier que $\alpha(G') = \alpha(G) - 1$. En fait, cette transformation n’est qu’un cas particulier de la struction (celui où le centre n’a que deux voisins). Elle est appelée repliage de sommet (*vertex folding*) et a été utilisée pour réduire la complexité des problèmes du stable maximum et du transversal minimum dans les graphes de faible degré maximum.

La séparation de sommet [3]

La séparation de sommet (*vertex splitting*) est l’inverse de la transformation précédente. C’est grâce à cette opération qu’Alekseev a démontré le théorème 2.10.

40. Nous avons jugé utile de laisser cette section dans le présent manuscrit pour les méthodes qui y sont abordées. Elle peut cependant être sautée en première lecture sans que cela ne gêne la compréhension de la suite.

FIGURE 2.18 – Struction totale de G avec $H = G[1,2,3]$ et $p = 2$.

La struction totale [13]

Il s'agit d'une généralisation de la struction où la notion de centre est étendue à un sous-graphe induit. On suppose ici que les sommets des graphes sont numérotés de 1 à n ; dans un ensemble de sommets A , le sommet ayant le plus grand numéro est noté $m(A)$ et on pose $A^- = A \setminus \{m(A)\}$. Étant donné un tel graphe G , un sous-graphe induit H et un entier positif $p \leq \alpha(H)$, on pose $R = V_G \setminus N[V_H]$ et on définit le graphe $S(G, H, p)$ de la façon suivante :

- il a pour ensemble de sommets $R \cup W$, où W est l'ensemble de tous les stables de cardinalité $p + 1$ dans $G[N[V_H]]$;
- il a pour ensemble de d'arêtes :
 - les arêtes de $G[R]$,
 - les arêtes reliant les sommets de $A \in W$ et $B \in W$ si $A^- \neq B^-$ ou $\{m(A), m(B)\} \in E_G$,
 - les arêtes $\{A, r\}$ ($A \in W, r \in R$), si dans G , r a un voisin dans A .

Alexe, Hammer, Lozin et de Werra ont montré [13] qu'on a alors :

Théorème 2.41 $\alpha(S(G, H, p)) = \alpha(G) - p$.

Un exemple de struction totale est donné à la figure 2.18.

La réduction de sous-graphe [13]

On désigne par cette expression plusieurs méthodes combinant la struction à des transformations préservant le nombre de stabilité. Pour un sous-graphe induit H de G et $v \in V_G \setminus V_H$, on note $H + v$ le graphe $G[V_H \cup \{v\}]$ et on dit que H est α -maximal dans G si $\alpha(H + v) = \alpha(H) + 1$ pour tout sommet $v \in V_G \setminus V_H$. Alors la H -réduction de G consiste à supprimer H et relier deux sommets x et y de $N(V_H)$ si $\alpha(H + x + y) = \alpha(H) + 1$.

Un premier exemple bien connu de H -réduction est la *contraction des blossoms* utilisée par Edmonds dans l'algorithme du couplage maximum (cf. section 2.1.4). En effet, la contraction des blossoms dans G est équivalente à la H -réduction dans $L(G)$ (pour un certain graphe H). Autrement dit, étant donné un graphe G , un couplage M , un cycle C (non nécessairement induit) de longueur $2k + 1$ contenant exactement k arêtes de M , et G' le graphe obtenu en contractant C en un seul sommet, $L(G')$ est obtenu par H -réduction de $L(G)$, et la proposition suivante

Proposition 2.25 (Edmonds) *M est un couplage maximum de G si et seulement si $M \setminus E_C$ est un couplage maximum de G' .*

permet de montrer que $\alpha(L(G')) = \alpha(L(G)) - \alpha(H)$.

Un autre exemple est la *clique-réduction*, i.e. une H -réduction avec une clique H , introduite par Lovász et Plummer [251] pour leur algorithme du stable maximum dans les graphes sans griffe (cf. section 2.1.4) :

Proposition 2.26 (Lovász et Plummer (1986)) *Soit H une clique d'un graphe sans griffe G vérifiant $\alpha(N(H)) \leq 2$. Le graphe G' obtenu par H -réduction de G vérifie $\alpha(G') = \alpha(G) + 1$.*

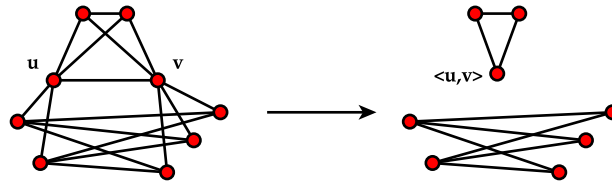
Suppression par sommet simplicial

Si l'on appelle α -**redondant** [49] un sommet dont la suppression ne change pas le nombre de stabilité d'un graphe, il n'est pas difficile de montrer que tout voisin d'un sommet simplicial est α -redondant. Autrement dit, si v est un sommet simplicial de G alors $\alpha(G) = \alpha(G \setminus N[v]) + 1$. Ce fait suggère d'une part un algorithme très efficace pour calculer un stable maximum dans un graphe cordal, et plus généralement la struction suivante (qui correspond à la struction générale lorsque le centre est un sommet simplicial) :

1. $k = 0$; $G' = G$;
2. Tant qu'il existe un sommet simplicial v dans G' , poser $k = k + 1$ et $G' = G' \setminus N_G[v]$;
3. Tester si G' contient un stable de taille t (t fixé); si non, calculer $\alpha(G')$ et poser $\alpha(G) = k + \alpha(G')$.

La complexité temporelle de cet algorithme est clairement bornée par $\mathcal{O}(n^t)$: on peut tester en temps $\mathcal{O}(n^3)$ si un graphe possède un sommet simplicial et cette étape est répétée au plus n fois; le test de l'étape 3 peut être effectué en $\mathcal{O}(n^t)$ en examinant tous les sous-ensembles de t sommets, et si $\alpha(G') < t$, il peut être déterminé en $\mathcal{O}(n^{t-1})$ de la même manière. Or, Brandstädt et Hammer [40] ont observé que les graphes ($P_5, K_{1,4}$, **drapeau, chaise**)-libres (qui généralisent les graphes (P_5 , griffe)-libres) possèdent une propriété particulière : ils ont un sommet simplicial ou leur nombre de stabilité est strictement inférieur à 4. Autrement dit, on peut appliquer l'algorithme précédent en posant $t = 4$, sans avoir besoin de tester au préalable si G est bien un élément de cette classe : si, à l'étape 3, on trouve un stable de taille 4, on sait que ce n'est pas le cas, et l'algorithme s'arrête. Notons que cet algorithme fonctionne aussi pour MWIS.

Les deux transformations qui suivent ne sont pas à proprement parler des structions, dans le sens où elles construisent à partir d'un graphe G un graphe G' ayant même nombre de stabilité (mais possédant néanmoins strictement moins de sommets). Elles sont cependant suffisamment utilisées pour mériter qu'on les évoque ici.

FIGURE 2.19 – Démagnétisation de l'aimant $\{u, v\}$.

La réduction de voisinage

Si a et b sont deux sommets adjacents tels que $N(b) \subseteq N[a]$, alors la suppression de a ne modifie pas le nombre de stabilité. Cette transformation est appelée **réduction de voisinage** et généralise la réduction par sommet simplicial (si v est simplicial, la suppression de $N(v)$ peut être vue comme une suite de réductions de voisinage pour tout voisin de a de v puisque $N(v) \subseteq N[a]$). La réduction de voisinage a été utilisée par Golumbic et Hammer [167], qui ont obtenu un algorithme linéaire pour calculer un stable maximum dans les graphes arc-circulaires⁴¹.

La démagnétisation

Hammer et Hertz [185] ont développé une transformation basée sur les fonctions pseudo-bouliennes qui généralise la réduction de voisinage, appelée **démagnétisation**, que nous trouvons intéressant de présenter rapidement.

Définition 2.19 Un *aimant* (magnet) est une paire de sommets adjacents $\{a, b\}$ tels que $N(a) \setminus N[b]$ est complet à $N(b) \setminus N[a]$.

Si $G = (V, E)$ contient un aimant $\{a, b\}$, on peut couvrir les arêtes incidentes à a ou à b avec les deux bipartis complets $G_1 = (N(b) \setminus N(a), N(a) \setminus N(b); E_1)$ et $G_2 = (\{a, b\}, N(a) \cap N(b); E_2)$, et les arêtes restantes avec des bipartis complets G_3, \dots, G_q . En supposant que G ne soit pas pondéré, la posiforme associée $f = \sum_{v \in V} T_v$ est telle que $T_a = x_1 x_2$ et $T_b = \bar{x}_1 x_2$; donc $T_a + T_b = (x_1 + \bar{x}_1) x_2 = x_2$. Autrement dit, f a le même maximum que $g = \sum_{v \in V \setminus \{a, b\}} T_v + x_2$. Le graphe de conflit $G_g = (V', E')$ de g vérifie alors $|V'| = |V| - 1$ et $\alpha(G_g) = \alpha(G)$. Bien sûr, il peut être construit directement, en supprimant a ainsi que toutes les arêtes $\{v, b\}$ où $v \in N(b) \setminus N(a)$ (cf. figure 2.19).

Il est immédiat d'observer que $\{u, v\}$ forme un aimant si et seulement si ce n'est pas l'arête du milieu dans un P_4 induit. Il s'ensuit qu'un graphe ne possède pas toujours un aimant. Chvátal et Rusu [91] ont obtenu le résultat suivant :

Proposition 2.27 Soit G un graphe sans cycle de longueur supérieure ou égale à 5. Alors soit G contient un aimant, soit il est stable.

41. Un graphe **arc-circulaire** *arc-circular graph* a pour ensemble de sommets des arcs de cercle, qui sont reliés lorsque les arcs s'intersectent.

Cependant, la démagnétisation peut, elle, créer de tels cycles. Hertz et de Werra [202] ont caractérisé les graphes pour lesquels on peut répéter l'utilisation de la démagnétisation, jusqu'à obtenir un stable.

2.5.6 Application aux graphes sans P_5

Les transformations évoquées ci-dessus ont permis de résoudre le problème du stable maximum dans plusieurs classes de graphes sans P_5 . Nous avons par exemple vu que la suppression de sommet simplicial a permis de résoudre le problème dans la classe des graphes $(P_5, K_{1,4}, \text{drapeau, chaise})$ -libres, qui contiennent les graphes (P_5, griffe) -libres. Mahadev [267] l'a utilisée pour une autre sur-classe des graphes (P_5, griffe) -libres, en montrant que tout sommet des graphes $(P_5, K_{2,3}, \text{drapeau})$ -libres est soit simplicial, soit α -redondant. Rappelons que Mosca a résolu MIS puis MWIS dans les graphes $(P_5, K_{2,3})$ -libres.

Dans [200], Hertz a développé une autre transformation et a obtenu un algorithme en $\mathcal{O}(n^2)$ pour trouver un stable maximum dans la classe des graphes $(2K_2, C_4)$ -libres, qu'on appelle aussi graphes pseudo-scindés (notons que par autocomplémentarité, cet algorithme permet aussi de trouver une clique maximum); nous reparlerons de ces graphes (pour lesquels il existe en fait un algorithme linéaire) un peu plus loin, dans la section 2.7.

Alekseev et Lozin [10, 12] ont proposé une transformation impliquant la réduction de voisinage, obtenant un algorithme polynomial pour les graphes (griffe, patte) -libres, les graphes (P_4, griffe) -libres et surtout pour les graphes $(P_k, C_{k-1}, C_{k-2}, \dots, C_4)$ -libres pour n'importe quel k fixé. En particulier, pour $k = 5$, on a la classe des graphes (P_5, C_4) -libres, contenue dans la classe des graphes $(P_5, K_{2,3}, \text{drapeau})$ -libres évoquée ci-dessus et qui contient les graphes pseudo-scindés. Signalons que Fouquet, Giakoumakis, Maire et Thuillier [138] avaient fourni une caractérisation des graphes (P_5, C_4) -libres à partir de laquelle on peut également dériver un algorithme polynomial pour MIS.

Hertz [201] a ensuite appliqué sa méthode aux classes des graphes $(P_5, 3K_2, \text{drapeau, gemme})$ -libres⁴² et $(P_5, \text{drapeau, diamant})$ -libres⁴³ (cf. figure 2.11 (d)) Cette dernière est contenue dans la classe des graphes $(P_5, \text{diamant})$ -libres, pour laquelle Arbib et Mosca [15] ont montré, en mettant à profit la structure de ces graphes, qu'il existe un algorithme de complexité $\mathcal{O}(n^2)$ pour résoudre MWIS (notons qu'ils ont également résolu les problèmes de la coloration, de la couverture par cliques, et donné une description complète du polytope des stables de cette classe); Brandstädt [37] a montré depuis, en utilisant la décomposition modulaire, qu'il existe des algorithmes linéaires. D'autre part, les deux classes étudiées par Hertz sont simultanément (P_5, gemme) -libres et $(P_5, \text{drapeau})$ -libres, et nous avons vu à la section précédente que MWIS et d'autres problèmes

42. La **gemme** (*gem*) est le graphe obtenu à partir d'un P_4 auquel on ajoute un sommet universel.

43. Le **diamant** (*diamond*) est le graphe obtenu à partir d'un K_4 en supprimant une arête; c'est donc un sous-graphe induit de la gemme.

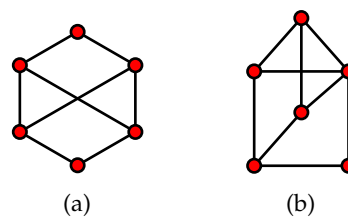


FIGURE 2.20 – (a) $K_{3,3} - e$ et (b) la maison jumelée.

d'optimisation combinatoire peuvent être résolus en temps linéaire dans la première classe.

En ce qui concerne les graphes $(P_5, \text{drapeau})$ -libres, MIS a été résolu par Lozin [252] : il a montré, en utilisant la réduction de voisinage, qu'un graphe $(P_5, \text{drapeau})$ -libre « réduit » est un graphe $(P_5, \text{drapeau, chaise})$ -libre, classe pour laquelle il a apporté une solution à l'aide des graphes augmentants (cf. section 2.3.5); la complexité de l'algorithme résultant est $\mathcal{O}(n^5)$. Cette borne a été ramenée à $\mathcal{O}(nm)$ par Brandstädt et Lozin [49] qui ont prouvé que dans un graphe $(P_5, \text{drapeau})$ -libre, tout milieu d'un P_4 ⁴⁴ est α -redondant; pour trouver un stable maximum dans un graphe $(P_5, \text{drapeau})$ -libre, il suffit donc de réduire le milieu d'un P_4 tant qu'il en existe un; l'algorithme de reconnaissance des cographes [98] permet de trouver un P_4 en temps linéaire, et si le graphe est un cographe, nous avons vu comment exploiter le coarbre pour trouver un stable maximum en temps linéaire (cf. section 2.4.4); signalons que l'algorithme de Brandstädt et Lozin est robuste, et peut être appliqué aux graphes d'une classe plus générale dans laquelle les P_4 utilisés pour la réduction ont une certaine propriété : la classe des graphes $(P_5, K_{3,3} - e, \text{maison jumelée})$ -libres⁴⁵. Enfin, un algorithme polynomial, de complexité $\mathcal{O}(n^3m)$, pour résoudre MWIS dans les graphes $(P_5, \text{drapeau})$ -libres a été donné par Brandstädt et Hoàng [42, 43] et repose sur les cliques d'articulation, que nous aborderons dans la section suivante.

Graphes $(P_5, K_{m,m})$ -libres (m fixé)

Gerber et Lozin [155, 159] ont montré que dans un graphe $(P_5, K_{m,m})$ -libre, le centre d'un $K_{1,2m-2}$ induit est α -redondant. Après avoir supprimé de tels sommets tant qu'il en existe (leur recherche peut se faire en $\mathcal{O}(n^{2m})$), il suffit d'appliquer l'algorithme de Mosca pour les graphes $(P_5, K_{1,m})$ -libres (cf. section 2.3.5); il est donc possible de trouver un stable maximum en temps $\mathcal{O}(n^{2m})$ dans cette classe qui généralise celles des graphes $(P_5, K_{1,m})$ -libres et, pour $m \leq 3$, des graphes $(P_5, K_{2,3})$ -libres et des graphes (P_5, C_4) -libres.

44. Pour tout P_4 $a-b-c-d$, les milieux (*midpoints*) sont les sommets b et c .

45. Comme son nom l'indique, un $K_{3,3} - e$ est obtenu à partir d'un $K_{3,3}$ en supprimant une arête. Une maison jumelée (*twin-house*) s'obtient à partir d'une maison en ajoutant un faux-jumeau à un sommet de degré 3 (cf. figure ??).

Graphes $(P_5, D_{m+1,m})$ -libres (m fixés)

Un $D_{m+1,m}$ généralise le D_m rencontré à la section 2.3.5 : on le construit à partir d'un mK_2 auquel on ajoute $m + 1$ sommets (qui induisent un stable) complets à ce mK_2 . Brandstädt, Le et Le [45] ont montré que dans un graphe $(P_5, D_{m+1,m})$ -libre, le sommet universel d'un D_m est α -redondant. Il suffit donc de rechercher les D_m et d'effectuer cette suppression tant que cela est possible, puis d'appliquer l'algorithme de Gerber et Lozin pour les graphes (P_5, D_m) -libres.

Nous présentons rapidement dans la section suivante une quatrième méthode : la décomposition par cliques d'articulation.

2.6 LA DÉCOMPOSITION PAR CLIQUES D'ARTICULATION

2.6.1 Présentation

L'intérêt pour les **cliques d'articulation** (*clique separator*, *clique cutset* ou encore *separating clique*), c'est-à-dire des cliques telle que leur suppression déconnecte un graphe, remonte à l'époque, pas si lointaine, où l'on ne disposait pas encore d'algorithme permettant de reconnaître les graphes parfaits⁴⁶. En effet, il est bien connu que si deux graphes parfaits G_1 et G_2 contiennent deux cliques de même taille, le graphe obtenu en identifiant ces deux cliques reste parfait ; ainsi, une première étape naturelle pour tester la perfection d'un graphe pourrait être de chercher s'il possède une clique d'articulation, et si oui, de tester récursivement la perfection des sous-graphes reliés par cette clique.

Cette **décomposition par cliques d'articulation** a été mise en œuvre pour la première fois en 1976 par Gavril [154], qui a défini la classe des graphes *cliques-séparables*⁴⁷, une sous-classe de graphes parfaits contenant notamment les arbres, les graphes cordaux et les graphes à seuil. Sans rentrer dans les détails, un graphe appartient à cette classe si tout sous-graphe induit qui n'a pas une structure particulière⁴⁸ possède justement une clique d'articulation. Cette propriété a permis à Gavril de mettre au point un algorithme polynomial (de complexité $\mathcal{O}(n^5)$) pour reconnaître les graphes cliques-séparables.

46. Ce problème a été résolu en 2002 par Chudnovsky et Seymour qui ont fourni un algorithme de complexité $\mathcal{O}(n^9)$, et indépendamment par Cornuéjols, Liu et Vušković qui ont obtenu un algorithme en $\mathcal{O}(n^{18})$. Les deux algorithmes reposent sur une même première phase dite de *nettoyage* élaborée communément par les deux équipes ; mais la deuxième phase est très différente : l'algorithme de Chudnovsky et Seymour consiste à rechercher un trou impair dans un graphe nettoyé, alors que celui de Cornuéjols, Liu et Vušković utilise un théorème de décomposition de Conforti, Cornuéjols et Vušković pour les graphes sans trou impair [94]. Les résultats de ces recherches figurent finalement dans un seul article [76].

47. Ces graphes apparaissent en fait dès 1962 dans un article de Gallai [147], mais sans nom particulier.

48. Nous donnons malgré tout cette structure pour le lecteur curieux : un graphe est clique-séparable si chacun de ses sous-graphe induits possède une clique d'articulation, ou est k -parti complet, ou bien peut être décomposé en un biparti complet à une clique.

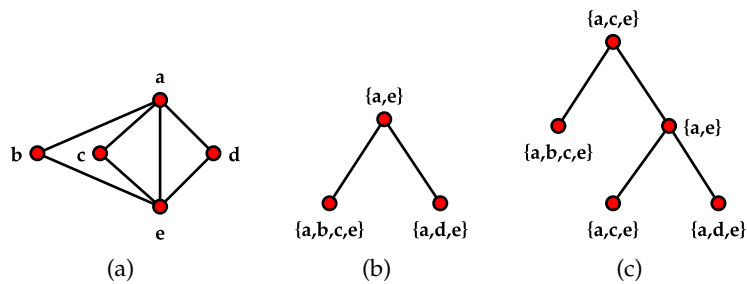


FIGURE 2.21 – (a) Un graphe et (b), (c) deux décompositions par cliques d'articulation possibles.

L'idée est que la nature récursive de cette décomposition amène, comme pour la décomposition modulaire, à un **arbre de décomposition par cliques d'articulation** (mais qui n'est cette fois-ci pas forcément unique (cf. figure 2.21) ; même si l'on ne considère que des cliques d'articulation *minimales*, l'arbre dépend de l'ordre dans lequel ces cliques sont traitées). Chaque nœud de cet arbre qui n'est pas une feuille correspond à une clique d'articulation C , et les sous-arbres issus d'un nœud sont les sous-graphes $G[V_1 \cup C], G[V_2 \cup C] \dots$ (où $V_1, V_2 \dots$ sont les composantes connexes de $G[V \setminus C]$) reliés par cette clique. Les feuilles sont quant à elles associées aux sous-graphes ne possédant pas de clique d'articulation ; ces sous-graphes « terminaux » sont appelés *sous-graphes primitifs* dans [342] et *atomes* dans [327], terme qui s'est imposé. Alors si l'on sait résoudre un problème donné dans ces « atomes », l'on peut, par composition des résultats obtenus, remonter jusqu'à la racine de l'arbre et ainsi résoudre le problème dans le graphe de départ. De plus, il est facile de voir que :

Proposition 2.28 (Gavril) *Un arbre de décomposition par cliques d'articulation possède au plus n^2 nœuds.*

Démonstration. Par définition d'une clique d'articulation, pour tout nœud de l'arbre autre que la racine, il existe un autre nœud ayant même père. Ces deux nœuds correspondent donc à deux graphes G_i et G_j tels qu'il existe deux sommets (distincts) non adjacents v_i dans G_i et v_j dans un G_j . On peut ainsi associer à chaque nœud un couple (i, j) , et un couple ne peut être associé à plus d'un nœud. Comme il existe n^2 couples différents, la proposition est démontrée. \square

Ceci permet à Gavril d'obtenir également des algorithmes polynomiaux pour les problèmes de la clique maximum et de la coloration minimum pour les graphes cliques-séparables. Toutefois, il laisse ouverts les problèmes du stable maximum et de la couverture par cliques, et écrit qu'« il n'est facile de savoir si un graphe contient une clique d'articulation ».

Whitesides [341, 342] et Tarjan [327] ont indépendamment répondu à ces questions, en montrant notamment qu'il est en fait possible de trouver une clique d'articulation en temps polynomial. Leurs deux algorithmes sont très différents. Celui de Whitesides est basé sur la recherche de trous dans le graphe et trouve une clique d'articulation en

temps $\mathcal{O}(nm)$ ⁴⁹ si le graphe donné en contient une; comme l'arbre de décomposition contient $\mathcal{O}(n^2)$ nœuds, on en déduit un algorithme de décomposition de complexité $\mathcal{O}(n^3m)$. Celui de Tarjan repose sur Lex-BFS et une approche « diviser pour mieux régner », et trouve aussi une clique d'articulation en $\mathcal{O}(nm)$, mais c'est l'intégralité de la décomposition peut être effectuée cette fois-ci en $\mathcal{O}(nm)$. Notons également que Kratsch et Spinrad [235, 236, 237] ont utilisé l'algorithme de multiplication de matrices de Coppersmith et Winograd pour abaisser la complexité de la recherche d'une clique d'articulation à $\mathcal{O}(n^{2.69})$; cette complexité a même été ramenée depuis à $\mathcal{O}(n^{2.376} \log n)$ [198], mais les méthodes résultantes sont difficiles à mettre en œuvre en pratique en raison de la présence de constantes numériques très grandes. Nous n'explicitons pas ces algorithmes ici, néanmoins nous allons voir comment la décomposition par cliques d'articulation permet d'aborder le problème du stable de poids maximum.

2.6.2 Calcul d'un stable de poids maximum

Bien que les algorithmes de décomposition de Whitesides et Tarjan soient différents, une fois que l'on a une décomposition, la recherche d'un stable (de poids) maximum est sensiblement la même, et consiste comme nous l'avons vu à résoudre le problème dans les atomes, puis à remonter ces résultats partiels jusqu'à la racine. Nous redonnons ici la méthode récursive décrite par Tarjan. Notons que la décomposition qu'il utilise est particulière : chaque clique d'articulation figurant dans l'arbre sépare le graphe en un atome et le reste du graphe (non nécessairement connexe); cette décomposition produit donc au plus $n - 1$ atomes.

Algorithme 7: Calcul d'un stable de poids maximum avec une décomposition par cliques d'articulation

Entrées : Un graphe pondéré $G = (V, E)$, de vecteur poids w

Sorties : Un stable de poids maximum S de G

- 1 Partitionner V en $\{A, B, C\}$, où C est une clique, A est anticomplet à B et $G[A \cup C]$ est un atome;
 - 2 Pour chaque sommet $v \in C$, déterminer un stable de poids maximum $S(v)$ dans $G[A \setminus N(v)]$ et un stable de poids maximum S' de $G[A]$;
 - 3 Pour chaque sommet $v \in C$, redéfinir $w(v)$ par $w(v) + w(S(v)) - w(S')$, puis déterminer un stable de poids maximum S'' dans $G[B \cup C]$ avec ces nouveaux poids;
 - 4 Finalement, $S = S(v) \cup S''$ où $v \in S'' \cap C$ si $S'' \cap C \neq \emptyset$, ou $S = S' \cup S''$ si $S'' \cap C = \emptyset$.
-

De manière moins formelle, l'étape 2 consiste à trouver pour chaque sommet v de C un stable maximum de $G[A]$ si l'on choisit d'inclure v dans le stable maximum S de G , et un stable maximum S' de $G[A]$ (pour le cas où S ne contient aucun sommet de C); le nouveau poids calculé à l'étape

⁴⁹. Whitesides écrit « en $\mathcal{O}(n^3)$ », mais Tarjan montre qu'une analyse plus fine conduit à $\mathcal{O}(nm)$.

3 correspondent donc à la perte ou au gain sur la valeur du stable si l'on choisit de privilégier v dans S plutôt que les sommets de S' .

Exemple 2.12 Donnons les poids 1, 2, 3, 2, 4 respectivement aux sommets a, \dots, e du graphe de la figure 2.21, et utilisons par exemple la décomposition (c). On a donc $A = \{b\}$, $B = \{d\}$ et $C = \{a, c, e\}$, d'où $S(a) = \emptyset$, $S(c) = \{b\}$, $S(e) = \emptyset$ et $S' = \{b, c\}$. Les nouveaux poids des sommets de C sont donc : $w(a) = 1 + 0 - 3 = -2$, $w(c) = 2 + 3 - 3 = 2$ et $w(e) = 4 + 0 - 3 = 1$. On cherche à présent un stable de poids maximum dans $G[B \cup C]$: ici, on trouve $S'' = \{c, d\}$. Comme $S'' \cap C = \{c\} \neq \emptyset$, $S = S(c) \cup S'' = \{b, c, d\}$, pour un stable de poids total 7.

L'algorithme consiste donc à résoudre $\mathcal{O}(n)$ problèmes de stables pour chaque atome, soit au total $\mathcal{O}(n^2)$ sous-problèmes. Par conséquent, à chaque fois que l'on sait résoudre M(W)IS en temps T pour les atomes d'un graphe, on sait résoudre le même problème en temps $\mathcal{O}(n^2T)$ dans le graphe tout entier. Cependant, on ne connaît encore que peu d'exemples pour lesquels cette approche peut être appliquée pour obtenir un algorithme polynomial pour M(W)IS dans une classe de graphes.

Remarquons que la décomposition par cliques d'articulation fournit une solution directe au problème du stable de poids maximum dans les graphes triangulés. En effet, ces graphes possédant toujours un sommet simplicial (cf. section 1.1.8), il s'ensuit que ce sont précisément les graphes dont les atomes sont des cliques. Notons également que cette méthode permet de résoudre d'autres problèmes, notamment ceux de la clique maximum et de la coloration. Enfin, Brandstädt et Hoàng [42] ont montré qu'une combinaison de la décomposition modulaire avec la décomposition par cliques d'articulation peut être efficace pour résoudre le problème du stable de poids maximum : en effet, s'il est résoluble en temps T sur les atomes premiers d'un graphe, alors il est résoluble en temps $\mathcal{O}(n^2T)$ sur le graphe entier.

2.6.3 Application aux graphes sans P_5

Un exemple de classe pour laquelle l'approche par cliques d'articulation fonctionne bien est la classe des graphes $(P_5, \overline{P_2 + P_3})$ -libres (le graphe $\overline{P_2 + P_3}$ est illustré à la figure 2.22). Alekseev [6] a en effet montré que les atomes des graphes $(P_5, \overline{P_2 + P_3})$ -libres sont sans $3K_2$, ce qui implique que l'on peut résoudre MWIS en $\mathcal{O}(n^7m)$ dans cette classe. Brandstädt et Hoàng [42] ont ensuite montré que tout atome d'un graphe $(P_5, \overline{P_2 + P_3})$ -libres est *presque* $(P_5, C_5, \text{maison})$ -libre (c'est-à-dire que pour chacun de ses sommets v , le non-voisinage de v induit un sous-graphe $(P_5, C_5, \text{maison})$ -libre) ou est *spécifique* (i.e. un type de graphe simple pour lequel on sait résoudre MWIS de manière triviale). Ceci a conduit à un algorithme permettant de calculer un stable de poids maximum en temps $\mathcal{O}(n^4m)$ dans un graphe $(P_5, \overline{P_2 + P_3})$ -libre.

Ce résultat a été généralisé dans [48] aux graphes (P_5, F_1) -libres (qui étendent notamment les graphes $(P_5, \text{cerf-volant})$ -libres et les graphes $(P_5, \text{drapeau})$ -libres) ; les atomes *premiers* de ces graphes sont aussi sans $3K_2$, ce qui conduit à un algorithme de complexité $\mathcal{O}(n^7m)$ (Mosca, dans [286], a

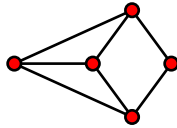


FIGURE 2.22 – Le graphe $\overline{P_2 + P_3}$

ramené cette complexité à $\mathcal{O}(n^4m)$). Les atomes des graphes (P_5, F_2) -libres sont quant à eux sans $4K_2$, ce qui conduit à un algorithme en $\mathcal{O}(n^9m)$. Il est possible d'étendre ces résultats à la classe des graphes (P_5, H_k) -libres, pour laquelle il est prouvé dans [48] qu'il existe un algorithme polynomial de complexité $\mathcal{O}(n^{8k+2})$, à condition de se restreindre à la version non pondérée du problème du stable (la technique utilisée pour les deux classes précédentes ne peut s'appliquer à ce cas plus général).

2.7 GÉNÉALOGIE DES GRAPHES SANS P_5

Nous nous proposons de donner ici une synthèse des classes rencontrées dans ce chapitre sous forme d'une « généalogie », c'est-à-dire un diagramme des relations existant entre ces différentes classes et quelques nouvelles (comme les graphes *scindés*) que nous jugeons intéressant de présenter ici. Bien sûr, on ne saurait être exhaustif dans une telle entreprise (il existe une infinité de classes de graphes sans P_5).

Cet « arbre généalogique » (figure 2.23) ne fait pas apparaître un ancêtre unique. Initialement, nous avons choisis comme « plus lointain ancêtre » la classe des **graphes à seuil**. Chvátal et Hammer [88] ont montré que ces graphes sont exactement les graphes $(2K_2, C_4, P_4)$ -libres. Ce sont donc d'une part des **cographe**s (graphes sans P_4) (ce qui implique qu'on peut résoudre MWIS en temps linéaire dans les graphes à seuil) et d'autre part des **graphes scindés** (*split graphs*).

Les **graphes scindés** ont été introduits par Földes et Hammer [133, 134] ; ce sont les graphes dont on peut partitionner les sommets en un stable et une clique⁵⁰. Földes et Hammer ont montré d'une part qu'un graphe est scindé si et seulement s'il est $(2K_2, C_4, C_5)$ -libre, et d'autre part que ces graphes ont la propriété remarquable de pouvoir être reconnus uniquement à partir de la séquence des degrés de leurs sommets.

Cette propriété a été étendue aux graphes $(2K_2, C_4)$ -libres ou **graphes pseudo-scindés** (*pseudo-split graphs*) par Maffray et Preissmann [264], qui ont montré qu'on peut obtenir un stable maximum à partir de l'algorithme de reconnaissance sans travail supplémentaire. Il est également possible de montrer que ces graphes possèdent au plus n stables maximaux (et donc autant de cliques maximales, par autocomplémentarité). Fouquet, Giakoumakis, Maire et Thuillier [138] ont montré que ceci reste vrai dans les **graphes $(2K_2, \text{maison})$ -libres** ; on peut par conséquent trouver un stable maximum en appliquant l'un des algorithmes évoqués à

50. Dans [133], Földes et Hammer donnent une définition plus générale, qui inclut aussi les bipartis (donc les graphes partitionnés en deux stables) et leurs complémentaires (graphes partitionnés en deux cliques). Dans [134], ils utilisent la définition ci-dessus, et c'est cette définition qui a été retenue ensuite dans la littérature.

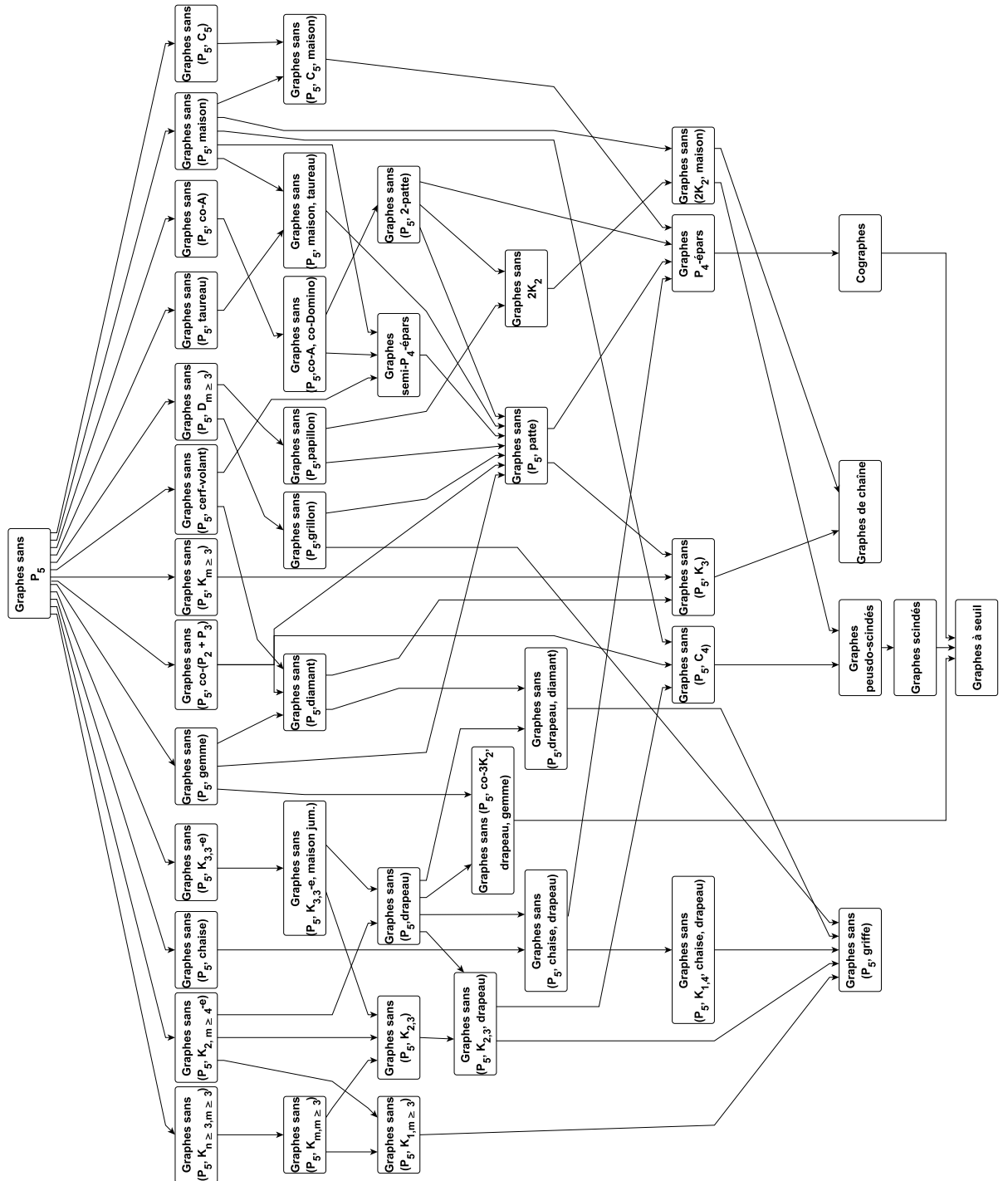


FIGURE 2.23 – Diagramme de relation d'inclusion entre différentes classes de graphes sans P_5 .

la section 2.1.4. On trouve également dans cet article une caractérisation des graphes (P_5, C_4) -libres (qui généralisent eux aussi les graphes peuscindés) à partir de laquelle on peut dériver un algorithme polynomial pour MIS.

D'une part, ces deux dernières classes sont contenues dans celle des graphes (P_5, maison) -libres, pour lesquels Giakoumakis et Rusu [160] ont obtenu, en utilisant la décomposition modulaire, un algorithme polynomial permettant de calculer un stable de poids maximum. D'autre part, les graphes $(2K_2, \text{maison})$ -libres sont aussi évidemment des graphes sans $2K_2$. Or Farber [130] a montré que les graphes sans $2K_2$ possèdent au plus $\binom{n}{2}$ stables maximaux; ce résultat a ensuite été généralisé aux graphes sans mK_2 (m fixé), qui ont $\mathcal{O}(n^{2m-2})$ stables maximaux [4, 131].

Les autres branches de cet arbre généalogique ne font apparaître que les classes présentées précédemment, et nous ne reviendrons donc pas sur ces sujets.

CONCLUSION DU CHAPITRE

En raison de deux résultats fondamentaux dus à Alekseev (les théorèmes 2.10 et 2.11), la classe des graphes sans P_5 est la plus petite classe de graphes définie par un seul sous-graphe connexe interdit pour laquelle on ne sait pas encore si le problème du stable maximum peut être résolu en temps polynomial.

De nombreux chercheurs se sont penchés sur la question, et ont apporté des solutions pour un grand nombre de sous-classes de graphes sans P_5 (en fait, même une infinité). Ils ont pour cela créé, modifié, utilisé des outils fondamentaux pour résoudre le problème du stable maximum. Nous en avons présenté quatre dans ce chapitre : les graphes augmentants, la décomposition modulaire, la structure et la décomposition par cliques d'articulation. Des résultats relativement récents liés à d'autres branches des mathématiques, comme la notion de largeur de clique, ont permis d'améliorer la complexité de certains algorithmes existants et apportent une vision nouvelle sur le problème. Afin de mieux comprendre comment de tels outils peuvent être utilisés, nous avons dressé un état de l'art très complet des classes de graphes sans P_5 pour lesquelles on sait aujourd'hui résoudre M(W)IS en temps polynomial. Il en existe néanmoins certaines que nous n'avons pas encore données, mais nous allons combler cette lacune grâce aux deux propositions suivantes :

Lemme 2.5 *Pour tout graphe G , si l'on note $\alpha_w(G)$ la valeur d'un stable de poids maximum, alors $\alpha_w(G) = \max\{w(v) + \alpha_w(G[A(v)]) \mid v \in V\}$.*

Ce lemme est trivial, mais permet de réduire MWIS au même problème sur le non-voisinage de chaque sommet. Autrement dit, si on sait résoudre MWIS pour un graphe H -libre en temps polynomial, on sait aussi le résoudre pour un graphe $H + K_1$ -libre en temps polynomial. Lozin et Mosca [256] ont montré que dans plusieurs classes, dont les graphes sans P_5 , on peut étendre cette propriété au non-voisinage d'une arête :

Théorème 2.42 (Lozin et Mosca (2005)) *Soit G un graphe sans P_5 . On peut résoudre MWIS en temps polynomial dans G si on sait le résoudre en temps polynomial dans $G[V_G \setminus N[\{u, v\}]]$ pour toute arête $\{u, v\}$ de G .*

Autrement dit, si l'on sait résoudre MWIS dans un graphe (P_5, H) -libre en temps polynomial, on peut le résoudre aussi en temps polynomial dans un graphe $(P_5, H + K_2)$ -libre (et donc, par induction, dans tout graphe $(P_5, H + mK_2)$ -libre avec m fixé). De plus, en vertu du théorème 2.32, nous savons résoudre le problème dans un graphe (P_5, H) -libre si H contient un sommet dominant. C'est alors un simple exercice de voir que les seuls graphes H d'ordre 5 ne vérifiant aucune de ces propriétés sont : la chaise, le cerf-volant, la maison, $K_{2,3}$, le drapeau, la 2-patte, le taureau, $\overline{P_2 + P_3}$ et C_5 . Nous avons vu au cours de ce chapitre que MWIS a une solution polynomiale dans les graphes ne contenant ni l'un des huit premiers, ni P_5 . Ainsi :

Observation 2.1 *Le problème du stable de poids maximum peut être résolu en temps polynomial dans tout graphe (P_5, H) -libre si H est un graphe d'ordre 5 non isomorphe à C_5 . Pour les graphes (P_5, C_5) -libres, la question de l'existence d'un algorithme polynomial est à ce jour ouverte.*

Soulignons une nouvelle fois que même si un certain nombre de résultats sont généralisés par les propositions précédentes, les algorithmes présentés pour chaque classe dans ce chapitre ont souvent une complexité bien moindre, et conservent donc toute leur importance.

Similarité avec le problème du dominant stable minimum

Un résultat similaire au théorème d'Alekseev a été prouvé par Boliac et Lozin [33] pour le problème du dominant stable (de poids) minimum $M(W)IDS$:

Théorème 2.43 (Boliac, Lozin (2003)) *Soit \mathcal{C} une classe de graphes définie par une famille finie \mathcal{F} de sous-graphes induits interdits. Si \mathcal{F} ne contient aucun graphe dont toutes les composantes connexes sont isomorphes à $S_{i,j,k}$ (figure 2.1) ou aucun graphe dont toutes les composantes connexes sont isomorphes à $T_{i,j,k}$ (figure 2.24), alors $M(W)IDS$ est \mathcal{NP} -difficile dans la classe \mathcal{C} .*

Pour les classes *monogènes*, c'est-à-dire définies par un seul sous-graphe interdit H , il découle de ce théorème que MWIDS peut être résolu en temps polynomial *seulement* si toutes les composantes de H sont des chaînes. On sait le résoudre dans les cograves et les graphes sans mK_2 (en utilisant respectivement le coarbre et le nombre de stables maximaux borné par un polynôme), et Lozin et Mosca [256] ont résolu MWIS et MWIDS dans la classe des graphes $(P_2 + P_3)$ -libres (alors que dans cette classe le problème du dominant minimum est \mathcal{NP} -difficile [28], et que MIDS est également \mathcal{NP} -difficile dans la classe des graphes sans $2P_3$

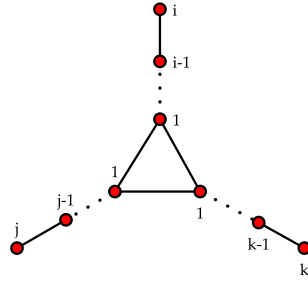


FIGURE 2.24 – Le graphe $T_{i,j,k}$.

[352]). Par conséquent, **pour les deux problèmes, la seule classe mono-gène (H étant connexe ou non), minimale pour laquelle on ne connaît pas la complexité est la classe des graphes sans P_5** (Boliac et Lozin ont néanmoins montré que MWIDS peut être résolu en temps polynomial dans les graphes $(P_5, \text{taureau})$ -libres et dans les graphes $(P_5, \text{co-A, co-domino})$ -libres). Un élément de réponse est peut-être apporté par le théorème 2.42 et son équivalent pour MWIDS :

Théorème 2.44 (Lozin et Mosca (2005)) *Soit G un graphe sans P_5 . On peut résoudre MWIDS en temps polynomial dans G si on sait le résoudre en temps polynomial dans $G[V_G \setminus N[\{u, v\}]]$ pour toute arête $\{u, v\}$ de G .*

Extension aux graphes sans P_6 , sans P_7 ...

Depuis quelques années, on observe une tendance naturelle à la généralisation aux graphes sans P_6 , sans P_7 , et au-delà, des résultats connus pour les graphes sans P_5 .

Ainsi, Mosca [283] a prouvé que l'on peut résoudre en temps polynomial MWIS dans les graphes $(P_6, \text{triangle})$ -libres et même dans les graphes (P_6, patte) -libres (en exploitant la structure de ces graphes), et MIS dans les graphes (P_6, C_4) -libres (en utilisant les graphes augmentant); pour cette classe, la version pondérée du problème a été résolue par Brandstädt et Hoàng [42, 43] (à l'aide des cliques d'articulation). Un algorithme polynomial pour MIS existe aussi pour les graphes $(S_{1,1,3}, K_{1,n})$ -libres [157]. Mosca [287] a montré ensuite (en utilisant la décomposition modulaire) que MWIS et MWIDS sont résolubles en temps polynomial dans la classe des graphes $(P_6, \text{diamant})$ -libres.

Alekseev et Lozin [11] ont résolu, par un argument augmentant, MIS dans les graphes $(P_7, \text{drapeau})$ -libres. Ce résultat a été étendu aux graphes $(P_8, \text{drapeau})$ -libres [156] et aux graphes $(S_{1,2,4}, \text{drapeau})$ -libres [156, 204] (avec tout de même une complexité en $\mathcal{O}(n^{11})$ pour ces derniers). Enfin, Mosca [286] a donné une solution au problème du stable de poids maximum dans la classe des graphes $(P_7, \text{drapeau})$ -libres.

Un algorithme sous-exponentiel

Avec la croyance désormais quasi unanime que $\mathcal{P} \neq \mathcal{NP}$, la mise au point d'algorithmes superpolynomiaux (*i.e.* dont le temps d'exécution n'est pas borné par un polynôme) rapides (c'est-à-dire plus efficaces que des algorithmes exponentiels) devient une tâche importante. Très récemment, en 2010, Randerath et Schiermeyer [302] ont conçu le premier algorithme *sous-exponentiel*⁵¹ permettant de résoudre MIS dans les graphes sans P_5 ; cet algorithme repose sur la structure des graphes sans P_5 décrite dans le théorème 2.13.

⁵¹. La définition d'un algorithme sous-exponentiel ne fait pas encore consensus dans la communauté. Nous renvoyons à [302] pour la définition utilisée par Randerath et Schiermeyer.

GRAPHES SANS P_5 3-COLORABLES

3

SOMMAIRE

3.1	À PROPOS DES GRAPHES SANS P_5 k -COLORABLES...	111
3.1.1	k -colorabilité des graphes sans P_5	111
3.1.2	Stable de poids maximum dans les graphes sans P_5 k -colorables	113
3.1.3	Caractérisation des graphes sans P_5 3-colorables	116
3.2	ALGORITHMES LINÉAIRES POUR LES GRAPHES SANS P_5 3-COLORABLES	116
3.2.1	Structures de données utilisées	117
3.2.2	Initialisation	118
3.2.3	Cordalité de \overline{G}	119
3.2.4	Cas où G contient un $2K_2$	120
3.2.5	Cas où G contient un C_5	125
3.2.6	Cas où G contient un \overline{C}_6	145
	CONCLUSION	150

Ce troisième chapitre est dédié à l'étude des graphes sans P_5 qui ont en plus la propriété d'être 3-colorables. Nous caractérisons ces graphes par une liste de douze sous-graphes induits interdits, et donnons une description complète de leur structure. Cette structure nous permet de dériver un algorithme de complexité *linéaire* pour tester l'appartenance d'un graphe à cette classe, calculer un stable de poids maximum et trouver une 3-coloration. Ceci répond à un problème laissé ouvert par Bruce, Hoàng et Sawada quant à l'existence d'un algorithme de reconnaissance des graphes sans P_5 3-colorables de complexité au plus $\mathcal{O}(n^4)$, et diminue la complexité de l'algorithme du stable de poids maximum dans toutes les classes de graphes sans P_5 k -colorables, avec $k \geq 3$. Ce travail a fait l'objet d'un rapport de recherche [263], a été présenté à la 8^{ème} French Combinatorial Conference à Orsay en juillet 2010, ainsi qu'au 12^{ème} congrès de la ROADEF à Saint-Étienne en mars 2011, et a été soumis pour publication à *SIAM Journal of Discrete Mathematics*.

3.1 À PROPOS DES GRAPHES SANS P_5 k -COLORABLES...

3.1.1 k -colorabilité des graphes sans P_5

Le problème de la k -colorabilité¹ ($k \geq 2$ fixé) d'un graphe est central en théorie des graphes, et a de très nombreuses applications [114]. C'est aussi l'un des plus difficiles : il est en général \mathcal{NP} -complet dès que $k \geq 3$ [226], et le reste pour les graphes planaires de degré au plus 4 [152], les graphes sans triangle [265] ou les graphes adjoints [214]. Il est même difficile d'approximer le nombre chromatique d'un graphe [230]. En revanche, le problème peut être résolu en temps polynomial dans les graphes parfaits grâce à la méthode de l'ellipsoïde [178, 179]; des algorithmes combinatoires et plus efficaces existent pour plusieurs sous-classes de graphes parfaits, notamment les graphes de comparabilité [129], les graphes triangulés [153], ou les graphes faiblement triangulés² [195].

Ces dernières années, les recherches se sont orientées vers la question de la k -colorabilité des graphes sans P_t . Pour $t = 3$ ou 4, on sait qu'il existe des algorithmes très efficaces (de complexité linéaire) qui permettent de dire si un graphe est k -colorable, quelle que soit la valeur de k (cf. section 2.4.4). D'un autre côté, Woeginger et Sgall [344] ont prouvé que la 5-colorabilité des graphes sans P_8 et la 4-colorabilité des graphes sans P_{12} sont \mathcal{NP} -complètes. Différentes équipes ont ensuite montré que la 4-colorabilité reste \mathcal{NP} -complète pour les graphes sans P_9 [242], et même pour les graphes sans P_8 [55]. Il a été aussi établi dernièrement que la 6-colorabilité des graphes sans P_7 est \mathcal{NP} -complète [54].

Au sujet des graphes sans P_6 , Sumner [323] avait prouvé qu'ils sont 3-colorables s'ils sont également sans C_6 . Cette propriété n'est pas généralisable à tous les graphes sans P_6 (voir figure 3.1), et un résultat de Randerath et Schiermeyer [301] affirme que l'on peut décider en temps polynomial si un graphe sans P_6 est 3-colorable ou non.

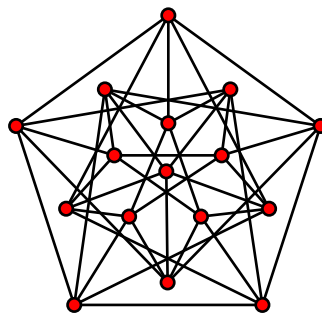


FIGURE 3.1 – Le graphe de Clebsch, un graphe sans P_6 , mais non 3-colorable.

Examinons à présent notre classe de graphes favorite. Il est bien connu qu'un graphe sans P_5 et sans triangle est 3-colorable [323]. Là encore, cette propriété ne peut se généraliser à la classe entière des graphes sans

1. Nous renvoyons le lecteur à la section 1.1.6 pour les définitions concernant la coloration d'un graphe.

2. Ces graphes ont été définis par Hayward [196] comme étant les graphes sans trou de longueur au moins 5 ni antitrou de longueur au moins 5; il est donc évident qu'ils sont parfaits et que cette classe contient les graphes triangulés et leurs complémentaires.

$k \setminus t$	3	4	5	6	7	8	9	...
3	Lin	Lin	Lin	Pol	?	?	?	...
4	Lin	Lin	Pol	?	?	\mathcal{NP} -c	\mathcal{NP} -c	...
5	Lin	Lin	Pol	?	?	\mathcal{NP} -c	\mathcal{NP} -c	...
6	Lin	Lin	Pol	?	\mathcal{NP} -c	\mathcal{NP} -c	\mathcal{NP} -c	...
7	Lin	Lin	Pol	?	\mathcal{NP} -c	\mathcal{NP} -c	\mathcal{NP} -c	...
...

FIGURE 3.2 – Connaissances actuelles sur la complexité du problème de la k -colorabilité des graphes sans P_t .

P_5 , et l'un des résultats que nous avons justement établis, et que **nous montrerons** à la section 3.2, est **qu'il existe une famille de douze graphes minimaux sans P_5 qui ne sont pas 3-colorables**. Plusieurs équipes ont prouvé que l'on peut décider en temps polynomial si un graphe sans P_5 est 3-colorable ou non [276, 303, 344]; **nous montrerons qu'il est possible de répondre à cette question en temps linéaire**. Hormis le fait que la question de la 4-colorabilité des graphes sans P_5 et sans C_5 peut être résolue en temps polynomial [242], on n'en savait pas beaucoup plus sur k -colorabilité des graphes sans P_5 , jusqu'à ce que Hoàng et al. [211] démontrent récemment que pour tout k fixé, il existe un algorithme polynomial qui décide si un graphe de cette classe admet une k -coloration, et qui fournit une telle coloration en cas de réponse positive.

Le tableau 3.2, que l'on retrouve complété au fur et à mesure dans la plupart des articles cités, donne une synthèse des résultats précédents. Il s'agit à notre connaissance de la version la plus complète à ce jour.

Ces résultats sont à mettre en contraste avec la \mathcal{NP} -difficulté d'évaluer le nombre chromatique, $\chi(G)$ (ce qui revient au problème de la *colorabilité*, *i.e.* lorsque k n'est pas fixé, mais fait aussi partie de l'instance), dans bon nombre de ces classes, dont les graphes sans P_5 . Ce fait est mis en évidence par un théorème très fort de Kral, Kratochvil, Tuza et Woeginger [234] :

Théorème 3.1 (Kral, Kratochvil, Tuza et Woeginger (2001)) *Le problème de la colorabilité dans un graphe H -libre est décidable en temps polynomial si H est un sous-graphe induit de P_4 ou de $P_3 \oplus K_1$ (où \oplus représente l'union disjointe des deux graphes), et est \mathcal{NP} -complet pour tout autre graphe H .*

Ainsi, il semble peu probable qu'il existe un algorithme permettant de calculer $\chi(G)$ en temps polynomial si G est un graphe sans P_5 (et même si G est un graphe sans $2K_2$). Le théorème précédent peut encore être renforcé aux graphes définis par plusieurs sous-graphes induits interdits. Pour l'énoncer, nous avons besoin de définir quatre types de graphes :

Type A les graphes qui contiennent un cycle,

Type B les graphes qui contiennent une griffe,

Type C les graphes qui contiennent un sous-graphe partiel de $2K_2$ en tant que sous-graphe induit,

Type D les sous-graphes induits de P_4 ou de $P_3 \oplus K_1$.

Théorème 3.2 (Kral, Kratochvil, Tuza et Woeginger (2001)) *Soit \mathcal{F} une famille finie de graphes.*

1. *si au moins un membre de \mathcal{F} est de type D , alors on peut calculer $\chi(G)$ en temps polynomial pour les graphes \mathcal{F} -libres ;*
2. *si tous les graphes de \mathcal{F} sont du même type A , B ou C , le problème est \mathcal{NP} -complet.*

Nous terminons ce paragraphe en mentionnant un résultat concernant la listes-colorabilité des graphes sans P_5 .

Définition 3.1

Une **coloration par listes** (list coloring) d'un graphe G est une coloration où chaque sommet ne peut recevoir une couleur que parmi celles figurant dans une liste qui lui est associée.

Un graphe est dit **k -listes-colorable**, ou **k -choisissable** (k -choosable), s'il admet une coloration quelles que soient les listes, toutes de taille k , affectées aux sommets.

Le **nombre chromatique par listes** ou **nombre de choisissabilité** de G , noté $ch(G)$, est le plus petit k tel que G soit k -listes-colorable.

La coloration par liste a été introduite par Vizing [340] et par Erdős, Rubin et Taylor [125]. Naturellement, la coloration telle que nous l'avons rencontrée jusqu'ici n'est qu'un cas particulier de la coloration par listes, où tous les sommets ont la même liste de couleurs (potentiellement au nombre de $|V|$). En particulier, pour tout graphe G , $\chi(G) \leq ch(G)$. Par exemple, le graphe suivant est 2-colorable, mais pas 2-listes-colorable.

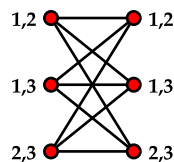


FIGURE 3.3 – Un exemple de graphe 2-colorable, mais pas 2-listes-colorable.

Il est \mathcal{NP} -complet de décider si un graphe est k -listes-colorable, avec $k \geq 3$ (puisque le problème de la k -colorabilité est déjà \mathcal{NP} -complet, mais on sait par exemple que pour tout graphe planaire G , $ch(G) \leq 5$ [332]. Très récemment, Golovach et Heggernes [163] ont montré que le problème de la k -listes-colorabilité (avec k faisant partie de l'instance) des graphes sans P_5 est \mathcal{NP} -complet, mais si k est fixé, il peut être résolu par un algorithme polynomial, ou plus exactement un algorithme dit à *complexité paramétrée* (FPT algorithm, pour Fixed Parameter Tractable), c'est à dire un algorithme de complexité $\mathcal{O}(n^c \times f(k))$ où f est une fonction quelconque (indépendante de n) et c une constante fixée.

3.1.2 Stable de poids maximum dans les graphes sans P_5 k -colorables

Au chapitre précédent, nous avons vu une proposition (la proposition 2.6) qui énonce qu'un graphe connexe *biparti* sans P_5 d'ordre n possède au

plus n stables maximaux. Il suffit donc d'examiner ces n stables pour en trouver un qui soit de cardinalité ou de poids maximum.

Proposition 3.1 *On peut calculer un stable (de poids) maximum dans un graphe sans P_5 biparti en temps $\mathcal{O}(n)$.*

La proposition 2.6 affirme également qu'on peut toujours trouver, dans un graphe sans P_5 biparti connexe, un sommet tel que son non-voisinage induit un stable. Maffray [260] a montré que ces deux propriétés sont en fait généralisables à tous les graphes sans P_5 :

Théorème 3.3 (Maffray (2009)) *Soit G un graphe sans P_5 k -colorable ($k \geq 2$) d'ordre n . G contient un sommet dont les non-voisins induisent un sous-graphe $(k - 1)$ -colorable.*

La preuve de ce théorème est essentiellement un algorithme, appelé FIND_k , qui étant donné un graphe connexe sans P_5 et une k -coloration C_1, \dots, C_k de G , trouve le sommet désiré :

- si $k = 2$, il suffit de renvoyer un sommet de plus grand degré (ceci se fait en temps $\mathcal{O}(n^2)$);
- si $k \geq 3$ et $G \setminus C_1$ n'est pas connexe, on contracte ses composantes connexes B_1, \dots, B_t en un ensemble de sommets $B = \{b_1, \dots, b_t\}$ et on construit en $\mathcal{O}(n^2)$ un graphe biparti $H = (C_1, B; E_H)$ où $E_H = \{\{c, b_i\} : c \in C_1, b_i \in B \text{ et il existe un sommet de } B_i \text{ adjacent à } c\}$; ce graphe est sans P_5 et il suffit de renvoyer un sommet de C_1 de plus grand degré;
- si $k \geq 3$ et G est connexe, on applique FIND_{k-1} sur $G \setminus C_1$.

Comme la dernière étape est répétée au plus $k - 2$ fois, la complexité de cet algorithme est $\mathcal{O}(n^2)$ si $k = 2$ et $\mathcal{O}(n^3)$ si $k \geq 3$. De plus, on peut faire appel à l'algorithme polynomial de Hoàng et al. [211] cité précédemment pour obtenir une k -coloration si celle-ci n'est pas fournie en entrée de l'algorithme.

On peut toutefois noter deux différences entre la proposition 2.6 et le théorème 3.3 :

1. dans le cas des graphes bipartis ($k = 2$, proposition 2.6), chaque classe de couleur contient un sommet dont les non-voisins forment un stable, *i.e.* induisent un sous-graphe $(k - 1)$ -colorable. Ce n'est plus vrai si G n'est pas biparti : si l'on considère par exemple deux cliques disjointes de taille $k \geq 3$ reliées par une arête $\{x, y\}$ où x est de couleur 1 et y de couleur 2, G est clairement sans P_5 , mais les classes de couleur $3, \dots, k$ ne contiennent aucun sommet dont le non-voisinage induit un sous-graphe $(k - 1)$ -colorable;
2. dans le cas des graphes bipartis, tout sommet de plus grand degré de chaque classe a un voisinage qui induit un sous-graphe $(k - 1)$ -colorable. Ce n'est plus vrai dans le cas général : considérons le graphe $G = (\{u, v, w, x, y, z_1, z_2, z_3\}, \{uv, uw, vw, wx, xy, z_i x, z_i y (i =$

$1, 2, 3\}$) et la coloration suivante : w et y sont de couleur 1, u et x de couleur 2 et v, z_1, z_2, z_3 de couleur 3. Ce graphe est sans P_5 , mais les voisins de y , qui est le sommet de plus degré de couleur 1, n'induisent pas un sous-graphe biparti.

Une conséquence importante du théorème précédent est qu'on peut trouver un stable de poids maximum en temps polynomial dans un graphe sans P_5 k -colorable.

Théorème 3.4 (Maffray (2009)) *Pour $k \geq 1$ fixé, il existe un algorithme de complexité $\mathcal{O}(n^k)$ qui trouve un stable de poids maximum dans tout graphe sans P_5 k -colorable.*

Mais c'est la preuve de ce théorème, par récursivité, qui permet d'obtenir le résultat très intéressant qui suit :

Théorème 3.5 (Maffray (2009)) *S'il existe un algorithme de complexité $\mathcal{O}(n^c)$ (où c est une constante) pour résoudre MWIS dans les graphes sans P_5 $(k-1)$ -colorables, alors il existe un algorithme de complexité $\mathcal{O}(n^{c+1})$ pour résoudre MWIS dans les graphes sans P_5 k -colorables.*

L'intérêt de ce résultat se situe à plusieurs niveaux. Tout d'abord, d'après le théorème 3.4, nous savions que MWIS peut être résolu par un algorithme de complexité $\mathcal{O}(n^3)$ dans la classe des graphes sans P_5 3-colorables. Or, puisqu'on peut le résoudre en temps $\mathcal{O}(n)$ dans les graphes sans P_5 bipartis, on peut finalement, d'après le théorème précédent, le résoudre en temps $\mathcal{O}(n^2)$ dans la classe des graphes sans P_5 3-colorables. Ensuite, ce théorème signifie que si l'on réussit à abaisser la complexité de l'algorithme pour la classe des graphes sans P_5 k -colorables, alors cette amélioration se répercute sur la complexité des algorithmes résolvant le problème dans la classe des graphes sans P_5 l -colorables, pour tout $l > k$. Or, nous avons été en mesure d'établir le résultat suivant :

Théorème 3.6 (Maffray, Morel (2011)) *On peut résoudre MWIS en temps linéaire dans la classe des graphes sans P_5 3-colorables.*

et par conséquent, nous avons réduit d'une unité la complexité de l'algorithme pour tout $k \geq 3$.

Notons pour terminer que ces résultats s'appliquent à la classe des graphes (P_5, K_{k+1}) -libres, qui contient clairement les graphes sans P_5 k -colorables. En effet, Gyárfás [180] a prouvé que pour tout $k > 0$, il existe une constante c_k (ne dépendant que de k) telle que tout graphe (P_5, K_{k+1}) -libre est c_k -colorable; par conséquent, il existe un algorithme de complexité $\mathcal{O}(n^{c_k})$ qui trouve un stable de poids maximum dans un graphe sans P_5 et sans K_{k+1} d'ordre n . Gyárfás a montré que $c_k \leq 4^{k-1}$, puis Gravier, Hoàng et Maffray [176] ont prouvé que $c_k \leq 3^{k-1}$.

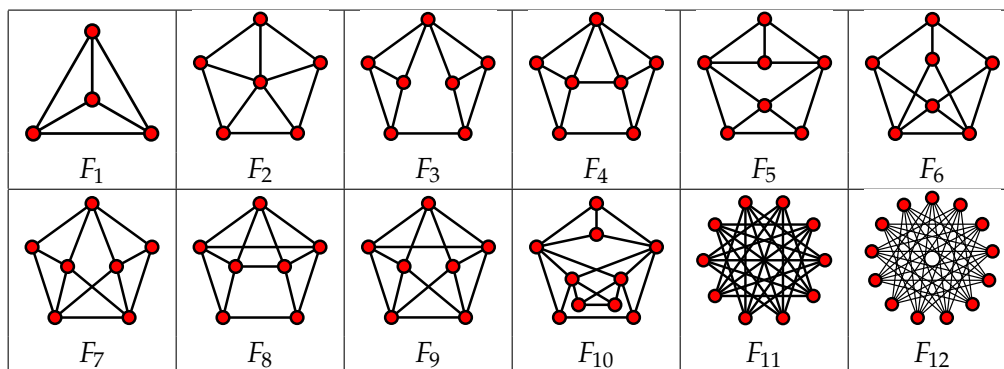


FIGURE 3.4 – Les 12 sous-graphes minimaux sans P_5 non 3-colorables.

Bien entendu, les résultats précédents ne sont valides que lorsque k est fixé, et ne conduisent donc pas à un algorithme polynomial de résolution du problème du stable maximum dans la classe générale des graphes sans P_5 .

3.1.3 Caractérisation des graphes sans P_5 3-colorables

La classe \mathcal{C} des graphes sans P_5 3-colorables est héréditaire : tout sous-graphe induit d'un graphe sans P_5 3-colorable est lui-même sans P_5 3-colorable. Il est donc possible de caractériser cette classe par une unique famille \mathcal{F} de sous-graphes induits minimaux *interdits*, c'est-à-dire qui soit contiennent un P_5 soit ne sont pas 3-colorables. Nous avons montré que cette famille \mathcal{F} est constituée de douze graphes :

Théorème 3.7 (Maffray, Morel (2011)) *Un graphe sans P_5 est 3-colorable si et seulement s'il ne contient aucun des douze sous-graphes F_1, \dots, F_{12} de la figure 3.4 en tant que sous-graphe induit.*

Nous prouverons ce théorème dans la section suivante. Mentionnons que Bruce, Hoàng et Sawada [59] ont aussi donné cette caractérisation au même moment, mais sous une forme légèrement différente (leur liste comprend seulement six graphes, mais non nécessairement induits). Signalons également que leur article se termine sur la question de l'existence d'un algorithme de reconnaissance des graphes sans P_5 3-colorables qui serait de complexité $\mathcal{O}(n^4)$ ou meilleure. Or nous allons voir que :

Théorème 3.8 (Maffray, Morel (2011)) *On peut décider en temps linéaire si un graphe est sans P_5 3-colorable.*

3.2 ALGORITHMES LINÉAIRES POUR LES GRAPHES SANS P_5 3-COLORABLES

Nous allons à présent démontrer les théorèmes 3.6, 3.7 et 3.8. La preuve, commune, est en fait un algorithme qui reçoit en entrée un graphe G_0 et qui retourne, en temps linéaire, l'une des sorties suivantes :

- un sous-graphe interdit de G_0 qui est soit un P_5 (auquel cas G_0 n'était donc pas sans P_5), soit un graphe de \mathcal{F} (auquel cas G_0 n'était donc pas 3-colorable) ;
- une certaine structure de G_0 à partir de laquelle on peut, en temps linéaire, tester s'il est 3-colorable, décider s'il est sans P_5 , et calculer un stable de poids maximum.

Grosso modo, l'algorithme fonctionne de la façon suivante : il commence par éliminer les ensembles homogènes du graphe. Ensuite, il teste si le complémentaire du graphe est cordal ; et s'il l'est, notre tâche est terminée. Si le complémentaire du graphe n'est pas cordal, l'algorithme trouve un ensemble de sommets qui induisent un trou dans ce complémentaire et continue de construire le graphe entier autour de cet ensemble. Le reste de cette section donne une présentation précise de l'algorithme et une preuve de sa correction.

3.2.1 Structures de données utilisées

Tout au long de l'algorithme, nous aurons à gérer les situations suivantes dans un graphe d'ordre n .

(a) Étant donnés des sous-ensembles deux à deux disjoints X_1, \dots, X_ℓ de V_G , dont la taille peut croître pendant l'exécution de l'algorithme, on veut savoir, pour chaque sommet $x \in V_G \setminus (X_1 \cup \dots \cup X_\ell)$, s'il est anticomplet à X_i ou non, pour tout $i \in \{1, \dots, \ell\}$. Cette information peut être mémorisée par un compteur $c_i(x)$, respectivement égal à 0 ou 1. Le vecteur $(c_1(x), \dots, c_\ell(x))$ est appelé le *type* de x par rapport à la famille X_1, \dots, X_ℓ . Comme ℓ sera toujours une constante, il n'y aura qu'un nombre constant de types différents. Chaque sommet de $V_G \setminus (X_1 \cup \dots \cup X_\ell)$ est placé dans un ensemble Q_t associé au type t de x . A chaque fois qu'un sommet $u \in V_G \setminus (X_1 \cup \dots \cup X_\ell)$ est ajouté à un ensemble X_i , on parcourt la liste d'adjacence de u et, pour chaque voisin x de u , nous mettons son compteur $c_i(x)$ à 1 et déplaçons x dans l'ensemble associé à son nouveau type. Ceci se fait en $\mathcal{O}(d(u))$. On peut aussi décider s'il existe un sommet d'un type t donné en vérifiant si l'ensemble Q_t correspondant est vide ou non, et on peut obtenir tous les tels sommets en temps $\mathcal{O}(|Q_t|)$.

(b) Étant donné un ensemble $X \subset V_G$ (qui ne changera pas), on veut déterminer pour tout sommet $u \in V_G \setminus X$ s'il est complet à X , anticomplet à X , ou ni l'un ni l'autre. Ceci peut être fait de la façon suivante : pour tout sommet $x \in X$, on parcourt sa liste d'adjacence, et pour chaque voisin u de x on ajoute 1 à un compteur $c_X(u)$ (initialisé à 0). Il suffit alors de comparer la valeur finale de $c_X(u)$ à celle de $|X|$. Cette étape nécessite $\mathcal{O}(n + m)$ itérations, et sera exécutée seulement pour un nombre constant d'ensembles X , donc le temps total reste linéaire.

Ces deux remarques garantissent que chaque étape principale de notre algorithme a une complexité temporelle en $\mathcal{O}(n + m)$. Elles ne seront pas répétées explicitement par la suite.

3.2.2 Initialisation

Soit G_0 un graphe, d'ordre n et à m arêtes, donné en entrée de notre algorithme. On peut supposer que G_0 est connexe, sinon, il suffit de traiter chaque composante connexe séparément. Nous noterons \mathcal{M} la partition modulaire maximale de V_{G_0} (cf. proposition 2.18), et f la fonction de poids sur les sommets de G_0 .

Lemme 3.1 *Si G_0 a au moins deux sommets, et si un élément Y de \mathcal{M} induit un graphe qui n'est pas biparti, alors G_0 n'est pas 3-colorable.*

Démonstration. Comme G_0 est connexe, il existe un sommet $u \in V_{G_0} \setminus Y$ complet à Y . Si Y n'induit pas un sous-graphe biparti, alors il contient un ensemble de sommets Z qui induit un cycle impair, et le sous-graphe induit par $Z \cup \{u\}$ n'est donc pas 3-colorable. En fait, soit Z est de cardinalité au moins 7, auquel cas il contient cinq sommets qui induisent un P_5 , soit il est de cardinalité 3 ou 5 et alors $Z \cup \{u\}$ induit un F_1 ou un F_2 . \square

Notre algorithme teste si tous les éléments de \mathcal{M} induisent des sous-graphes bipartis. Comme ils sont disjoints, ceci se fait en temps $\mathcal{O}(n + m)$. Si l'un d'eux n'induit pas un sous-graphe biparti, l'algorithme retourne un P_5 ou un membre de \mathcal{F} comme montré dans la preuve précédente, et s'arrête. Supposons donc désormais que tout $Y \in \mathcal{M}$ induit un sous-graphe biparti. Soit G le graphe quotient de G_0 , obtenu en contractant chaque élément de \mathcal{M} . Lors de la contraction, si un élément Y de \mathcal{M} n'est pas un stable, on choisit un sommet y qui a un voisin \bar{y} dans Y , et on dira que y est un *sommet double* dans G (notons que par définition d'un module, y et \bar{y} ont même voisinage dans $V(G_0) \setminus Y$). Systématiquement, quand nous traiterons un sommet double dans la suite, nous nous rappellerons qu'il représente un sous-graphe biparti de G_0 comportant au moins une arête, et qu'en particulier il nécessite deux couleurs. Ainsi, G_0 est 3-colorable si et seulement si G admet une 3-coloration telle que (avec un léger abus de définition) chaque sommet double a deux couleurs, qui sont différentes des couleurs de ses voisins. Un sommet qui n'est pas double sera appelé *sommet simple*.

Lemme 3.2 *Les graphes G et G_0 satisfont les propriétés suivantes :*

- (i) *Soit G est une clique, soit c'est un graphe premier.*
- (ii) *G_0 est sans P_5 si et seulement si G et tous les sous-graphes $G_0[Y]$ ($Y \in \mathcal{M}$) sont sans P_5 .*
- (iii) *On peut déduire en temps linéaire un stable de poids maximum de G_0 à partir d'un stable de poids maximum de G ³.*

Démonstration. (i) découle de la propriété de la décomposition modulaire vues au chapitre 2.

³. Ce résultat figure également dans [254], mais la preuve y est plus compliquée que celle que nous donnons ici.

(ii) Si G_0 est sans P_5 , G est aussi sans P_5 puisque c'est un sous-graphe induit de G_0 . Réciproquement, supposons que G_0 contienne un P_5 d'ensemble de sommets X . Comme un P_5 ne contient pas d'ensemble homogène, soit X est inclus dans un membre Y de \mathcal{M} , donc $G_0[Y]$ contient un P_5 , soit X contient au plus un sommet de chaque membre de \mathcal{M} et donc X induit un P_5 dans G .

(iii) Soit $\{y_1, \dots, y_p\}$ l'ensemble des sommets de G , et Y_i l'élément de \mathcal{M} représenté par y_i pour chaque $i \in \{1, \dots, p\}$. Pour chaque $i \in \{1, \dots, p\}$, soit S_i un stable de poids maximum du sous-graphe $G_0[Y_i]$, avec pour fonction de poids la restriction de f à Y_i . Si S est un stable de G , alors l'ensemble $U(S) = \bigcup_{y_i \in S} S_i$ est un stable de G_0 . Réciproquement, tout stable de G_0 est inclus dans un ensemble de la forme $U(S) = \bigcup_{y_i \in S} T_i$ où T_i est un stable dans $G_0[Y_i]$. Il s'ensuit que si S est un stable de poids maximum dans G , $U(S)$ est un stable de poids maximum dans G_0 . \square

3.2.3 Cordalité de \overline{G}

Dans cette étape, nous testons si le complémentaire \overline{G} de G est cordal. Ceci provient du fait que nous souhaitons utiliser le théorème 2.39; ainsi si G contient un $2K_2$, on reconstruit le co- A ou le co-Domino qu'il contient nécessairement, puis on essaie de construire le reste du graphe autour de ce « squelette ». Le problème est qu'on ne connaît pas d'algorithme permettant de trouver un $2K_2$ dans un graphe autre que l'algorithme exhaustif, de complexité $\mathcal{O}(n^4)$. En revanche, il existe des algorithmes linéaires, de complexité $\mathcal{O}(n + m)$, permettant de décider si un graphe est cordal ou non [306, 329]; ces algorithmes sont basés sur des ordres dans les graphes cordaux, mais Tarjan et Yannakakis [330] ont montré que l'algorithme donné dans [329] permet de détecter un trou en temps $\mathcal{O}(n + m)$ si le graphe n'est pas cordal. L'idée était donc d'utiliser cet algorithme dans le complémentaire de G pour détecter la présence d'un C_4 (le complémentaire de $2K_2$). Ceci pose cependant deux problèmes : l'algorithme ne renvoie pas forcément un trou de longueur minimum, et surtout la construction de \overline{G} peut nécessiter $\mathcal{O}(n^2)$ itérations. Heureusement, Hoàng [209] a montré qu'on peut tester *directement* si G est le complémentaire d'un graphe cordal, toujours en temps $\mathcal{O}(n + m)$. La reconstruction d'un trou n'est pas donnée explicitement, mais peut être implémentée en temps linéaire de manière similaire à celle donnée dans [330]. Cette procédure peut se terminer de trois manières différentes :

- (1) \overline{G} n'est pas cordal. Alors l'algorithme de [209] retourne un sous-ensemble de sommets Z qui induit un trou dans \overline{G} . Si $|Z| = 4$, alors $G[Z]$ est un $2K_2$ et notre algorithme va à la section 3.2.4. Si $|Z| = 5$ ou 6, alors $G[Z]$ est un C_5 ou un \overline{C}_6 et notre algorithme va respectivement à la section 3.2.5 ou 3.2.6. Si $|Z| = 7$, notre algorithme retourne $G[Z]$, qui est un \overline{C}_7 (donc un membre de \mathcal{F} , car isomorphe à F_9) et s'arrête. Si $|Z| \geq 8$, alors $G[Z]$ contient un K_4 ; notre algorithme retourne ce K_4 (qui est aussi un membre de \mathcal{F}) et s'arrête.
- (2) \overline{G} est cordal, mais $\omega(G) \geq 4$ (l'algorithme de [209] permet de calculer une clique (de poids) maximum en temps linéaire). Dans ce cas, notre algorithme renvoie un K_4 et s'arrête.

- (3) \overline{G} est cordal, et $\omega(G) \leq 3$. Notre algorithme affiche le message « G est sans P_5 ». Ceci est correct parce que le complément d'un graphe cordal est sans P_5 (en effet, la maison, qui est le complémentaire du P_5 , contient un cycle de longueur 4). Il reste à savoir si G_0 est 3-colorable ou non. Pour cela, on définit une nouvelle fonction de poids en attribuant un poids 2 aux sommets doubles et un poids 1 aux sommets simples. Ensuite on cherche une clique de poids maximum par rapport à cette fonction. Clairement, G_0 est 3-colorable si et seulement si le poids d'une clique est au maximum 3. L'algorithme dans [209] permet également de calculer une coloration de poids minimum, c'est-à-dire une famille S_1, \dots, S_k de stables, de poids respectifs s_1, \dots, s_k telle que tout sommet x satisfait $w(x) \leq \sum \{s_i | S_i \ni x\}$. Il est prouvé dans [209] que la complexité de cet algorithme est bornée par $\sum_{i=1}^k |S_i|$. Comme nos sommets ont pour poids 1 ou 2, une simple analyse de l'algorithme montre que chaque S_i a pour poids 1 ou 2 et chaque sommet de G apparaît dans au plus deux des S_i . Par conséquent $\sum_{i=1}^k |S_i| \leq 2n$, donc la complexité est linéaire.

3.2.4 Cas où G contient un $2K_2$

À partir de ce point et afin d'alléger le texte, nous noterons H_1 et H_2 les graphes que nous notions respectivement co- A et co-Domino (figure 2.12). Comme nous l'avons indiqué au début de la section précédente, puisque G est sans P_5 mais contient en $2K_2$, en vertu du théorème 2.39, nous cherchons à reconstruire un sous-graphe de G isomorphe à H_1 ou à H_2 . Voici comment procède l'algorithme. Notons $\{a_1, a_2, b_1, b_2\}$ les sommets du $2K_2$ et $\{a_1a_2, b_1b_2\}$ ses arêtes, et soient $A = \{a_1, a_2\}$ et $B = \{b_1, b_2\}$.

Tant qu'il existe un sommet $x \in V_G \setminus (A \cup B)$ qui a un voisin dans A et pas de voisin dans B , ajouter x à A .

Tant qu'il existe un sommet $y \in V_G \setminus (A \cup B)$ qui a un voisin dans B et pas de voisin dans A , ajouter x à B .

Recommencer ces deux opérations tant que c'est possible.

Considérons à présent les ensembles A et B à la fin de cette procédure. Comme G est un graphe quotient, A n'est pas un ensemble homogène, donc il existe un sommet $x \in V_G \setminus A$ qui n'est ni complet ni anticomplet à A ; donc on peut trouver deux sommets adjacents a et a' dans A tels que x est adjacent à a mais pas à a' . Clairement, $x \notin B$, puisque les sommets de B sont anticomplets à A . De la même façon, il existe un sommet $y \in V_G \setminus B$ et deux sommets adjacents $b, b' \in B$ tels que y est adjacent à b et pas à b' , et $y \notin A$. Si x est adjacent à exactement un sommet parmi $\{b, b'\}$, alors $\{a, a', x, b, b'\}$ induit un P_5 , et l'algorithme s'arrête; si x n'est adjacent ni à l'un ni à l'autre, il a nécessairement un autre voisin c dans B (sinon, on aurait inclus x dans A), et il existe un plus court chemin P dans B entre c et $\{b, b'\}$. Alors $\{a, a', x\} \cup V_P$ contient un P_5 . Supposons maintenant que x est adjacent à b et à b' , et de la même façon, on peut supposer que y est adjacent à a et a' . Alors $\{a, a', b, b', x, y\}$ induit un H_1 ou un H_2 , selon que x et y sont adjacents ou non.

Il est facile de voir que cette procédure s'exécute en temps $\mathcal{O}(n + m)$ en appliquant ce que nous avons expliqué dans la section 3.2.1. Si l'algo-

rithme a trouvé un H_1 , il continue à la section 3.2.4 ; s'il a trouvé un H_2 , il continue à la section 3.2.4.

G contient un H_1

Soient a_1, \dots, a_6 six sommets de G qui induisent un H_1 , avec $E_{H_1} = \{a_1a_2, a_1a_3, a_2a_3, a_4a_5, a_4a_6, a_5a_6, a_2a_4, a_3a_5\}$. Notons $A' = \{a_1, a_2, a_3\}$, $A'' = \{a_4, a_5, a_6\}$ et $A = A' \cup A''$.

Lemme 3.3 Soit G ne contient que six sommets, soit il contient un P_5 ou un membre de \mathcal{F} .

Démonstration. Supposons que G ait au moins sept sommets. Comme G est connexe, il existe un sommet $u \in V_G \setminus A$ qui a au moins un voisin dans A . Si u est adjacent à tous les sommets de A' ou de A'' , G contient un K_4 . On peut donc supposer, à symétrie près, que u a un ou deux voisins dans A' .

Supposons tout d'abord que $N(u) \cap A' = \{a_1\}$. Si u n'est adjacent ni à a_4 ni à a_5 , alors $u-a_1-a_2-a_4-a_5$ induit un P_5 . Admettons donc, à symétrie près, que u soit adjacent à a_4 . Si $u \approx a_6$, alors $a_6-a_4-u-a_1-a_3$ est un P_5 ; inversement, si $u \sim a_6$, alors $u \approx a_5$ comme indiqué précédemment, mais alors $a_5-a_6-u-a_1-a_2$ est un P_5 .

Supposons à présent que $N(u) \cap A' = \{a_2\}$. Si u n'a pas de voisin dans $\{a_5, a_6\}$, alors $u-a_2-a_3-a_5-a_6$ est un P_5 induit, et si u a exactement un voisin parmi $\{a_5, a_6\}$, alors $\{u, a_1, a_2, a_5, a_6\}$ induit un P_5 . Si u est adjacent à la fois à a_5 et a_6 , alors comme mentionné précédemment, $u \approx a_4$. Donc $N(u) \cap A = \{a_2, a_5, a_6\}$. Soit $X = \{x \in V_G \mid N(x) \cap A = \{a_2, a_5, a_6\}\}$. Nous savons que $|X| \geq 2$ puisque a_4 et u sont dans X . Comme G est un graphe quotient, X n'est pas un ensemble homogène, et donc il existe des sommets $v \in V_G \setminus X$ et $x, y \in X$ tels que $v \sim x$ et $v \approx y$. Clairement, $v \notin A$. Si $v \approx a_5$, alors $v-x-a_5-a_3-a_1$ est un P_5 . Supposons donc que $v \sim a_5$. Si $v \approx a_2$, alors $v-a_5-y-a_2-a_1$ est un P_5 . Supposons donc que $v \sim a_2$. À présent, si $v \approx a_6$, alors $a_6-a_5-v-a_2-a_1$ est un P_5 ; et si $v \sim a_6$, $\{v, x, a_5, a_6\}$ est un K_4 . La preuve est similaire si $N(u) \cap A' = \{a_3\}$.

Nous pouvons à présent supposer que u a deux voisins dans A' . Par symétrie, on peut supposer que u a soit zero, soit deux voisins dans A'' .

Admettons que $N(u) \cap A' = \{a_1, a_2\}$. Si u est adjacent à la fois à a_4 et a_6 , alors $A \cap \{u\}$ induit un F_4 . Si u est adjacent soit à a_4 , soit à a_6 , alors $\{u, a_1, a_3, a_4, a_6\}$ induit un P_5 . Supposons donc que u ne soit adjacent ni à a_4 , ni à a_6 . Comme u a zero ou deux voisins dans A'' , il n'est pas adjacent à a_5 . Alors $u-a_2-a_3-a_5-a_6$ est un P_5 . La preuve est similaire lorsque $N(u) \cap A' = \{a_1, a_3\}$.

Nous pouvons à présent supposer que $N(u) \cap A' = \{a_2, a_3\}$ et, par symétrie, $N(u) \cap A''$ est soit égal à $\{a_4, a_5\}$ soit vide. Dans le premier cas, $a_1-a_2-u-a_5-a_6$ est un P_5 . Dans le second cas, on a $N(u) \cap A = \{a_2, a_3\}$. Soit alors $X = \{x \in V_G \mid N(x) \cap A = \{a_2, a_3\}\}$. Nous savons que $|X| \geq 2$ puisque a_1 et u sont dans X . Comme G est un graphe quotient, X n'est pas un ensemble homogène, donc il existe des sommets $v \in V_G \setminus X$ et $x, y \in X$ tels que $v \sim x$ et $v \approx y$. Clairement, $v \notin A$. Si v est adjacent à un sommet $a \in \{a_5, a_6\}$, alors $a-v-x-a_2-y$ est un P_5 . Donc admettons que $v \approx a_5$, $v \approx a_6$

et par symétrie $v \approx a_4$. Si $vv \approx a_2$, alors $v-x-a_2-a_4-a_6$ est un P_5 . Donc $v \sim a_2$ et par symétrie $v \sim a_3$. Mais alors $\{v, x, a_2, a_3\}$ induit un K_4 , ce qui termine la preuve du lemme. \square

Si l'algorithme n'a trouvé ni un P_5 ni un membre de \mathcal{F} , alors G a seulement six sommets. Clairement G est sans P_5 . De plus, résoudre le problème du stable de poids maximum est alors trivial. En outre, on peut remarquer que chaque sommet de G appartient à un triangle. Donc si l'un de ces sommets est double, G_0 contient un K_4 ; si G n'a pas de sommet double, alors chaque sommet a_i de G représente un stable Y_i (un membre de \mathcal{M}) dans G_0 , et G_0 est 3-colorable : il suffit de donner la couleur 1 à $Y_1 \cup Y_6$, la couleur 2 à $Y_2 \cup Y_5$ et la couleur 3 à $Y_3 \cup Y_4$. Remarquons pour clore cette section que si G contient un H_1 et ne contient ni P_5 ni membre de \mathcal{F} , il ne contient pas non plus de C_5 ou de \overline{C}_6 .

G contient un H_2

Soient a_1, \dots, a_6 six sommets de G qui induisent un H_2 , avec $E_{H_2} = \{a_1a_2, a_1a_3, a_2a_3, a_4a_5, a_4a_6, a_5a_6, a_2a_4, a_3a_4, a_3a_5\}$. Nous dirons qu'un graphe H a une H_2 -structure si son ensemble de sommets V_H peut être partitionné en six ensembles non vides A_1, \dots, A_6 tels que :

- Chaque ensemble A_1, \dots, A_6 est stable,
- A_3 est complet à $A_1 \cup A_2 \cup A_4 \cup A_5$ et anticomplet à A_6 ,
- A_4 est complet à $A_2 \cup A_3 \cup A_5 \cup A_6$ et anticomplet à A_1 ,
- $A_1 \cup A_2$ est anticomplet à $A_5 \cup A_6$,
- Chaque sommet de A_1 a un voisin dans A_2 et chaque sommet de A_2 a un voisin dans A_1 ,
- Chaque sommet de A_6 a un voisin dans A_5 et chaque sommet de A_5 a un voisin dans A_6 .

Notons que les six sommets a_1, \dots, a_6 induisent un graphe avec une H_2 -structure $(\{a_1\}, \dots, \{a_6\})$. Notre but est de « faire grossir » cette H_2 -structure au maximum.

Soit H un sous-graphe induit de G avec une H_2 -structure (A_1, \dots, A_6) , et x un sommet de $V_G \setminus V_H$. On dit que x peut être *ajouté* à cette H_2 -structure si le sous-graphe induit par $V_H \cup \{x\}$ a une H_2 -structure (A'_1, \dots, A'_6) où $A'_j = A_j \cup \{x\}$ pour un $j \in \{1, \dots, 6\}$ et $A'_i = A_i$ pour tout $i \neq j$.

Lemme 3.4 Soient H un sous-graphe induit de G possédant une H_2 -structure (A_1, \dots, A_6) , et x un sommet de $V_G \setminus V_H$ qui a un voisin dans $A_1 \cup A_2 \cup A_5 \cup A_6$. Alors soit x appartient à un sous-graphe de G qui induit un P_5 ou un membre de \mathcal{F} , ou x peut être ajouté à la H_2 -structure (A_1, \dots, A_6) .

Démonstration. À symétrie près, on peut supposer que x a un voisin dans $A_1 \cup A_2$.

Cas 1 : x a un voisin u_1 dans A_1 .

Soit u_2 un voisin de u_1 dans A_2 . Un tel sommet existe d'après la définition d'une H_2 -structure.

Supposons que x ait un voisin $u_3 \in A_3$. Si $x \sim u_2$, alors $\{x, u_1, u_2, u_3\}$ induit un K_4 ; donc $x \sim u_2$. Soient $u_4 \in A_4$ et $u_5 \in A_5$. Si x est adjacent à

ces deux sommets, alors $\{x, u_3, u_4, u_5\}$ induit un K_4 ; et s'il n'est adjacent à aucun des deux, alors $x-u_1-u_2-u_4-u_5$ induit un P_5 . Si x est adjacent à u_5 et pas à u_4 , alors $\{x, u_1, u_2, u_3, u_4, u_5\}$ induit un F_2 . Par conséquent, nous pouvons supposer que x est complet à A_4 et anticomplet à A_5 . De plus, si x a un voisin $u_6 \in A_6$, alors $u_2-u_1-x-u_6-u_5$ est un P_5 , quel que soit $u_5 \in A_5 \cap N(u_6)$; donc nous pouvons admettre que x est anticomplet à A_6 . Si x a un non-voisin $v_3 \in A_3$, alors $x-u_1-v_3-u_5-u_6$ est un P_5 , quels que soient $u_5 \in A_5$ et $u_6 \in A_6 \cap N(u_5)$. Si x a un voisin $v_2 \in A_2$, alors $\{x, v_2, u_3, u_4\}$ induit un K_4 pour tout $u_4 \in A_4$. Finalement, x est donc complet à $A_3 \cup A_4$, anticomplet à $A_2 \cup A_5 \cup A_6$, et un voisin dans A_1 ; il peut donc être placé dans A_2 et ainsi ajouté à la H_2 -structure.

Supposons maintenant que x n'ait aucun voisin dans A_3 . Soient $u_3 \in A_3$, $u_4 \in A_4$, $u_5 \in A_5$ et $u_6 \in A_6 \cap N(u_5)$. Supposons tout d'abord que x ne soit pas adjacent à u_2 . Si x n'est adjacent ni à u_5 ni à u_6 , alors $x-u_1-u_3-u_5-u_6$ est un P_5 . Si x est adjacent à l'un et pas à l'autre, alors $\{x, u_1, u_2, u_5, u_6\}$ induit un P_5 . Donc $x \sim u_5$ et $x \sim u_6$. Si x est adjacent à u_4 , alors $\{x, u_4, u_5, u_6\}$ induit un K_4 , et dans le cas contraire $\{x, u_1, u_2, u_3, u_4, u_5, u_6\}$ induit un F_4 ; par conséquent, x est adjacent à u_2 . Supposons à présent que x ait un voisin $u_6 \in A_6$, et prenons un sommet $u_5 \in A_5 \cap N(u_6)$. Si $x \approx u_4$, alors $\{x, u_1, \dots, u_6\}$ induit un F_4 ou un F_7 (selon que $x \sim u_5$ ou non). Si x est adjacent à u_4 et pas à u_5 , alors $\{x, u_2, \dots, u_6\}$ induit un F_2 . Si $x \sim u_4$ et $x \sim u_5$, alors $\{x, u_4, u_5, u_6\}$ induit un K_4 ; par conséquent x est anticomplet à A_6 . Si x a un non-voisin $w \in A_4 \cup A_5$, alors $x-u_1-u_2-w-u_6$ est un P_5 , quel que soit $u_6 \in A_6 \cap N(w)$. Supposons donc que w soit complet à $A_4 \cup A_5$. Si x a un voisin $v_3 \in A_3$, alors $\{x, v_3, u_4, u_5\}$ induit un K_4 pour tous $u_4 \in A_4$ et $u_5 \in A_5$; donc x est anticomplet à A_3 . Supposons que x ait un non-voisin $v_2 \in A_2$; si v_2 est adjacent à u_1 , on peut recommencer le raisonnement comme au début de ce paragraphe (avec u_2), donc nous supposons que $v_2 \approx u_1$. Par définition d'une H_2 -structure, v_2 a un voisin $v_1 \in A_1$, et par le même argument on peut supposer que x n'est pas adjacent à v_1 . Alors $u_1-x-u_4-v_2-v_1$ est un P_5 ; par conséquent, x est complet à A_2 . À présent, si x a un non-voisin $w_1 \in A_1$, alors $w_1-w_2-x-u_5-u_6$ est un P_5 quels que soient $w_2 \in A_2 \cap N(w_1)$, $u_5 \in A_5$, $u_6 \in A_6 \cap N(u_5)$. Finalement, x est complet à $A_1 \cup A_2 \cup A_4 \cup A_5$ et anticomplet à A_6 , donc x peut être ajouté à la H_2 -structure (dans A_3).

Cas 2 : x a un voisin u_2 dans A_2 .

Comme nous ne sommes plus dans le cas 1, et par symétrie, on peut supposer que x n'a pas de voisin dans $A_1 \cup A_6$. Si x a un voisin $u_5 \in A_5$, alors $u_1-u_2-x-u_5-u_6$ est un P_5 , quels que soient $u_1 \in A_1 \cap N(u_2)$, $u_5 \in A_5$ et $u_6 \in A_6 \cap N(u_5)$; donc x est anticomplet à A_5 . Si x a un non-voisin u_3 dans A_3 , alors $x-u_2-u_3-u_5-u_6$ est un P_5 ; donc x est complet à A_3 . À présent, si x a un voisin u_4 dans A_4 , alors $\{x, u_2, u_3, u_4\}$ induit un K_4 pour tout $u_3 \in A_3$; donc x est anticomplet à A_4 . Finalement, x est complet à A_3 , anticomplet à $A_1 \cup A_4 \cup A_5 \cup A_6$, et a un voisin dans A_2 , donc il peut être ajouté à la H_2 -structure (dans A_1). \square

D'un point de vue algorithmique, il est facile de tester les propriétés du lemme précédent. Initialement, on pose $A_i = \{a_i\}$ pour chaque $i \in \{1, \dots, 6\}$. Ensuite, on applique les remarques (a) et (b) de la section 3.2.1.

Le temps de calcul total est donc en $\mathcal{O}(m + n)$.

On peut désormais supposer que H a une H_2 -structure (A_1, \dots, A_6) telle qu'aucun sommet de $V_G \setminus (A_1, \dots, A_6)$ n'a de voisin dans $A_1 \cup A_2 \cup A_5 \cup A_6$. Posons $A = A_1 \cup \dots \cup A_6$ et $R = V_G \setminus A$.

Lemme 3.5 *Si G est sans P_5 , alors tout sommet de R a un voisin dans $A_3 \cup A_4$. De plus, R est stable.*

Démonstration. Supposons tout d'abord qu'un sommet $x \in R$ n'ait pas de voisin dans A . Comme G est connexe, il existe une plus courte chaîne $p_0 \dots p_k$ entre x et A , c'est-à-dire $x = p_0, p_0, \dots, p_{k-1} \in R$ et $p_k \in A$, avec $k \geq 2$. Par définition de R et à symétrie près, on peut supposer que $p_k \in A_3$. Prenons des sommets u_5 dans A_5 et u_6 dans $A_6 \cap N(u_5)$. Si $k \geq 4$, alors $x-p_1-p_2-p_3-p_4$ est un P_5 . Si $k = 3$, alors $x-p_1-p_2-p_3-u_5$ est un P_5 . Si $k = 2$, alors $x-p_1-p_2-u_5-u_6$ est un P_5 .

Supposons à présent que R ne soit pas stable, et soit Y l'ensemble des sommets d'une composante de $G[R]$ de taille au moins deux. Comme G est un graphe quotient, Y n'est pas un ensemble homogène, et il existe donc des sommets adjacents y et z de Y et un sommet $x \in V_G \setminus Y$ adjacent à y et pas à z . Par définition de R et de Y , et à symétrie près, soit $x \in A_3$. Prenons des sommets u_5 dans A_5 et u_6 dans $A_6 \cap N(u_5)$. Alors $z-y-x-u_5-u_6$ est un P_5 . \square

Soit R' l'ensemble des sommets de R qui ont un voisin dans A_3 et un voisin dans A_4 . Pour $j \in \{3, 4\}$, soit R_j l'ensemble des sommets de $R \setminus R'$ qui ont un voisin dans A_j . Ainsi, $R = R_3 \cup R_4 \cup R'$.

Lemme 3.6 *G_0 est 3-colorable si et seulement si $A \cup R'$ ne contient pas de sommet double.*

Démonstration. Supposons que $A \cup R'$ contienne un sommet double x . Par définition de A_1, \dots, A_6 et R' , il est facile de voir que x a deux voisins adjacents u et v dans G . Alors $\{x, \bar{x}, u, v\}$ induit un K_4 dans G_0 , donc G_0 n'est pas 3-colorable. Réciproquement, supposons que $A \cup R'$ ne contienne aucun sommet double. Donnons les couleurs 1 aux sommets de $A_1 \cup A_4$, 2 aux sommets de $A_2 \cup A_5$, 3 aux sommets de $A_3 \cup A_6$; si u est un sommet double de R_3 , on lui attribue les couleurs 1 et 2; si u est un sommet double de R_4 , on lui attribue les couleurs 2 et 3; enfin, on donne la couleur aux sommets non encore colorés de R . Il est facile de voir qu'il s'agit d'une 3-coloration de G , à partir de laquelle on déduit une 3-coloration de G_0 . \square

Lemme 3.7 *Tout stable de G est un sous-ensemble de l'un des trois sous-ensembles suivants : $V_0 = A_1 \cup A_2 \cup A_5 \cup A_6 \cup R$, $V_3 = A_3 \cup A_6 \cup R$, $V_4 = A_4 \cup A_1 \cup R$. De plus, chacun de ces trois ensembles induit un sous-graphe biparti de G .*

Démonstration. Soit S un stable de G . Supposons que S contienne un sommet de A_3 . Alors la définition d'une H_2 -structure implique que S ne

contient aucun sommet de $A_1 \cup A_2 \cup A_4 \cup A_5$. Ainsi, $S \subseteq V_3$. De même, si S contient un sommet A_4 alors $S \subseteq V_4$. D'un autre côté, si S ne contient aucun sommet de $A_3 \cup A_4$, alors $S \subseteq V_0$. De plus, V_3 induit un sous-graphe biparti de G parce que $A_3 \cup A_6$ et R sont deux stables; de manière similaire, V_4 induit un sous-graphe biparti de G ; et V_0 induit aussi un sous-graphe biparti de G parce que $A_1 \cup A_6 \cup R$ et $A_2 \cup A_5$ sont deux stables. \square

Il découle du lemme précédent que, pour résoudre le problème du stable de poids maximum dans G , il suffit de le résoudre pour les trois sous-graphes bipartis $G[V_0]$, $G[V_3]$, $G[V_4]$ et de retourner la meilleure solution parmi les trois. D'après le théorème 3.1, ceci peut être réalisé en temps linéaire.

Même si l'algorithme a atteint ce point sans trouver de P_5 et a résolu avec succès le problème du stable de poids maximum en temps linéaire, il n'est pas exclu que le graphe contienne un P_5 . Nous pouvons clarifier cette question avec le lemme suivant.

Lemme 3.8 G est sans P_5 si et seulement si chacun des quatre sous-graphes $G[A_1 \cup A_2]$, $G[A_5 \cup A_6]$, $G[A_3 \cup R]$, $G[A_4 \cup R]$ est sans $2K_2$.

Démonstration. Si $G[A_1 \cup A_2]$ ou $G[A_3 \cup R]$ contient un $2K_2$, alors en ajoutant un sommet de A_4 à ce $2K_2$ on obtient un P_5 . Il en va de même pour $G[A_5 \cup A_6]$ et $G[A_4 \cup R]$ (en ajoutant un sommet de A_5).

Réciproquement, soit $x_1-x_2-x_3-x_4-x_5$ un P_5 de G . Si aucun des x_1, \dots, x_5 n'appartient à $A_3 \cup A_4$, alors le P_5 se trouve dans l'un des sous-graphes $G[A_1 \cup A_2]$ ou $G[A_5 \cup A_6]$ donc ce sous-graphe n'est pas sans $2K_2$. Supposons à présent, à symétrie près, que $x_1 \in A_3$. Alors ses non-voisins x_4 et x_5 sont dans $A_3 \cup A_6 \cup R$, et puisqu'ils sont adjacents, on a soit $x_4 \in A_3$ et $x_5 \in R$, ou $x_4 \in R$ et $x_5 \in A_3$. Dans chaque cas, comme x_2 est adjacent à x_1 et pas à x_4 ni x_5 , il doit être dans R et par conséquent $G[A_3 \cup R]$ contient un $2K_2$ de sommets x_1, x_2, x_4, x_5 . Le même argument est vérifié si $x_2 \in A_3$. Enfin, supposons que $x_3 \in A_3$; alors c'est une question de routine de vérifier que soit $x_2, x_4 \in R$ et $x_1, x_5 \in A_3$, soit $x_2, x_4 \in A_5$ et $x_1, x_5 \in A_6$, et donc $G[A_3 \cup R]$ ou $G[A_5 \cup A_6]$ contient un $2K_2$ de sommets x_1, x_2, x_4, x_5 . \square

Il ressort du lemme précédent que, pour vérifier que G est sans P_5 , il suffit de vérifier que chacun des quatre sous-graphes bipartis évoqués ci-dessus est sans $2K_2$, ce que l'on peut faire en temps linéaire (voir la section 2.2.3). Remarquons pour clore cette section que si G contient un H_2 et ne contient ni P_5 ni membre de \mathcal{F} , il ne contient pas non plus de C_5 ou de \bar{C}_6 (ce fait est facilement vérifiable, et nous laissons la preuve en exercice).

3.2.5 Cas où G contient un C_5

Dans cette section, on suppose que G contient un C_5 , renvoyé par l'algorithme de la section 3.2.3.

Remarquons que si, dans cette section, l'algorithme trouve un $2K_2$ induit, alors nous pouvons retourner à la section 3.2.4 et appliquer le raisonnement qui y est décrit (en réalité, nous avons précédemment observé que

si G contient un $2K_2$, alors soit ce n'est pas un graphe de \mathcal{C} soit il ne contient pas de C_5 . Autrement dit, on pourrait simplement arrêter l'algorithme avec en sortie le message « G n'est pas dans \mathcal{C} »; l'étape précédente n'est nécessaire que si l'on désire explicitement un sous-graphe interdit). Soient a_1, \dots, a_5 cinq sommets qui induisent un C_5 d'arêtes $a_i a_{i+1}$ (les indices étant pris modulo 5). Nous allons « faire grossir » ce C_5 en une structure plus grande, comme suit. Pour chaque $i \in \{1, \dots, 5\}$, soient $R_i = \{a_i\}$ et $R = R_1 \cup \dots \cup R_5$. Comme expliqué dans la remarque (a) de la section 3.2.1, chaque sommet $x \in V_G \setminus R$ a un type $(c_1(x), \dots, c_5(x))$, où $c_i(x)$ est égal à 0 si x est anticomplet à R_i , et 1 sinon. Appliquons la procédure suivante :

Tant qu'il existe un sommet $x \in V_G \setminus R$ anticomplet à $R_{i-2} \cup R_{i+2}$ pour un $i \in \{1, \dots, 5\}$ et pas anticomplet à R , tester si x est complet à $R_{i-1} \cup R_{i+1}$.
 Si x n'est pas complet à $R_{i-1} \cup R_{i+1}$, alors renvoyer le message « G contient un P_5 » et arrêter.
 Sinon, ajouter x à R_i , mettre à jour le type de chaque voisin de x et continuer.

Trouver de tels sommets et les tester peut être fait en temps $\mathcal{O}(m+n)$ d'après les remarques (a) et (b) de la section 3.2.1. Lorsque la procédure renvoie le message « G contient un P_5 », ce message est correct. En effet, considérons le sommet x correspondant. x a un voisin y dans R , et $y \in R_{i-1} \cup R_i \cup R_{i+1}$ puisque x est anticomplet à $R_{i-2} \cup R_{i+2}$. x a également un non-voisin $z \in R_{i-1} \cup R_{i+1}$; à symétrie près, supposons que $z \in R_{i-1}$. Prenons un sommet r_j dans R_j pour tout $j \neq i-1$. Si $y \in R_{i+1}$, alors $x-y-r_{i+2}-r_{i-2}-z$ est un P_5 . Si $y \in R_i$, alors $x-y-z-r_{i-2}-r_{i+2}$ est un P_5 . Si $y \in R_{i-1}$, alors soit $x-y-r_{i-2}-r_{i+2}-r_{i+1}$, soit $x-r_{i+1}-r_{i+2}-r_{i-2}-z$ est un P_5 .

Supposons désormais que l'algorithme n'ait pas trouvé de P_5 , et considérons les cinq ensembles R_1, \dots, R_5 à la fin de la procédure précédente. Par définition, il est facile de voir que chaque R_i est complet à $R_{i-1} \cup R_{i+1}$ et anticomplet à $R_{i-2} \cup R_{i+2}$. D'après la remarque (a) de la section 3.2.1, nous pouvons supposer que chaque sommet x de $V_G \setminus R$ se trouve dans l'ensemble Q_t qui correspond au type t de x par rapport à la famille R_1, \dots, R_5 . Notons W l'ensemble des sommets de $V_G \setminus R$ qui ont un voisin dans R . Pour chaque $i = 1, \dots, 5$, définissons les ensembles suivants :

- Soit W_i l'ensemble de tous les $x \in W$ anticomplets à R_i qui ont un voisin dans chaque R_j ($j \neq i$);
- Soit T_i l'ensemble de tous les $x \in W$ anticomplets à $R_{i-1} \cup R_{i+1}$ qui ont un voisin dans chaque R_j ($j \neq i \pm 1$);
- $Z = V_G \setminus (R \cup W)$.

Autrement dit, Z est l'ensemble des sommets de type $(0, 0, 0, 0, 0)$, W_1 est l'ensemble des sommets de type $(0, 1, 1, 1, 1)$, T_1 est l'ensemble des sommets de type $(1, 0, 1, 1, 0)$, et W_j et T_j sont définis de manière similaire pour $j \in \{2, \dots, 5\}$. Nous dirons que $(R_1, \dots, R_5, W_1, \dots, W_5, T_1, \dots, T_5, Z)$ est une C_5 -structure de G .

Lemme 3.9 $W = W_1 \cup \dots \cup W_5 \cup T_1 \cup \dots \cup T_5$, ou G contient un P_5 ou un membre de \mathcal{F} .

Démonstration. Soit un sommet $w \in W$. Si w est anticomplet à au moins trois des ensembles R_1, \dots, R_5 , alors il est anticomplet à $R_j \cup R_{j+1}$ pour un j . Dans ce cas, w a été examiné durant la procédure ci-dessus, et alors soit la procédure a trouvé un P_5 et s'est arrêtée, soit elle a ajouté w à l'un des R_i ; dans les deux cas il s'agit d'une contradiction. Donc w est anticomplet à au plus deux des R_1, \dots, R_5 ; et par le même argument, s'il est anticomplet à deux ensembles R_h et R_j , alors $j \neq h \pm 1$. Donc si w est anticomplet à deux des R_1, \dots, R_5 , alors w appartient à T_i pour un certain i ; et si w est anticomplet à un seul des R_1, \dots, R_5 , alors w appartient à W_i pour un certain i . Enfin, si w n'est anticomplet à aucun des R_1, \dots, R_5 , il a un voisin $u_i \in R_i$ pour tout $i \in \{1, \dots, 5\}$ et alors $\{w, u_1, \dots, u_5\}$ induit un F_2 . \square

Lemme 3.10 *On peut supposer que chacun des ensembles R_1, R_2, R_3, R_4, R_5 et Z est stable.*

Démonstration. Supposons, à symétrie près, que R_1 ou Z contienne deux sommets adjacents u et v . Alors $\{u, v, r_3, r_4\}$ induit un $2K_2$ pour tout $r_3 \in R_3$ et $r_4 \in R_4$. Dans ce cas, comme mentionné précédemment au début de cette section, l'algorithme retourne le message (correct) « G n'est pas dans la classe \mathcal{C} ». \square

Pour tous ensembles K et L disjoints, on note :

- K^L l'ensemble $\{x \in K \mid x \text{ a un voisin dans } L\}$,
- K^{-L} l'ensemble $\{x \in K \mid x \text{ a un non-voisin dans } L\}$,
- K^{0L} l'ensemble $\{x \in K \mid x \text{ n'a aucun voisin dans } L\}$.

(Autrement dit, $K^{0L} = K \setminus K^L$). Nous nous intéresserons à de tels ensembles pour certaines paires K, L parmi les ensembles $(R_1, \dots, R_5, W_1, \dots, W_5, T_1, \dots, T_5, Z)$; leur nombre sera donc constant. De plus, d'après la remarque (a) de la section 3.2.1, tous ces ensembles peuvent être calculés en temps linéaire. Analysons à présent les relations d'adjacence entre les ensembles de la C_5 -partition.

Lemme 3.11 *Les propriétés suivantes sont vérifiées pour tout $i \in \{1, \dots, 5\}$, ou G contient un P_5 ou un membre de \mathcal{F} :*

1. T_i est complet à R_i et à $T_{i-1} \cup T_{i+1}$;
2. T_i est stable;
3. tout sommet de T_i est complet à R_{i-2} ou à R_{i+2} ;
4. T_i^Z est complet à $R_{i-2} \cup R_{i+2}$;
5. W_i est complet à $R_{i-1} \cup R_{i+1} \cup T_i \cup T_{i-2} \cup T_{i+2} \cup W_{i-1} \cup W_{i+1}$ et anticomplet à $T_{i-1} \cup T_{i+1}$;
6. W_i est stable;
7. Si $W_i \neq \emptyset$, alors $W_{i-2} \cup W_{i+2}$ est vide, T_i^Z est vide, et il n'y a pas d'arête entre T_{i-1} et T_{i+1} ;
8. un sommet de R_{i+3} est complet à $W_i \cup T_{i+1}$ et un sommet de R_{i-3} est complet à $W_i \cup T_{i-1}$;

9. $T_{i-1}^{R_{i+1}}$ est complet à $T_{i+1}^{-R_{i-1}}$;
10. si $T_i^{0T_{i+2}} \neq \emptyset$, alors tout sommet de R_i est complet à T_{i+2} ou à $T_{i+3}^{0T_i}$;
si $T_i^{0T_{i-2}} \neq \emptyset$, alors tout sommet de R_i est complet à T_{i-2} ou à $T_{i-3}^{0T_i}$;
11. tout sommet de Z est soit complet soit anticomplet à $T_{i-1}^{-T_{i+1}} \cup T_{i+1}^{-T_{i-1}}$.

Démonstration. Pour chaque point du lemme, la preuve est un algorithme qui soit trouve un P_5 ou un membre de \mathcal{F} , soit établit la propriété. Cet algorithme est *robuste* dans le sens où si le graphe donné en entrée *n'est pas sans P_5* , soit il exhibe un certificat de la non-appartenance du graphe à notre classe (ce certificat est donc un P_5), soit il fournit quand même, s'il y parvient, une 3-coloration et un stable de poids maximum. Pour l'ensemble de cette démonstration, r_j désigne un sommet arbitraire de R_j pour tout $j \in \{1, \dots, 5\}$.

1. À symétrie près, considérons un sommet $t_1 \in T_1$. Par définition de T_1 , nous savons que t_1 n'a pas de voisin dans $R_1 \cup R_5$ et a des voisins $u_3 \in R_3$ et $u_4 \in R_4$. Supposons que t_1 ait un non-voisin $u_1 \in R_1$. Alors $t_1-u_4-r_5-u_1-r_2$ est un P_5 . Ainsi, T_j est complet à R_j pour chaque $j \in \{1, \dots, 5\}$. Supposons maintenant, à symétrie près, que t_1 ait un non-voisin $t_2 \in T_2$; t_2 a un voisin $v_5 \in R_5$, et $u_3-t_1-r_1-v_5-t_2$ est un P_5 .
2. À symétrie près, supposons que T_1 contienne deux sommets adjacents u et v . D'après la propriété 1, u et v sont adjacents à r_1 , et par définition de T_1 , u a des voisins $u_3 \in R_3$ et $u_4 \in R_4$ et v a des voisins $v_3 \in R_3$ et $v_4 \in R_4$. Si $u_3 = v_3$ et $u_4 = v_4$, alors $\{u, v, u_3, u_4\}$ induit un K_4 . Supposons donc, à symétrie près, que $u_3 \neq v_3$ et, par le même argument, que $u \sim v_3$ et $v \sim u_3$. Si $u_4 = v_4$, alors $\{u, v, r_1, u_3, v_3, u_4\}$ induit un F_6 . Si $u_4 \neq v_4$, par le même argument $u \sim v_4$ et $v \sim u_4$, et alors $u-r_1-r_2-v_3-v_4$ est un P_5 .
3. Supposons qu'un sommet $t_1 \in T_1$ ait des non-voisins $u_3 \in R_3$ et $u_4 \in R_4$. On sait par ailleurs que t_1 a un voisin $u_1 \in R_1$ et un non-voisin $u_2 \in R_2$. Alors $t_1-u_1-u_2-u_3-u_4$.
4. Soit $t_1 \in T_1$ un sommet ayant un voisin dans $z \in Z$. Supposons que t_1 ait un non-voisin $x \in R_3 \cup R_4$, disons $x \in R_4$. On sait que t_1 a un voisin $u_4 \in R_4$ et x a un voisin $u_5 \in R_5$. Alors $z-t_1-u_4-u_5-x$ est un P_5 .
5. À symétrie près, considérons un sommet $w_1 \in W_1$. On sait que w_1 n'a aucun voisin dans R_1 et a des voisins $v_3 \in R_3$ et $v_4 \in R_4$. Supposons tout d'abord que w_1 ait un non-voisin x dans $R_5 \cup R_2 \cup T_1 \cup T_4 \cup T_3 \cup W_5 \cup W_2$.
Supposons que $x \in R_2$. On sait que w_1 a un voisin $u \in R_2$, donc $x-r_1-u-w_1-u_4$ est un P_5 . Le même argument est valide si $x \in R_5$. Par conséquent, w_1 est complet à $R_2 \cup R_5$ et W_j est complet à $R_{j-1} \cup R_{j+1}$ pour tout $j \in \{1, \dots, 5\}$.
Supposons maintenant que $x \in T_1$. Si x n'est pas adjacent à v_3 , alors $x-r_1-r_5-w_1-v_3$ est un P_5 . Donc $x \sim v_3$, et par symétrie $x \sim v_4$. Mais alors $\{w_1, x, r_1, r_2, v_3, v_4, r_5\}$ induit un F_6 .
Supposons que $x \in T_3 \cup W_2$. Ainsi, par définition, x a un voisin $u_1 \in R_1$ et aucun voisin dans R_2 . Si $x \sim v_4$, alors $x-u_1-r_2-w_1-v_4$ est un

P_5 . Si $x \sim v_4$, alors $x \in W_2$ et x est donc complet à $R_1 \cup R_3$, et a un voisin $v_5 \in R_5$. Mais alors $\{w_1, x, r_1, r_2, v_3, v_4, u_5\}$ induit un F_8 . On a le même résultat si $x \in T_4 \cup W_5$.

Supposons enfin que w_1 ait un voisin $t_2 \in T_2$. Par définition, t_2 a un voisin $u_5 \in R_5$. Si t_2 est adjacent à v_4 , alors $\{w_1, t_2, v_4, u_5\}$ induit un K_4 . Si t_2 n'est pas adjacent à v_4 , alors $\{w_1, t_2, r_2, v_3, v_4, u_5\}$ induit un F_2 . On aboutit évidemment à la même conclusion si w_1 a un voisin dans T_5 .

6. Supposons, à symétrie près, que W_1 contienne deux sommets adjacents u et v . D'après la propriété 5, u et v sont adjacents à r_2 et à r_5 et pas à r_1 . Par définition de W_1 , u a des voisins $u_3 \in R_3$ et $u_4 \in R_4$. Si v est adjacent à u_3 , alors $\{u, v, r_2, u_3\}$ induit un K_4 . Donc $v \approx u_3$ et par symétrie $v \approx u_4$. Alors $\{u, v, r_2, u_3, u_4, r_5\}$ induit un F_2 .

7. À symétrie près, soit $w_1 \in W_1$. Nous savons que w_1 a des voisins $v_3 \in R_3$ et $v_4 \in R_4$.

Supposons tout d'abord qu'il existe un sommet w_3 dans W_3 . Par définition, w_3 a des voisins $v_5 \in R_5$ et $v_1 \in R_1$. Si w_1 est adjacent à w_3 , alors d'après la propriété 5 $\{w_1, w_3, v_4, v_5\}$ induit un K_4 . Si w_1 n'est pas adjacent à w_3 , alors toujours d'après la propriété 5, $\{w_1, w_3, v_1, r_2, v_3, v_4, v_5\}$ induit un F_7 . Donc $W_3 = \emptyset$, et de manière similaire $W_4 = \emptyset$.

Supposons à présent qu'il existe une arête t_1z avec $t_1 \in T_1$ et $z \in Z$. En vertu de la propriété 3, t_1 est complet à $R_3 \cup R_4$. Selon la propriété 5, t_1 est adjacent à w_1 . Mais alors $\{t_1, w_1, v_3, v_4\}$ induit un K_4 .

Supposons enfin qu'il existe une arête t_2t_5 entre $t_2 \in T_2$ et $t_5 \in T_5$. Le sommet t_2 a un voisin $u_5 \in R_5$ et le sommet t_5 a un voisin $u_2 \in R_2$. Alors, d'après les propriétés 1 et 5, $\{w_1, t_2, t_5, u_2, v_3, v_4, u_5\}$ induit un F_5 , F_7 ou F_9 (selon l'existence des arêtes t_2v_4 et t_5v_3).

8. À symétrie près, prenons $i = 1$. Soit u un sommet de R_4 ayant le plus grand nombre de voisins dans $W_1 \cup T_2$, et supposons qu'il existe un sommet $v \in W_1 \cup T_2$ non adjacent à u . Nous savons que v a un voisin $u_4 \in R_4$. Le choix de u implique qu'il existe un sommet $w \in W_1 \cup T_2$ adjacent à u et pas à u_4 . Mais alors $u-w-r_2-v-u_4$ est un P_5 .

9. À symétrie près, prenons $i = 2$. Supposons qu'il existe un sommet $t_1 \in T_1^{-R_3}$ non adjacent à un sommet $t_3 \in T_3^{-R_1}$. Par définition de $T_1^{-R_3}$, le sommet t_1 a un non-voisin $x_3 \in R_3$, et de manière similaire, t_3 a un non-voisin $x_1 \in R_1$. Alors $t_1-x_1-r_2-x_3-t_3$ est un P_5 .

10. Prenons $i = 1$, et supposons que $T_1^{0T_3}$ contienne un sommet t_1 , et qu'un sommet $x_1 \in R_1$ a à la fois un non-voisin dans $t_3 \in T_3$ et un non-voisin $t_4 \in T_4^{0T_1}$. Par définition de ces ensembles, nous savons que $t_1 \approx t_3$ et $t_1 \approx t_4$. De plus, t_3 est adjacent à t_4 d'après la propriété 1, et à r_5 d'après la propriété 3. Mais alors $t_1-x_1-r_5-t_3-t_4$ est un P_5 .

11. Prenons $i = 1$, et supposons qu'il existe un sommet $z \in Z$ ayant à la fois un voisin u et un non-voisin v dans $T_5^{-T_2} \cup T_2^{-T_5}$. On peut supposer que $u \in T_5^{-T_2}$; donc u a un non-voisin $u' \in T_2^{-T_5}$. Si $z \approx u'$, alors $z-u-a_3-b_4-u'$ est un P_5 (où a_3 est un voisin de u dans R_3 et b_4 un voisin de u' dans R_4), donc $z \sim u'$. u et u' jouent donc des rôles symétriques, et on peut supposer que $v \in T_5^{-T_2}$. Si $v \approx u'$, alors $z-u'-$

b_4-a_3-v est un P_5 ; donc $v \sim u'$. Comme $v \in T_5^{-T_2}$, il a un non-voisin $v' \in T_2$. Si $z \sim v'$, alors $z-v'-b_4-a_3-v$ est un P_5 ; donc $z \approx v'$. Enfin, si $u \sim v'$, alors $v'-u-a_3-v-u'$ est un P_5 , donc $u \approx v'$. Mais finalement, $z-u-a_3-b_4-v'$ est un P_5 .

□

Tester la 3-colorabilité

Nous pouvons à présent tester si G est 3-colorable. Pour toute fonction $\gamma : V_G \rightarrow \{1, 2, \dots\}$ et tout sous-ensemble $X \subseteq V_G$, on note $\gamma(X)$ l'ensemble $\{\gamma(x) | x \in X\}$ (pour alléger l'écriture, nous omettrons les accolades lorsque cet ensemble sera réduit à un singleton).

Lemme 3.12 *Si au moins deux des R_1, \dots, R_5 contiennent un sommet double, ou si W contient un sommet double, alors G n'est pas 3-colorable.*

Démonstration. Supposons tout d'abord, à symétrie près, qu'il existe un sommet double u dans R_1 et un sommet double v dans R_j avec $j \in \{2, 3\}$. Rappelons qu'il existe un sommet \bar{u} dans G_0 adjacent à u et qui a les mêmes voisins que u dans $V_G \setminus \{u\}$, et il en va de même pour v . Si $j = 2$, alors $\{u, \bar{u}, v, \bar{v}\}$ induit un K_4 dans le graphe original G_0 . Si $j = 3$, prenons un sommet r_i dans R_i pour chaque $i \in \{2, 4, 5\}$. Alors $\{u, \bar{u}, v, \bar{v}, r_2, r_4, r_5\}$ induit un F_3 dans G_0 .

Supposons à présent que W contienne un sommet double u . Par définition de W , il y a un entier $i \in \{1, \dots, 5\}$ tel que u a des voisins $u_i \in R_i$ et $u_{i+1} \in R_{i+1}$. Alors $\{u, \bar{u}, u_i, u_{i+1}\}$ induit un K_4 dans le graphe original G_0 . □

Les hypothèses du lemme précédent sont facilement vérifiables en temps $\mathcal{O}(n)$. Si l'une est satisfaite, l'algorithme s'arrête avec la réponse « G n'est pas 3-colorable ». Supposons désormais que W et au moins quatre des ensembles R_1, \dots, R_5 ne contiennent pas de sommet double.

Lemme 3.13 *Si G admet une 3-coloration γ , alors il existe un entier h dans $\{1, \dots, 5\}$ tel que (à la renumérotation des couleurs près), $\gamma(R_{h+1}) = \gamma(R_{h+4}) = 1$, $\gamma(R_{h+2}) = 2$ et $\gamma(R_{h+3}) = 3$.*

Démonstration. Si les trois couleurs apparaissent dans un des R_i , alors aucune couleur ne peut apparaître dans R_{i+1} , une contradiction. Par conséquent, il y a au plus deux couleurs dans chaque R_i . Supposons, à symétrie près, qu'on trouve les couleurs 1 et 2 dans R_1 . Alors on doit avoir $\gamma(R_2 \cup R_5) = 3$, donc $\gamma(R_3 \cup R_4) \subseteq \{1, 2\}$, donc $\gamma(R_3) = 1$ et $\gamma(R_4) = 2$ ou vice versa; donc le lemme est vérifié avec $h = 2$. À présent supposons que seulement une couleur apparaisse dans chaque R_i . Alors on peut supposer à symétrie près que $\gamma(R_1) = 1$, $\gamma(R_2) = 2$, $\gamma(R_3) = 3$, $\gamma(R_4) = 1$, $\gamma(R_5) = 2$, et le lemme est vérifié pour $h = 5$. □

Pour chaque sommet $h \in \{1, \dots, 5\}$, soit $\gamma(h)$ la *précoloration* (coloration partielle) où la couleur 1 est attribuée aux sommets de $R_{h+1} \cup R_{h+4}$,

la couleur 2 aux sommets de R_{h+2} , la couleur 3 aux sommets de R_{h+3} , les autres sommets n'étant pas encore colorés. Le lemme précédent montre que, pour savoir si G est 3-colorable, il suffit de tester, pour chaque $h \in \{1, \dots, 5\}$, si $\gamma(h)$ s'étend à une 3-coloration de G . Ceci peut se faire facilement, en temps linéaire, comme suit.

Initialement, chaque sommet x a une liste de couleurs disponibles $L(x) = \{1, 2, 3\}$. Chaque sommet de $R \setminus R_h$ reçoit la couleur désignée par γ_h ; en particulier, si $R \setminus R_h$ contient un sommet double, alors l'algorithme déclare que γ_h ne peut s'étendre à une 3-coloration (l'algorithme enregistre les sommets non colorables, et si $\gamma < 5$ l'algorithme examine γ_{h+1} , sinon l'algorithme s'arrête avec le message « G n'est pas 3-colorable »). Quand un sommet est coloré, sa couleur est retirée de la liste de ses voisins. Si la liste d'un sommet est réduite à une seule couleur, ou si la liste d'un sommet double contient deux couleurs, alors on colore ce sommet avec la ou les couleurs disponibles, puis on met à jour la liste de ses voisins. Si la liste d'un sommet devient vide, ou si la liste d'un sommet double ne contient plus qu'une seule couleur, alors ce sommet n'est pas colorable et l'algorithme déclare que γ_h ne s'étend pas à une 3-coloration (et ici encore, l'algorithme enregistre les sommets non colorables, et si $\gamma < 5$ l'algorithme examine γ_{h+1} , sinon l'algorithme s'arrête avec le message « G n'est pas 3-colorable »). Une observation cruciale est que *chaque sommet de W a deux voisins de couleurs différentes dans $R \setminus R_h$* . Ceci découle directement de la définition de W . Cette observation signifie que, juste après les sommets que les sommets de $R \setminus R_h$ ont été colorés, chaque sommet de W a une liste de taille au plus 1, et assure que soit l'algorithme s'arrête parce qu'un sommet de W n'est pas colorable, soit tous les sommets de W peuvent être colorés. De plus, on peut supposer que tous les sommets de W sont colorés avant que les sommets de $R_h \cup Z$ ne soient considérés. Enfin, les sommets non colorés restants sont dans $R_h \cup Z$, qui est un stable, donc on peut les examiner un par un, indépendamment des autres. Si un tel sommet x n'est pas colorable, alors l'algorithme le mémorise (et si $h < 5$, il examine γ_{h+1}), sinon il affecte à x une couleur de $L(x)$ (ou deux couleurs si x est un sommet double).

Tout au long de cette procédure, colorer un sommet u et mettre à jour ses voisins demande un temps en $\mathcal{O}(d(u))$, donc le temps total est $\mathcal{O}(m + n)$.

À la fin de la procédure, deux issues sont possibles :

- (1) l'algorithme a essayé chacune des précolorations γ_h et aucune d'entre elles ne s'étend à une 3-coloration. L'algorithme retourne le message (correct) « G n'est pas 3-colorable » et s'arrête. Dans la section suivante, nous montrons que dans ce cas, G contient un membre de \mathcal{F} .
- (2) l'une des précoloration γ_h s'étend à une 3-coloration. Dans la section 3.2.5, nous analysons cette situation afin de trouver un stable de poids maximum.

Si G n'est pas 3-colorable.

Étant donnés deux ensembles $X, Y \subseteq V_G$, on appelle XY -arête une arête xy avec $x \in X$ et $y \in Y$.

Lemme 3.14 *Supposons que γ_h ne s'étende pas à une 3-coloration de G . Alors soit G_0 contient un membre de \mathcal{F} , soit l'une des propriétés suivantes est vérifiée :*

- i) $R \setminus R_h$ contient un sommet double.
- ii) $W_{h-1} \cup W_h \cup W_{h+1} \neq \emptyset$.
- iii) Il existe une $T_{h+1}T_{h+3}$ -arête ou une $T_{h-1}T_{h-3}$ -arête.
- iv) R_h contient un sommet qui a un voisin dans $T_{h+2} \cup W_{h+3}$ et un voisin dans $T_{h-2} \cup W_{h-3}$.
- v) Un sommet de Z a un voisin dans chacun des ensembles T_h, T_{h-2}, T_{h+2} .
- vi) Il existe un sommet double de Z qui a soit un voisin dans T_h et un voisin dans $T_{h-2} \cup T_{h+2}$, soit un voisin dans T_{h-1} et un voisin dans T_{h+1} .

Démonstration. Pour simplifier les notations, posons $h = 5$. Donc les sommets de $R_1 \cup R_4$ sont de couleur 1, les sommets de R_2 sont de couleur 2 et les sommets de R_3 sont de couleur 3. Supposons qu'aucune des propriétés (i)-(iii) ne soit vérifiée. Après que l'algorithme a coloré les sommets de $R \setminus R_5$, la définition des T_i et des W_i implique que chaque sommet $x \in T_1$ satisfait $L(x) = \{2\}$, car x a des voisins dans R_1, R_3 et R_4 ; nous résumons ce fait en le notant $L(T_1) = \{2\}$. De la même façon, on a $L(T_2) = \{3\}$, $L(T_3) = \{2\}$, $L(T_4) = \{3\}$, $L(T_5) = \{1\}$, $L(W_2) = \{2\}$ et $L(W_3) = \{3\}$. Notons que, puisque (i) n'est pas vérifiée, et d'après le lemme 3.12, il n'y a pas de sommet double dans l'ensemble $(R \setminus R_5) \cup W$. Donc la raison pour laquelle γ_5 ne s'étend pas à une coloration de G est qu'il existe un sommet x dans $R_5 \cup Z$ qui n'est pas colorable. Ceci signifie que, après que les sommets de W ont été colorés, on a l'un des deux cas suivants. Pour chaque $i \in \{1, \dots, 5\}$, on choisit un sommet $r_i \in R_i$.

Cas 1 : $L(x) = \emptyset$.

Ainsi, x a des voisins u, v, w de couleurs respectives 1, 2 et 3. Comme x est dans $R_5 \cup Z$, il n'a pas de voisin dans $R_2 \cup R_3$, et par conséquent $v \in T_1 \cup T_3 \cup W_2$ et $w \in T_4 \cup T_2 \cup W_3$. Si $x \in R_5$, alors par définition des T_i et des W_i , on a $v \in T_3 \cup W_2$ et $w \in T_2 \cup W_3$, et la propriété (iv) du lemme est vérifiée. Si $x \in Z$, alors $u \in T_5$ et d'après le lemme 3.11 3., u est complet à $R_2 \cup R_3$. En fonction de la position de v et w , neuf sous-cas sont possibles :

Cas 1.1 : $v \in T_1$ et $w \in T_4$. D'après le lemme 3.11 1., u est adjacent to v et w . Si v est adjacent à w , alors $\{u, v, w, x\}$ induit un K_4 . Si v n'est pas adjacent à w , alors $\{u, v, w, x, r_2, r_3\}$ induit un F_2 .

Cas 1.2 : $v \in T_1$ et $w \in T_2$. Ce cas est similaire au précédent (u, v, w sont dans trois T_i consécutifs).

Cas 1.3 : $v \in T_1$ et $w \in W_3$. D'après le lemme 3.11 1., u est adjacent à v , et d'après 5., w est adjacent à u et v . Alors $\{u, v, w, x\}$ induit un K_4 .

Cas 1.4 : $v \in T_3$ et $w \in T_4$. Ce cas est similaire au cas 1.1.

Cas 1.5 : $v \in T_3$ et $w \in T_2$. Dans ce cas, la propriété (v) du lemme est vérifiée.

Cas 1.6 : $v \in T_3$ et $w \in W_3$. Dans ce cas, la propriété 3. du lemme 3.11 est contredite.

Cas 1.7 : $v \in W_2$ et $w \in T_4$. Ce cas est similaire au cas 1.3.

Cas 1.8 : $v \in W_2$ et $w \in T_2$. Dans ce cas, la propriété 3. du lemme 3.11 est contredite.

Cas 1.9 : $v \in W_2$ et $w \in W_3$. D'après le lemme 3.11 5., u, v et w sont deux à deux adjacents. Donc $\{u, v, w, x\}$ induit un K_4 .

Cas 2 : x est un sommet double avec $|L(x)| \leq 1$.

Si $x \in R_5$, alors x doit avoir un voisin u de couleur 2 ou 3, et on peut supposer, à symétrie près, que $u \in W_2 \cup T_3$; donc u a un voisin $u_1 \in R_1$, et alors $\{x, \bar{x}, u, u_1\}$ induit un K_4 dans G_0 . Si $x \in Z$, alors il a deux voisins u et v de couleurs différentes, et on a $u, v \in T_1 \cup \dots \cup T_5 \cup W_2 \cup W_3$. Si u et v sont adjacents, alors $\{x, \bar{x}, u, v\}$ induit un K_4 . Donc supposons que u et v ne soient pas adjacents. D'après les propriétés 1. et 5. du lemme 3.11, ceci conduit aux sous-cas suivants :

Cas 2.1 : $u \in T_5$ et $v \in T_2 \cup T_3$. Dans ce cas, la propriété (vi) est vérifiée.

Cas 2.2 : $u \in T_1$ et $v \in T_4$. Dans ce cas, la propriété (vi) est vérifiée.

Cas 2.3 : $u \in T_j$ et $v \in W_j$ pour $j \in \{2, 3\}$. Dans ce cas, la propriété 3. du lemme 3.11 est contredite. Ceci termine la preuve. □

Lemme 3.15 Si l'algorithme déclare que G n'est pas 3-colorable, alors G contient un membre de \mathcal{F} .

Démonstration. Notons tout d'abord que si l'algorithme déclare que G n'est pas 3-colorable, ceci est correct en vertu des arguments exposés à la section 3.2.5. À présent, puisque G n'est pas 3-colorable, la caractérisation donnée par [59] implique que G contient un membre de \mathcal{F} . Nous allons à présent montrer comment l'algorithme peut trouver un tel sous-graphe en temps linéaire.

Pour tout $h \in \{1, \dots, 5\}$, l'algorithme trouve que la précoloration γ_h ne s'étend pas à une 3-coloration. Donc chaque γ_h satisfait l'une des propriétés (i)-(vi) du lemme précédent, et l'algorithme trouve un ensemble de sommets A_h qui valide cette propriété. Autrement dit, et pour être plus précis, si la propriété (i) du lemme précédent est vérifiée, A_h consiste en un sommet double de $R \setminus R_h$; si (ii) est vérifiée, A_h consiste en un sommet w de $W_{h-1} \cup W_h \cup W_{h+1}$; si (iii) est vérifiée, A_h consiste en des sommets adjacents t et t' avec soit $t \in T_{h+1}$ et $t' \in T_{h+3}$, soit $t \in T_{h-1}$ et $t' \in T_{h-3}$; si (iv) est vérifiée, A_h consiste en trois sommets x, t, t' tels que $x \in R_h$, $t \in T_{h+2} \cup W_{h+3}$, $t' \in T_{h-2} \cup W_{h-3}$ avec x adjacent à t et t' ; si (v) est vérifiée, A_h consiste en un sommet $z \in Z$ et trois voisins t, t', t'' de z avec $t \in T_h$, $t' \in T_{h-2}$ et $t'' \in T_{h+2}$; et si (vi) est vérifiée, A_h consiste en un sommet double $z \in Z$ et deux voisins t, t' de z avec soit $t \in T_h$ et $t' \in T_{h-2} \cup T_{h+2}$ soit $t \in T_{h-1}$ et $t' \in T_{h+1}$. Notons que dans chaque cas, nous avons $|A_h| \leq 4$. D'après le lemme 3.11 8, pour chaque i l'ensemble R_i contient un sommet a_i complet à $T_{i+2} \cup W_{i+3}$ et un sommet b_i complet à $T_{i-2} \cup W_{i-3}$ (avec potentiellement $a_i = b_i$). Soit H le sous-graphe de G induit par le sous-ensemble $\{a_1, b_1, \dots, a_5, b_5\} \cup A_1 \cup \dots \cup A_5$. Alors nous savons que H lui-même n'est pas 3-colorable, puisque chaque précoloration γ_h satisfait une propriété du lemme 3.14 dans H . Donc H contient un

membre de \mathcal{F} . Notons que H contient au plus 30 sommets, donc trouver un membre de \mathcal{F} dans H peut se faire en temps constant.

Nous analysons à présent cette situation, ce qui prouvera le théorème 3.7. Tout au long de cette preuve, r_i désigne un sommet arbitraire de R_i pour tout $i \in \{1, \dots, 5\}$; par exemple r_i peut désigner a_i ou b_i .

Cas 1 : R contient un sommet double.

On peut supposer, à symétrie près, que R_5 contient un sommet double x et, par le lemme 3.12, que $R \setminus R_5$ ne contient pas de sommet double. Donc γ_5 ne satisfait pas la propriété (i) du lemme 3.14.

Supposons que γ_5 satisfasse (ii); alors il existe un sommet $w \in W_4 \cup W_5 \cup W_1$. Si w est dans W_5 , alors $\{x, \bar{x}, w, r_1, a_2, b_3, r_4\}$ induit un F_5 dans G_0 . Si $w \in W_4 \cup W_1$, alors il est adjacent à x et a un voisin $u \in R_1 \cup R_4$, donc $\{x, \bar{x}, w, u\}$ induit un K_4 .

Supposons que γ_5 satisfasse (iii); alors, à symétrie près, il existe une arête $t_1 t_3$ avec $t_1 \in T_1$ et $t_3 \in T_3$. Nous savons que $t_1 \sim b_3$, $t_1 \sim a_4$ et $t_3 \sim a_1$. Si t_3 est adjacent à x , alors $\{x, \bar{x}, a_1, t_3\}$ induit un K_4 . Si t_3 n'est pas adjacent à x , alors $\{x, \bar{x}, a_1, b_3, a_4, t_1, t_3\}$ induit un F_5 .

Supposons que γ_5 satisfasse (iv); alors un sommet u de R_5 a des voisins $v \in T_2 \cup W_3$ et $w \in T_3 \cup W_2$. On sait que $v \sim r_2$, $v \sim b_4$, $w \sim r_3$, $w \sim a_1$, et $v \approx r_3$, $w \approx r_2$. D'après la propriété 5 du lemme 3.11, v et w sont adjacents. Si $v \sim x$, alors $\{x, \bar{x}, b_4, v\}$ induit un K_4 . Si $v \sim a_1$, alors $\{u, v, w, a_1\}$ induit un K_4 . Donc $v \approx x$ et $v \approx a_1$, et de manière analogue, $w \approx x$ et $w \approx b_4$. Alors $\{x, \bar{x}, u, v, w, b_4, a_1\}$ induit un F_5 .

Supposons que γ_5 satisfasse (v); alors il existe un sommet $z \in Z$ qui a un voisin $t_2 \in T_2$. D'après la propriété 3 du lemme 3.11, t_2 est complet à $R_4 \cup R_5$. Alors $\{x, \bar{x}, t_2, r_4\}$ induit un K_4 .

Supposons que γ_5 satisfasse (vi) pour un sommet double $z \in Z$. Si z a un voisin dans T_2 , alors on peut raisonner comme quand γ_5 satisfait (v) ci-dessus. Donc supposons que z a des voisins $t_1 \in T_1$ et $t_4 \in T_4$. Toujours selon la propriété 3 du lemme 3.11, t_1 est complet à $R_3 \cup R_4$ et t_4 est complet à $R_1 \cup R_2$. Si $t_1 \sim t_4$, alors $\{z, \bar{z}, t_1, t_4\}$ induit un K_4 . Et si $t_1 \approx t_4$, alors $\{x, \bar{x}, z, \bar{z}, t_1, t_4, r_1, r_2, r_3, r_4\}$ induit un F_{10} . Ceci termine la preuve du cas 1.

Supposons à présent qu'une des précolorations γ_h satisfasse (vi). Alors il existe un sommet double $z \in Z$ et un entier $i \in \{1, \dots, 5\}$ tels que z a un voisin $t_i \in T_i$ et un voisin $t_{i+2} \in T_{i+2}$. Si $t_i \sim t_{i+2}$, alors $\{z, \bar{z}, t_i, t_{i+2}\}$ induit un K_4 . Si $t_i \approx t_{i+2}$ alors d'après la propriété 3 du lemme 3.11, $\{z, t_i, r_{i+3}, r_{i+4}, t_{i+2}\}$ induit un C_5 . On peut ré-exécuter l'algorithme à partir de ce C_5 , et z sera un sommet double dans le nouvel ensemble R , ce qui nous permet de conclure comme dans le cas 1. Par conséquent, on supposera désormais qu'aucune γ_h ne satisfait (vi).

Cas 2 : deux des ensembles W_1, \dots, W_5 sont non vides.

D'après le lemme 3.11 3 et à symétrie près, on peut supposer qu'il existe des sommets $w_2 \in W_2$ et $w_3 \in W_3$ et que $W_1 \cup W_4 \cup W_5 = \emptyset$. Donc γ_5 ne satisfait pas (ii). D'après ce qui précède, on sait aussi que γ_5 ne satisfait ni (i) ni (vi). D'après le lemme 3.11 7, il n'y a pas d'arête entre T_1 et T_3 ni entre T_2 et T_4 , ni entre $T_2 \cup T_3$ et Z , donc γ_5 ne satisfait ni (iii) ni (v).

Par conséquent, γ_5 satisfait (iv), c'est-à-dire qu'un sommet $a \in R_5$ a un voisin u dans $W_2 \cup T_3$ et un voisin v dans $W_3 \cup T_2$ (avec potentiellement $u = w_2$ ou $v = w_3$). D'après les propriétés 1 et 5 du lemme 3.11, u et w_2 sont adjacents à v et w_3 . Si $u \in W_2$ et $v \in W_3$ alors $\{a_4, a, u, v\}$ induit un K_4 . Par conséquent, et à symétrie près, que $u \in T_3$; donc $u \neq w_2$. Nous savons que $w_2 \sim a_4$, $w_3 \sim b_1$, $w_2 \sim b_5$, $u \sim b_5$, $w_3 \sim a_5$, $v \sim a_5$ et $u \sim a_1$. On peut supposer que $b_5 \approx w_3$ (autrement on prend b_5, w_2, w_3 à la place de a, u, v) et de même, $a_5 \approx w_2$. Donc $b_5 \neq a_5$.

Supposons que $v \in W_3$. Si $a \sim w_2$, alors $\{a, v, w_2, a_4\}$ induit un K_4 ; donc $a \approx w_2$. Alors $a \neq b_5$ et, par le même argument, $v \approx b_5$. À présent, si $v \sim a_1$, alors $\{v, a_1, a, u\}$ induit un K_4 et dans le cas contraire $\{a, u, v, a_1, w_2, a_4, b_5\}$ induit un F_7 .

Supposons à présent que $v \in T_2$; donc $v \neq w_3$ et la symétrie entre u et v est restaurée. Alors $a \approx w_2$ (sinon, on prend $u = w_2$), et de même $a \approx w_3$. Si $u \sim b_1$ et $v \sim a_4$, alors $\{a, u, v, w_2, w_3, b_1, a_4\}$ induit un F_9 ($= \overline{C}_7$). Donc on peut supposer, à symétrie près, que $u \approx b_1$ et, par le même argument, que $w_3 \approx a_1$; mais alors $\{u, a_1, b_1, w_2, w_3, b_5, a_5\}$ induit un F_9 .

Cas 3 : exactement l'un des ensembles W_1, \dots, W_5 n'est pas vide.

À symétrie près, supposons que W_1 ne soit pas vide et $W_2 \cup W_3 \cup W_4 \cup W_5 = \emptyset$. Alors γ_3 et γ_4 ne satisfont pas (ii). De plus, d'après ce qui précède, γ_3 et γ_4 ne satisfont pas non plus (i) ou (vi). D'après la propriété 7 du lemme 3.11, il n'y a pas d'arête entre T_1 et Z , donc γ_3 et γ_4 ne satisfont pas (v). Par conséquent, γ_3 et γ_4 vérifient (iii) ou (iv). Notons également que, d'après la propriété 7 du lemme 3.11, il n'y a pas d'arête entre T_2 et T_5 . Par conséquent, pour γ_3 soit il existe une T_1T_4 -arête soit un sommet de R_3 qui a un voisin dans $T_5 \cup W_1$ et un voisin dans $T_1 \cup W_5$; et pour γ_4 , soit il existe une T_1T_3 -arête soit un sommet de R_4 qui a un voisin dans $T_1 \cup W_2$ et un voisin dans $T_2 \cup W_1$. À symétrie près, ceci conduit à considérer les deux sous-cas ci-dessous. Tout d'abord, on choisit un sommet $w_1 \in W_1$. On sait que w_1 est complet à $R_2 \cup R_5$, anticomplet à R_1 et adjacent à a_3 et b_4 .

Cas 3.1 : il existe (pour γ_3) une arête t_1t_4 avec $t_1 \in T_1$ et $t_4 \in T_4$.

D'après le lemme 3.11, on sait que $w_1 \sim t_1$, $w_1 \sim t_4$, $t_1 \sim b_3$, $t_1 \sim a_4$, $t_4 \sim b_4$ et que $t_4 \approx b_3$ et $t_4 \approx a_3$. Si $t_1 \sim b_4$, alors $\{t_1, b_4, w_1, t_4\}$ induit un K_4 . Si $w_1 \sim a_4$ alors $\{w_1, a_4, t_1, t_4\}$ induit un K_4 . Donc on peut supposer que $t_1 \approx b_4$ et $w_1 \approx a_4$, et donc $a_4 \neq b_4$.

Supposons que (pour γ_4) il existe une arête s_1s_3 avec $s_1 \in T_1$ et $s_3 \in T_3$ (avec potentiellement $s_1 = t_1$). Comme ci-dessus, on sait que $w_1 \sim s_1$, $w_1 \sim s_3$, $s_1 \sim b_3$, $s_1 \sim a_4$, $s_3 \sim b_3$ et que $s_3 \approx a_4$, $s_3 \approx b_4$, $w_1 \approx b_3$, et que $a_3 \neq b_3$. D'après la propriété 1 du lemme 3.11, $s_3 \sim t_4$ et $s_1 \approx t_1$. Si de plus $t_1 \sim s_3$, alors $\{t_1, s_3, w_1, t_4\}$ induit un K_4 . Ainsi, $t_1 \sim s_3$ et de même $s_1 \approx t_4$. Il s'ensuit que $t_1 \neq s_1$, mais alors $\{w_1, s_1, t_1, s_3, t_4, b_3, a_4\}$ induit un F_9 .

Supposons à présent que (pour γ_4) un sommet $b \in R_4$ ait un voisin $v \in W_1 \cup T_2$ et un voisin $s_1 \in T_1$ (avec potentiellement $v = w_1$, ou $s_1 = t_1$, ou $b \in \{a_4, b_4\}$). D'après la propriété 8 du lemme 3.11, nous savons que $t_4 \sim a_2$, $w_1 \sim a_2$, $w_1 \sim s_1$, $v \sim t_1$, $v \sim s_1$, $v \sim a_2$, $v \sim b_4$, $s_1 \sim b_3$, $s_1 \sim a_4$, $t_4 \sim b$ et que $s_1 \approx t_1$, $s_1 \sim a_2$, $t_1 \sim a_2$ et $v \approx w_1$.

Supposons que $b \sim w_1$; alors $b \neq a_4$. Si $b \sim t_1$, alors $\{b, t_1, w_1, t_4\}$ induit

un K_4 ; si $b \approx t_1$, alors $t_1 \neq s_1$ et l'ensemble $\{w_1, t_1, s_1, t_4, b, b_3, a_4\}$ contient un K_4 ou un F_9 .

Supposons désormais que $b \approx w_1$; alors $b \neq b_4$, $v \neq w_1$, et on peut supposer que $v \in T_2$ (sinon on prend $w_1 = v$). Nous savons que $v \sim a_5$ et $w_1 \sim a_5$ et $v \approx a_3$. Si $b \approx t_1$, alors $s_1 \neq t_1$, $b \neq a_4$, et l'ensemble $\{w_1, a_4, b, b_4, t_1, s_1, a_5, t_4, b_3, v\}$ contient un K_4 , un F_9 ou un F_{11} (selon la présence ou non des arêtes t_4s_1 , vt_4 , s_1b_4 , b_3w_1 , va_4). Par conséquent, $b \sim t_1$. Maintenant, si $t_1 \sim a_3$, alors $\{w_1, b, b_4, t_1, a_5, a_3, v\}$ induit un F_9 ; si $t_1 \approx a_3$ et $w_1 \sim b_3$ alors $\{w_1, b, b_4, t_1, a_5, b_3, v\}$ induit un F_9 ; enfin, si $t_1 \approx a_3$ et $w_1 \approx b_3$ alors $\{v, t_4, a_5, t_1, b_4, b, w_1\}$ contient un K_4 ou un F_9 (selon que $v \sim t_4$ ou non).

Cas 3.2 : un sommet $a \in R_3$ a un voisin $u \in W_1 \cup T_5$ et un voisin $t_1 \in T_1$, et un voisin $b \in R_4$ a un voisin $v \in W_1 \cup T_2$ et un voisin $s_1 \in T_1$.

D'après les propriétés 1, 5 et 8 du lemme 3.11, nous savons que $\{b_2, a_5, t_1, s_1\}$ est stable et complet à $\{r_1, u, v, w\}$, et que u et v ne sont pas adjacents à r_1 ni w_1 . De plus, t_1 et s_1 sont adjacents à b_3 et a_4 , et $u \sim a_3$ et $v \sim b_4$.

Supposons que $a \sim w_1$ et $b \sim w_1$. Si $b \sim t_1$, alors $\{b, t_1, u, a\}$ induit un K_4 . Donc $b \approx t_1$; de même $a \approx s_1$. Alors $a \neq b_3$, $b \neq a_4$, et $\{u, a, b, t_1, s_1, b_3, a_4\}$ contient un K_4 ou un F_9 .

Supposons à présent que $b \sim w_1$ et $a \approx w_1$. Donc $w_1 \neq u$ et $a_3 \neq a$. Si $s_1 \sim a_3$ alors $\{s_1, a_3, b, w_1\}$ induit un K_4 ; donc posons $s_1 \approx a_3$. Alors $b_3 \neq a_3$. Si on peut choisir $s_1 = t_1$, alors $\{s_1, a_3, a, w_1, u, b, b_2\}$ induit un F_9 . Donc posons $s_1 \neq t_1$, et par le même argument $a \approx s_1$ et $b \approx t_1$. Donc $b_3 \neq a$ et $a_4 \neq b$. Si $u \sim b_3$ alors $\{s_1, a_3, b_3, w_1, u, b, b_2\}$ contient un K_4 ou un F_9 (selon que $b_3 \sim w_1$ ou non). Donc posons $u \approx b_3$. Si $w_1 \sim a_4$, alors $\{w_1, u, a_4, b_2, t_1, a_3, a\}$ contient un K_4 ou un F_9 (selon la présence ou non des arêtes ua_4 et t_1a_3). Donc on pose $w_1 \approx a_4$, ce qui implique que $a_4 \notin \{b, b_4\}$; mais alors $\{s_1, a_3, a, b_3, w_1, u, a_4, b, b_4, t_1\}$ contient un K_4 ou un F_9 ou un F_{11} (selon la présence ou non des arêtes t_1a_3 , t_1b_4 , s_1b_4).

Par conséquent, on peut supposer que $a \sim w_1$ et $b \approx w_1$, et plus généralement que a et b n'ont pas de voisin dans W_1 . Ainsi on a $u \in T_5$, $v \in T_2$ et $w_1 \notin \{u, v\}$. On sait aussi que $a_3 \neq a$ et $b_4 \neq b$. Comme $v \in T_2$, nous avons $v \approx a$ et $v \approx a_3$ et de même $u \approx b$ et $u \approx b_4$. Si on peut choisir $s_1 = t_1$, alors $\{s_1, a_5, a_3, a, v, w_1, u, b, b_4, b_2\}$ contient un K_4 ou un F_9 ou F_{11} (selon la présence ou non des arêtes uv , s_1a_3 , s_1b_4). Par conséquent, posons $s_1 \neq t_1$ et par le même argument posons $a \approx s_1$ et $b \approx t_1$, d'où $b_3 \neq a$ et $a_4 \neq b$. Nous savons que $v \approx b_3$ et $u \approx a_4$. Si $t_1 \sim b_4$ alors $\{t_1, a_3, a, w_1, u, b_4, b_2\}$ contient un K_4 ou un F_9 (selon que $t_1 \sim a_3$ ou non); donc on pose $t_1 \approx b_4$ et de manière similaire $s_1 \approx a_3$; donc $b_3 \neq a_3$ et $a_4 \neq b_4$. Mais alors G contient un K_4 ou un F_9 ou un F_{11} ou un F_{12} induit par des sommets de l'ensemble $\{a, b_3, v, w_1, u, a_4, b, b_4, b_2, t_1, s_1, a_5, a_3\}$, selon que l'on a ou non les arêtes uv , w_1b_3 , w_1a_4 , t_1a_3 , s_1b_4 , ub_3 , va_4 . Ceci termine la preuve du cas 3.

Maintenant, nous pouvons supposer que les ensembles W_1, \dots, W_5 sont vides. Soit J le graphe à six sommets x_0, x_1, \dots, x_5 tels que $\{x_1, \dots, x_5\}$ induit un C_5 et x_0 est adjacent à exactement quatre sommets de ce C_5 . Si G contient une copie de J , alors on peut recommencer la procédure du

début de cette section en commençant par les sommets x_1, \dots, x_5 . Dans ce cas, le sommet x_0 est placé dans l'ensemble W_i correspondant et on peut conclure comme dans les cas 2 ou 3 ci-dessus. Par conséquent, on peut supposer que G est sans J . De plus, la propriété suivante, notée (T), est vérifiée :

(T) Pour toute arête xy avec $x \in T_{i-1}$ et $y \in T_{i+1}$, x est complet à R_{i+2} et y est complet à R_{i-2} .

En effet, si x a un non-voisin $z \in T_{i+2}$, alors $\{a_{i-1}, b_{i+1}, b_{i-2}, x, y, z\}$ induit un J et la même chose vaut pour y par symétrie.

Cas 4 : une précoloration γ_h vérifie (v).

Admettons que γ_h satisfasse (v). Alors il y a un sommet $z \in Z$ qui a des voisins $t_2 \in T_2$, $t_3 \in T_3$ et $t_5 \in T_5$. D'après la propriété 1 du lemme 3.11, on a $t_2 \sim t_3$, et la propriété 4 du même lemme s'applique à t_2, t_3, t_5 . Si on a à la fois les arêtes t_2t_5 et t_3t_5 , alors $\{t_5, t_2, t_3, z\}$ induit un K_4 . Si on a seulement, par exemple, l'arête t_2t_5 , alors $\{r_1, r_2, z, t_5, t_2, t_3\}$ induit un J . Supposons donc que $t_2 \approx t_5$ et $t_3 \approx t_5$, et considérons un sommet $t_1 \in T_1$. D'après les propriétés 1 et 8 du lemme 3.11, on sait que $t_1 \sim t_2$, $t_1 \sim t_5$, $t_1 \sim b_3$, $t_1 \sim a_4$. Alors $\{b_3, a_4, t_2, z, t_5, t_1\}$ induit un J ou un F_2 . Par conséquent $T_1 = \emptyset$, et de même $T_4 = \emptyset$. Donc γ_3 ne satisfait ni (iv) ni (v). D'après les points précédents, γ_3 ne satisfait pas non plus (i), (ii) ni (vi). Donc γ_3 satisfait (iii) et alors (puisque $T_1 \cup T_4 = \emptyset$), il existe une arête s_5s_2 avec $s_5 \in T_5$ et $s_2 \in T_2$. Supposons que s_5 soit adjacent à z . Alors (comme pour t_5), s_5 est complet à $R_2 \cup R_3$, $s_5 \approx t_2$, $s_5 \approx t_3$ et $s_2 \approx z$; mais alors $\{r_1, r_2, s_5, z, t_3, s_2\}$ induit un J . Supposons à présent que s_5 ne soit pas adjacent à z ; alors $s_5 \neq t_5$. On sait que $s_5 \sim b_2$ et $s_5 \sim a_3$. Alors $s_5 \sim t_2$ sinon les sommets z, t_2 et s_5 violent la propriété (11) du lemme 3.11, et par un argument similaire $s_5 \sim t_3$. Mais finalement $\{s_5, t_2, t_3, u_5\}$ induit un K_4 , ce qui complète la preuve du cas 4.

On en déduit que toute précoloration γ_h vérifie soit (iii) soit (iv).

Cas 5 : pour un i , il existe une T_iT_{i+2} -arête et une T_iT_{i-2} -arête.

Supposons qu'il existe des arêtes t_5t_2 et t'_5t_3 avec $t_5, t'_5 \in T_5$, $t_2 \in T_2$ et $t_3 \in T_3$. Nous savons que t_5 et t'_5 sont adjacents à b_2 et a_3 et que $t_2 \sim t_3$, $t_2 \sim b_4$, $t_2 \sim a_5$, $t_3 \sim a_1$, $t_3 \sim b_5$. On peut également supposer qu'on a soit $t'_5 = t_5$, soit que $t'_5 \approx t_2$ et $t_5 \approx t_3$. Si un sommet $x \in R_5$ est adjacent à la fois à t_2 et t_3 , alors soit $\{x, t_2, t_3, t_5\}$ induit un K_4 (si $t'_5 = t_5$), soit $\{x, b_2, t_3, t_5, t'_5, t_2, a_3\}$ induit un F_9 (sinon). Donc on peut supposer désormais qu'il n'existe pas de tel x ; donc $a_5 \sim t_3$ et $b_5 \sim t_2$.

Supposons que γ_5 satisfasse (iii). Alors, à symétrie près, il existe une arête $t_1t'_3$ avec $t_1 \in T_1$ et $t'_3 \in T_3$. Nous avons $t_1 \sim b_3$. On peut raisonner avec les arêtes $t_1t'_3$ et t'_5t_3 comme ci-dessus avec les arêtes t_5t_2 et t'_5t_3 , donc nous savons que $t_1 \approx a_3$ et $t'_5 \approx b_3$. D'après la propriété (T), t'_3 est complet à R_5 , donc $t'_3 \neq t_3$. De la même façon, t_5 est complet à R_3 donc $t_5 \neq t'_5$. Mais alors $\{a_1, t_2, a_3, b_5, t_1, b_2, t_3, t'_3, t_5, t'_5\}$ induit un F_{11} .

Supposons que γ_5 satisfasse (iv). Alors il existe un sommet $a \in R_5$ qui a des voisins $s_2 \in T_2$ et $s_3 \in T_3$. D'après l'argument sur x ci-dessus, et à symétrie près, on peut supposer que $a \approx t_2$. Donc $s_2 \neq t_2$, et on peut supposer que $s_2 \approx t_5$ et $s_2 \approx t'_5$ (sinon nous pourrions prendre $t_2 = s_2$). Si $s_2 \sim b_5$, alors soit $t'_5 = t_5$ et $\{b_5, a_5, t_3, b_4, t_5, s_2, t_2\}$ induit un F_9 , soit

$t'_5 \neq t_5$ et $\{b_5, a_5, b_2, t_3, b_4, t_5, t'_5, s_2, t_2, a_3\}$ induit un F_{11} . Donc on peut poser $s_2 \approx b_5$. Par le même argument, avec a à la place de b_5 , on obtient que $a \approx t_3$. Par conséquent $s_3 \neq t_3$ et comme ci-dessus on peut supposer que $s_3 \approx t_5$, $s_3 \approx t'_5$ et $s_3 \approx a_5$. Alors $\{a, a_5, v_2, t_3, s_3, b_4, t'_5, t_5, a_1, s_2, t_2, a_3, b_5\}$ induit un F_{12} .

Cas 6 : pour un i , il existe une $T_i T_{i+2}$ -arête et une $T_{i+1} T_{i+3}$ -arête.

Supposons qu'il existe des arêtes $t_1 t_3$ et $t_2 t_4$, avec $t_i \in T_i$ pour chaque $i = 1, 2, 3, 4$. Nous savons que $t_1 \sim t_2$, $t_2 \sim t_3$, $t_3 \sim t_4$, $t_1 \sim b_3$, $t_2 \sim b_4$, $t_3 \sim a_1$ et $t_4 \sim a_2$ et, d'après la propriété (T), que $t_1 \sim b_4$, $t_4 \sim a_1$, $t_2 \sim r_5$ et $t_3 \sim r_5$. Nous pouvons supposer que nous ne sommes plus dans le cas 5, donc il n'existe pas d'arête entre T_1 et T_4 ni entre $T_2 \cup T_3$ et T_5 . Par conséquent γ_2 et γ_3 satisfont (iv), donc il existe un sommet $t_5 \in T_5$. D'après la propriété 3 du lemme 3.11 et à symétrie près, on peut supposer que t_5 est complet à R_2 . Nous savons que $t_5 \sim a_3$. Si $t_5 \sim b_3$, alors $\{t_5, t_3, a_2, t_1, t_4, b_3, t_2\}$ induit un F_9 . Par conséquent, on pose $t_5 \approx b_3$; donc $a_3 \approx b_3$ et par le même argument $t_1 \approx a_3$. Alors si a_3 a un voisin $t'_1 \in T_1$ on peut supposer que $t'_1 \approx t_3$ (sinon prendre $t_1 = t'_1$), et alors $\{t_1, t'_1, a_2, t_3, b_4, t_5, a_1, b_3, a_3, r_5\}$ induit un F_{11} . Par conséquent, admettons que a_3 soit anticomplet à T_1 ; supposons également que b_3 ait un voisin $t'_5 \in T_5$. Nous pouvons dire que $t'_5 \approx a_2$ (sinon prendre $t_5 = t'_5$), donc, d'après la propriété 3 du lemme 3.11, t'_5 est complet à R_3 , et nous savons que t_5 et t'_5 sont adjacents à b_2 . Alors $G[\{b_2, a_2, b_4, t'_5, t_5, t_2, b_3, a_3, t_4, t_1\}]$ contient un K_4 ou un F_{11} (selon que $t_4 \sim b_2$ ou non). Par conséquent, disons que b_3 est anticomplet à T_5 . Comme γ_3 satisfait (iv), il y a un sommet $y \in R_3$ qui a un voisin $s_1 \in T_1$ et un voisin $s_5 \in T_5$. Si $y \sim t_1$, on peut conclure comme précédemment, avec y à la place de b_3 . Donc disons que $y \approx t_1$ et, par le même argument, que $s_1 \approx t_3$. Comme a_3 est anticomplet à T_1 , on a $a_3 \approx s_1$ et donc, d'après la propriété 3 du lemme 3.11, s_1 est complet à R_4 . De la même façon, $b_3 \approx s_5$, donc $s_5 \approx a_2$. Mais finalement, $\{s_1, a_2, t_3, b_4, s_5, a_1, b_3, y, t_4, t_1\}$ induit un F_{11} . Ceci termine la preuve du cas 6.

Désormais, nous pouvons supposer que nous ne sommes pas dans les cas 1 à 6 qui précèdent. Donc il existe une $T_i T_{i+2}$ -arête pour au plus une valeur de i , et nous pouvons supposer à symétrie près que s'il existe une telle arête, alors $i = 4$. Par conséquent, γ_1 , γ_4 et γ_5 ne satisfont pas (iii), et donc elles satisfont (iv); autrement dit, pour tout $j \in \{1, 4, 5\}$, il existe un sommet $u_j \in R_j$ qui a des voisins $t_{j+2} \in T_{j+2}$ et $s_{j+3} \in T_{j+3}$. Nous savons également que T_1 est complet à T_2 et anticomplet à T_3 , que T_4 est complet à T_3 et anticomplet à T_2 , et que T_2 est complet à T_3 .

Cas 7 : Il existe une $T_4 T_1$ -arête.

Soit $t_4 s_1$ une arête avec $t_4 \in T_4$ et $s_1 \in T_1$. On choisit des sommets t_i et s_i ($i \in \{1, 2, 3, 4\}$) tels que $t_i = s_i$ si possible, c'est-à-dire si un sommet de T_i peut jouer à la fois le rôle de t_i et de s_i (donc si, par exemple, $t_2 \neq s_2$, alors u_4 n'est pas adjacent à t_2 et u_5 n'est pas adjacent à s_2 . Ceci implique aussi que $u_4 \neq a_4$ et $u_5 \neq a_5$). Considérons l'ensemble $D = \{i \in \{1, 2, 3, 4\} | t_i \neq s_i\}$.

Si $D = \emptyset$, alors $\{u_1, u_4, t_3, t_1, u_5, t_4, t_2\}$ induit un F_9 .

Supposons à présent que D contienne 1. Alors s_1 n'est pas adjacent à u_4 , t_1 n'a pas de voisin dans T_4 et $u_4 \neq a_4$. Si nous avons aussi $3 \in D$, alors s_3 n'est pas adjacent à u_1 , donc on a $s_3 \in T_3^{-R_1}$ et $t_1 \in T_i^{-T_4}$, et ces deux sommets violent la propriété 10 du lemme 3.11. Par conséquent $3 \notin D$, c'est-à-dire $t_3 = s_3$. Supposons maintenant que $4 \in D$. Alors t_4 n'est pas adjacent à u_1 , s_4 n'a pas de voisin dans T_1 et $u_1 \neq b_1$. De plus, par le même argument que précédemment, on a $2 \notin D$, i.e. $t_2 = s_2$. Alors l'ensemble de sommets $\{u, b_1, a_4, u_4, t_3, a_2, s_1, t_1, u_5, s_4, t_4, b_3, t_2\}$, avec les arêtes optionnelles b_1t_3 et a_4t_2 , implique que G contient un K_4 ou un F_9 ou un F_{11} ou un F_{12} . Donc $4 \notin D$. Si $D = \{1\}$, alors $\{u_1, a_4, u_4, t_3, s_1, t_1, u_5, t_4, b_3, t_2\}$ contient un K_4 ou un F_9 ou un F_{11} selon que $a_4 \sim t_2$ ou non. Ainsi, $D = \{1, 2\}$. Donc $u_4 \notin \{a_4, b_4\}$. Si $a_4 = b_4$, alors $\{u_1, a_4, t_3, s_1, u_5, t_4, t_2\}$ induit un F_9 . Si $a_4 \neq b_4$, alors $\{u_1, a_4, u_4, b_4, t_3, s_1, t_1, a_5, u_5, t_4, b_3, s_2, t_2\}$ contient l'un des graphes K_4, F_9, F_{11} ou F_{12} (selon la présence ou non des arêtes $a_4s_2, a_4t_2, b_4s_1, b_4t_1$). On peut désormais supposer que D ne contient ni 1 ni 4.

Supposons que $D = \{2\}$. Alors on a $u_4 \neq b_4$ et $u_5 \neq a_5$. Mais alors $\{u_1, u_4, b_4, t_3, t_1, a_5, u_5, t_4, s_2, t_2\}$ contient l'un des graphes K_4, F_9 ou F_{11} (selon la présence ou non des arêtes b_4t_1 et t_3a_5). Par symétrie, il en va de même si $D = \{3\}$.

Enfin, posons $D = \{2, 3\}$. Alors nous avons $u_1 \neq a_1, u_4 \neq b_4$ et $u_5 \notin \{a_5, b_5\}$. Si $a_5 = b_5$, alors $\{u_1, u_4, t_3, t_1, a_5, t_4, s_2\}$ induit un F_9 . Et si $a_5 \neq b_5$, alors $\{u_1, u_4, b_4, s_3, t_3, t_1, a_5, u_5, b_5, t_4, s_2, t_2, a_1\}$ contient l'un des graphes K_4, F_9, F_{11}, F_{12} (selon la présence ou non des arêtes $a_1t_4, b_4t_1, a_5s_3, a_5t_3, b_5s_2, b_5t_2$). Ceci termine la preuve du cas 7.

Cas 8 : pour tout i , il n'y a pas de T_iT_{i+2} -arête.

Désormais, toute précoloration γ_j ($j \in \{1, \dots, 5\}$) satisfait (iv). Donc il existe un sommet $u_j \in R_j$ qui a des voisins $t_{j+2} \in T_{j+2}$ et $s_{j+3} \in T_{j+3}$. Supposons que pour un j nous ne puissions pas choisir $s_j = t_j$, disons pour $j = 2$; en d'autres termes, $t_2 \approx u_4$ et $s_2 \approx u_5$, et $u_4 \neq b_4$ et $u_5 \neq a_5$. Alors $\{u_2, s_2, u_4, u_5, t_2\}$ induit un C_5 , donc on peut appliquer l'algorithme sur ce C_5 , en partant des ensembles $R'_1 = \{u_2\}$, $R'_2 = \{s_2\}$, $R'_3 = \{u_4\}$, $R'_4 = \{u_5\}$, $R'_5 = \{t_2\}$ et en faisant grossir la C_5 -structure correspondante. Notons que b_4 et a_5 ont des voisins dans au trois des R'_1, \dots, R'_5 . Donc soit l'un des sommets b_4 ou a_5 a des voisins dans au moins quatre de ces ensembles, et alors G contient un F_2 ou un J et l'on peut conclure comme ci-dessus; soit on a $b_4 \in T'_2$ et $a_5 \in T'_5$ (avec la notation habituelle), donc b_4a_5 est une $T'_2T'_5$ -arête et on peut raisonner comme dans le cas précédent. Par conséquent, on peut supposer que $t_j = s_j$ pour chaque $j \in \{1, \dots, 5\}$. Mais alors $\{u_1, \dots, u_5, t_1, \dots, t_5\}$ induit un F_{11} . Ceci achève la preuve du lemme. \square

Si G est 3-colorable

Dans cette section, on analyse la situation où G est 3-colorable. On peut supposer, à symétrie près, que l'algorithme a trouvé que γ_5 s'étend à une 3-coloration γ de G . Donc on a $\gamma(R_1 \cup R_4) = 1$, $\gamma(R_2) = 2$, $\gamma(R_3) = 3$, $\gamma(R_5) \subseteq \{2, 3\}$ et nous savons que $\gamma(W_2 \cup T_1 \cup T_3) = 2$, $\gamma(W_3 \cup T_2 \cup T_4) = 3$, $\gamma(T_5) = 1$ et qu'aucune des propriétés (i)-(vi) du lemme 3.14 n'est vérifiée pour $h = 5$; en particulier, $W_1 \cup W_4 \cup W_5 = \emptyset$.

Pour tout $i \in \{1, \dots, 5\}$, on définit un ensemble $U_i = T_{i-1}^{-T_{i+1}} \cup T_{i+1}^{-T_{i-1}}$. D'après la propriété 11 du lemme 3.11, nous savons que chaque sommet de Z est soit complet soit anticomplet à U_i . Soit Z_j l'ensemble des sommets de Z qui sont anticomplets à U_i .

Lemme 3.16 *Tout stable de G est un sous-ensemble de l'un des neuf ensembles*

$W'_2, W'_3, W''_2, W''_3, T'_2, T'_3, U'_1, U'_4, U'_5$ *qui sont définis comme suit :*

- $W'_2 = W_2 \cup R_2 \cup R_4 \cup T_1 \cup T_3 \cup T_5^{0Z} \cup Z,$
- $W'_3 = W_3 \cup R_1 \cup R_3 \cup T_2 \cup T_4 \cup T_5^{0Z} \cup Z,$
- $W''_2 = W_2 \cup R_2 \cup R_5 \cup T_1 \cup T_3 \cup T_4^{0Z} \cup Z,$
- $W''_3 = W_3 \cup R_3 \cup R_5 \cup T_2 \cup T_4 \cup T_1^{0Z} \cup Z,$
- $T'_2 = R_1 \cup R_4 \cup T_2^{0Z} \cup T_5 \cup Z,$
- $T'_3 = R_1 \cup R_4 \cup T_3^{0Z} \cup T_5 \cup Z,$
- $U'_j = R_j \cup U_j \cup Z_j$ *pour $j \in \{1, 4, 5\}$.*

Démonstration. Soit S un stable de G . Supposons dans un premier temps que S ne contienne aucun sommet des W_i ou des T_i . Donc $S \subseteq R_1 \cup \dots \cup R_5 \cup Z$, et donc $S \subseteq R_j \cup R_{j+2} \cup Z$ pour un $j \in \{1, \dots, 5\}$ et est donc respectivement un sous-ensemble de W'_3, W'_2, W''_3, T'_2 ou W''_2 .

Supposons à présent que S contienne un sommet de W_2 . D'après la propriété 5 du lemme 3.11, S ne contient pas de sommet de $R_1 \cup R_3 \cup T_2 \cup T_4 \cup T_5$. Donc $S \subseteq W_2 \cup R_2 \cup R_4 \cup R_5 \cup T_1 \cup T_3 \cup Z$. Si S contient aussi un sommet de R_4 , alors S ne contient aucun sommet de R_5 et donc $S \subseteq W_2 \cup R_2 \cup R_4 \cup T_1 \cup T_3 \cup Z$, qui est un sous-ensemble de W'_2 ; sinon $S \subseteq W_2 \cup R_2 \cup R_5 \cup T_1 \cup T_3 \cup Z$, qui est un sous-ensemble de W''_2 .

De même (par symétrie), si S contient un sommet de W_3 , on obtient que S est inclus dans W'_3 ou W''_3 .

Par conséquent, nous pouvons supposer que S ne contient aucun sommet de $W_2 \cup W_3$.

Supposons que S contienne un sommet de T_1 . D'après la propriété 1 du lemme 3.11, S ne contient aucun sommet de $R_1 \cup T_2 \cup T_5$. Donc $S \subseteq R_2 \cup R_3 \cup R_4 \cup R_5 \cup T_1 \cup T_3 \cup T_4 \cup Z$. Si S contient un sommet de R_3 , alors il ne contient aucun sommet de $R_2 \cup R_4$ et, d'après les propriétés 1 et 4 du lemme 3.11, S ne contient aucun sommet de $T_3 \cup T_1^Z$; donc $S \subseteq R_3 \cup R_5 \cup T_1^{0Z} \cup T_3 \cup Z \subseteq W''_3$. Si S contient un sommet de R_4 , alors il ne contient aucun sommet de $R_3 \cup R_5$ et, d'après les propriétés 1 et 4 du lemme 3.11, S ne contient aucun sommet de $T_4 \cup T_1^Z$; donc $S \subseteq R_2 \cup R_4 \cup T_1^{0Z} \cup T_3 \cup Z \subseteq W''_2$. Admettons alors que S ne contienne aucun sommet de $R_3 \cup R_4$. Si S contient un sommet de $R_2 \cup T_3$, alors il ne contient aucun sommet de T_4^Z , et donc $S \subseteq R_2 \cup R_5 \cup T_1 \cup T_3 \cup T_4^{0Z} \cup Z \subseteq W''_2$. Admettons maintenant que S ne contienne pas non plus de sommet de $R_2 \cup T_3$. Donc $S \subseteq R_5 \cup T_1 \cup T_4 \cup Z$. Si S contient un sommet de T_1 complet à T_4 , alors S ne contient aucun sommet de T_4 , donc $S \subseteq R_5 \cup T_1 \cup Z \subseteq W''_2$. De la même façon, si S contient un sommet de T_4 complet à T_1 , alors S ne contient aucun sommet de T_1 donc $S \subseteq R_5 \cup T_4 \cup Z \subseteq W''_3$. Dans le cas restant, nous avons $S \subseteq R_5 \cup T_1^{-T_4} \cup T_4^{-T_1} \cup Z = R_5 \cup U_5 \cup Z$. Si S contient un sommet de $Z \setminus Z_5$ (qui est anticomplet à U_5 d'après la propriété 11 du lemme 3.11, alors on a $S \subseteq R_5 \cup Z \subseteq W''_2$. Sinon, on a $S \subseteq R_5 \cup U_5 \cup Z_5 = U'_5$.

Par symétrie, si S contient un sommet de T_4 , nous obtenons que S est inclus dans l'un des ensembles W_2'', W_3', W_3'', U_5' .

Par conséquent, nous pouvons considérer désormais que S ne contient aucun sommet de $T_1 \cup T_4$.

Supposons que S contienne un sommet de T_2 . D'après la propriété 1 du lemme 3.11, S ne contient aucun sommet de $R_2 \cup T_3$. Donc $S \subseteq R_1 \cup R_3 \cup R_4 \cup R_5 \cup T_2 \cup T_5 \cup Z$. Si S contient un sommet de R_5 , alors il ne contient aucun sommet de R_4, R_1 ni T_5 et donc $S \subseteq R_3 \cup R_5 \cup T_2 \cup Z \subseteq W_3''$. Posons maintenant que S ne contient aucun sommet de R_5 . Si S contient un sommet de R_4 , alors d'après la propriété 4 du lemme 3.11, S ne contient aucun sommet de $R_3 \cup T_2^Z$, et donc $S \subseteq R_1 \cup R_4 \cup T_2^{0Z} \cup T_5 \cup Z = T_2'$. Si S contient un sommet de R_3 , alors d'après la propriété 4 du lemme 3.11, S ne contient aucun sommet de $R_4 \cup T_5^Z$, et donc $S \subseteq R_1 \cup R_3 \cup T_5^{0Z} \cup T_2 \cup Z \subseteq W_3'$. Donc nous posons maintenant que S ne contient pas non plus de sommet de $R_3 \cup R_4$. Donc $S \subseteq R_1 \cup T_2 \cup T_5 \cup Z$. Si S contient un sommet de T_2 complet à T_5 , alors S ne contient aucun sommet de T_5 , donc $S \subseteq R_1 \cup T_2 \cup Z \subseteq W_3'$. Si S contient a sommet de T_5 complet à T_2 , alors S ne contient aucun sommet de T_2 , donc $S \subseteq R_1 \cup T_5 \cup Z \subseteq T_2'$. Dans le cas restant, nous avons que $S \subseteq R_1 \cup T_2^{-T_5} \cup T_5^{-T_2} \cup Z = R_1 \cup U_1 \cup Z$. Si S contient un sommet de $Z \setminus Z_1$ (qui est complet à U_1 d'après la propriété 11 du lemme 3.11), alors nous avons $S \subseteq R_1 \cup Z \subseteq W_3'$. Sinon, nous avons $S \subseteq R_1 \cup U_1 \cup Z_1 = U_1'$.

Par symétrie, si S contient un sommet de T_3 , nous obtenons que S est inclus dans l'un des ensembles W_2'', T_3', W_2', U_4' .

Par conséquent, on peut supposer désormais que S ne contient aucun sommet de $T_2 \cup T_3$.

Posons que S contient un sommet de T_5 . Alors S ne contient pas de sommet de R_5 , donc $S \subseteq R_1 \cup R_2 \cup R_3 \cup R_4 \cup T_5 \cup Z$. Si S contient un sommet de R_2 , alors il ne contient pas de sommet de $R_1 \cup R_3 \cup T_5^Z$, et donc $S \subseteq R_2 \cup R_4 \cup T_5^{0Z} \cup Z \subseteq W_2'$. Par symétrie, si S contient un sommet de R_3 , alors $S \subseteq W_3'$. Enfin, si S ne contient aucun sommet de $R_2 \cup R_3$, alors $S \subseteq R_1 \cup R_4 \cup T_5 \cup Z \subseteq T_2'$. Ceci achève la preuve du lemme. \square

Ainsi, le lemme précédent montre que le calcul d'un stable de poids maximum de G peut se réduire au même problème sur un nombre fixé de graphes plus petits. De plus, le lemme suivant montre que chacun de ces problèmes est facile.

Lemme 3.17 *Pour chaque ensemble X parmi les neuf ensembles du lemme 3.16, le sous-graphe induit $G[X]$ est biparti et par conséquent, on peut calculer un stable de poids maximum de $G[X]$ en temps linéaire.*

Démonstration. Tout d'abord, posons $X = W_2'$. Alors X peut être partitionné en les deux ensembles $W_2 \cup R_2 \cup T_1 \cup T_3$ et $R_4 \cup T_5^{0Z} \cup Z$. Le premier est stable car tous ses sommets sont de couleur 2. Le second est aussi stable, par définition de Z et de T_5^{0Z} . Donc $G[X]$ est biparti, et trouver un stable maximum dans $G[X]$ peut se faire en temps linéaire, comme expliqué à la section 2.2.3. Le même argument peut être utilisé si $X = W_3'$, par symétrie. Pareillemet, W_2'' peut être partitionné en deux stables $W_2 \cup R_2 \cup T_1 \cup T_3$ et $R_5 \cup T_4^{0Z} \cup Z$, donc $G[W_2'']$ est biparti ; par symétrie il en va de même pour W_3'' . De même, T_2' peut être partitionné en

les deux ensembles $R_1 \cup R_4 \cup T_5$ et $T_2^{0Z} \cup Z$ donc $G[T_2']$ est biparti ; par symétrie, $G[T_3']$ est biparti aussi. Enfin, d'après la propriété 11 du lemme 3.11 et la définition de Z_1 , l'ensemble $U_1' = R_1 \cup T_5^{-T_2} \cup T_2^{-T_5} \cup Z_1$ peut être partitionné en les deux stables $R_1 \cup T_5^{-T_2}$ et $T_2^{-T_5} \cup Z_1$; il en va de même pour U_4' (par symétrie), et aussi pour U_5' . \square

Le lemme précédent montre que le problème du calcul d'un stable de poids maximum dans G peut se réduire à neuf instances du même problème sur des sous-graphes bipartis de G . Donc la complexité totale est linéaire. Ceci termine notre preuve du cas où G contient un C_5 .

Comme dans la section précédente, nous savons que si l'algorithme atteint ce point, il n'a pas trouvé de P_5 , établi avec succès que le graphe est 3-colorable, et résolu en temps linéaire le problème du stable de poids maximum. Mais ceci ne signifie pas que G ne contienne pas de P_5 . Ce dernier point est éclairci par les lemmes suivant. Auparavant nous introduisons de nouveaux ensembles : pour tout $i \in \{1, \dots, 5\}$, on définit :

$$\begin{aligned} X_i &= R_{i-2} \cup T_{i-1} \cup T_{i+1} \cup R_{i+2} \text{ et} \\ Y_i &= R_i \cup T_i \cup T_{i-2} \cup T_{i+2}. \end{aligned}$$

Lemme 3.18 G contient un P_5 si et seulement si le sous-graphe induit par l'un des quinze ensembles $W_2', W_2'', W_3', W_3'', X_1, \dots, X_5, Z \cup T_5, Y_1, \dots, Y_5$ contient un $2K_2$.

Démonstration. Supposons que l'un des quinze ensembles précités (appelons le X) contienne un $2K_2$, de sommets a, b, a', b' . Posons d'abord $X = W_2'$. Nous savons d'après le lemme 3.17 que X peut être partitionné en deux stables $W_2 \cup R_2 \cup T_1 \cup T_3$ et $R_4 \cup T_5^{0Z} \cup Z$. Nous savons également d'après la propriété 8 du lemme 3.11 qu'un sommet $a_1 \in R_1$ est complet à T_3 . Par conséquent a_1 est complet à $W_2 \cup R_2 \cup T_1 \cup T_3$ et anticomplet à $R_4 \cup T_5^{0Z} \cup Z$, et il s'ensuit que $\{a_1, a, b, a', b'\}$ induit un P_5 dans G . Posons à présent $X = W_2''$. On sait que X peut être partitionné en deux stables $W_2 \cup R_2 \cup T_1 \cup T_3$ et $R_5 \cup T_4^{0Z} \cup Z$. D'après la propriété 8 du lemme 3.11, il existe un sommet $b_3 \in R_3$ complet à T_1 , donc complet à $W_2 \cup R_2 \cup T_1 \cup T_3$ et anticomplet to $R_5 \cup T_4^{0Z} \cup Z$, et $\{b_3, a, b, a', b'\}$ induit un P_5 . Il en va de même si X désigne W_3' ou W_3'' par symétrie. Maintenant, posons $X = X_1$. Nous savons que X peut être partitionné en deux stables $T_2 \cup R_3$ et $R_4 \cup T_5$. D'après la propriété 8 du lemme 3.11, il y a un sommet $a_3 \in R_3$ complet à T_5 . Donc $a_3 \notin \{a, b, a', b'\}$ et $\{a_3, a, b, a', b'\}$ induit un P_5 . Par symétrie, il en va de même si X désigne X_i ($i \in \{2, \dots, 5\}$). À présent, posons $X = Z \cup T_5$. Alors on peut choisir un sommet $r_5 \in R_5$ et $\{r_5, a, b, a', b'\}$ induit un P_5 . Enfin, posons $X = Y_i$. D'après la propriété 8 du lemme 3.11, il existe un sommet $a_2 \in R_2$ complet à T_4 et un sommet b_5 complet à T_3 . Comme R_1 est complet à T_1 et T_3 complet à T_4 , soit il y a un sommet de $\{a, b, a', b'\}$ dans chaque ensemble $\{R_1, T_1, T_3, T_4\}$, soit il y en a deux dans l'un des ensembles R_1 ou T_1 , et deux dans l'un des ensembles T_3 ou T_4 . Dans chaque cas, $\{a_2, a, b, a', b'\}$ ou $\{b_5, a, b, a', b'\}$ induit un P_5 . Par symétrie, il en va de même si X est égal à l'un des Y_i ($i \in \{2, \dots, 5\}$).

Réciproquement, supposons que G contienne un P_5 $a-b-c-b'-a'$. Comme le sous-graphe induit par $R_1 \cup \dots \cup R_5 \cup Z$ ne contient pas de $2K_2$,

au moins l'un des quatre sommets a, b, a', b' doit se trouver dans $V_G \setminus (R_1 \cup \dots \cup R_5 \cup Z)$, c'est-à-dire dans $W_2 \cup W_3 \cup T_1 \cup \dots \cup T_5$. À symétrie près, ceci conduit aux cas suivants.

Cas 1 : a ou b est dans W_2 . Appelons w le sommet de $\{a, b\} \cap W_2$ et y l'autre sommet de $\{a, b\}$. Comme a' et b' ne sont pas adjacents à w , ils doivent, d'après la propriété 5 du lemme 3.11, se trouver tous les deux dans $R_2 \cup R_4 \cup R_5 \cup W_2 \cup T_1 \cup T_3 \cup Z$. Comme a' et b' sont adjacents, nous avons à examiner les sous-cas suivants.

Sous-cas 1.1 : l'un des sommets a', b' est dans Z et l'autre dans $W_2 \cup T_1 \cup T_3$. Appelons z le sommet de $\{a', b'\} \cap Z$ et t l'autre sommet de $\{a', b'\}$. Supposons que t soit dans $W_2 \cup T_1$. Donc t est dans $W_2 \cup T_1^Z$. Alors, puisque y est adjacent à w et pas à z ni t , d'après la propriété 4 ou 5 du lemme 3.11, $y \in R_4 \cup R_5, T_4 \cup Z$. En fait, si $y \in T_4$, les sommets y, t, z violent la propriété 5 ou 11 du lemme 3.11; si $y \in R_4 \cup Z$ alors $\{a, b, a', b'\}$ induit un $2K_2$ dans $R_4 \cup Z \cup W_2 \cup T_1$, qui est un sous-ensemble de W_2' ; et si $y \in R_5$ alors $\{a, b, a', b'\}$ induit un $2K_2$ dans $R_5 \cup Z \cup W_2 \cup T_1$, qui est un sous-ensemble de W_2'' . Maintenant, supposons que t soit dans T_3 , et donc plus précisément dans T_3^Z . Comme y est adjacent à w et pas à z ni t , y doit être dans $R_4 \cup T_5 \cup Z$ et $\{a, b, a', b'\}$ induit un $2K_2$ dans $R_4 \cup Z \cup W_2 \cup T_3$, qui est un sous-ensemble W_2' .

Sous-cas 1.2 : l'un des sommets a', b' est dans R_4 et l'autre dans R_5 . Comme y est adjacent à w et pas à a' ni b' , et d'après le lemme 3.11, y ne peut être dans l'un des ensembles de la C_5 -partition.

Sous-cas 1.3 : l'un des sommets a', b' est dans R_4 et l'autre dans $W_2 \cup T_1$. Alors, d'après la propriété 1 ou 5 du lemme 3.11, y doit être dans $R_4 \cup Z$, donc $\{a, b, a', b'\}$ induit un $2K_2$ dans $R_4 \cup Z \cup W_2 \cup T_1$, qui est un sous-ensemble de W_2' .

Sous-cas 1.4 : l'un des sommets a', b' est dans R_5 et l'autre dans $W_2 \cup T_3$. Alors, d'après la propriété 1 ou 5 du lemme 3.11, y doit être dans $R_5 \cup Z$, donc $\{a, b, a', b'\}$ induit un $2K_2$ dans $R_5 \cup Z \cup W_2 \cup T_3$, qui est un sous-ensemble de W_2' . Ceci termine la preuve du cas 1.

Par symétrie, si l'un des sommets a, b est dans W_3 , alors $\{a, b, a', b'\}$ induit un $2K_2$ dans W_3' ou W_3'' . On peut maintenant supposer qu'un sommet de $\{a, b, a', b'\}$ ne se trouve dans $W_2 \cup W_3$.

Cas 2 : a ou b est dans T_j ($j \in \{1, \dots, 5\}$). Pour simplifier les notations, posons $j = 1$, et appelons u le sommet de $\{a, b\} \cap T_1$ et y l'autre sommet de $\{a, b\}$. Comme a' et b' ne sont pas adjacents à u , ils doivent se trouver dans l'ensemble $R_2 \cup R_3 \cup R_4 \cup R_5 \cup T_1 \cup T_3 \cup T_4 \cup Z$. Comme a' et b' sont adjacents, d'après la propriété 1 du lemme 3.11, nous avons à examiner les sous-cas suivants :

Sous-cas 2.1 : l'un des sommets a', b' est dans Z . On appelle z ce sommet, et t l'autre sommet de $\{a', b'\}$. Donc $t \in T_1 \cup T_3 \cup T_4$. En fait, si $t \in T_3 \cup T_4$, alors u, t, z violent la propriété 11 du lemme 3.11. Par conséquent $t \in T_1$, et donc plus précisément dans T_1^Z . Comme y est adjacent à u et pas à t (qui sont tous deux dans T_1), $y \in Z$ (d'après la propriété 1 du lemme 3.11). Alors $\{a, b, a', b'\}$ induit un $2K_2$ dans $Z \cup T_1$, qui est un sous-ensemble de W_2' (si $j \in \{1, 2, 3, 4\}$, $Z \cup T_j$ est un sous-ensemble de W_2' ou W_3' ; lorsque $j = 5$, rappelons que $Z \cup T_5$ est l'un des quinze ensemble de ce lemme).

Sous-cas 2.2 : l'un des sommets a, b est dans T_1 . On appelle u' ce sommet, et y' l'autre sommet de $\{a', b'\}$. On peut supposer que $y, y' \notin Z$ (sinon nous sommes dans le sous-cas précédent). Par conséquent, y et y' sont soit tous les deux dans $R_3 \cup R_4$, soit tous les deux dans $R_4 \cup T_3$, et donc $\{a, b, a', b'\}$ induit un $2K_2$ (soit dans $R_3 \cup T_4 \cup T_1 \subseteq X_5$, soit dans $R_4 \cup T_3 \cup T_3 \subseteq X_2$).

Sous-cas 2.3 : l'un des sommets a', b' est dans R_3 . On appelle u' ce sommet et y' l'autre sommet de $\{a', b'\}$. Alors $y' \in R_2 \cup T_3 \cup R_4$. En fait, d'après la propriété 3 du lemme 3.11, $y' \notin R_4$. Supposons qu'il soit dans R_2 ; alors d'après la propriété 1 du lemme 3.11, $y \in T_4$ et alors $\{a, b, a', b'\}$ induit un $2K_2$ dans $T_1 \cup R_2 \cup R_3 \cup T_4 = X_5$. Supposons désormais que $y' \in T_3$; alors $y \in R_1 \cup T_5$. Si $y \in R_1$, alors $u \in T_1^{-R_3}$ et $y' \in T_3^{-R_1}$ donc ils violent la propriété 9 du lemme 3.11. Si $y \in T_5$ alors $\{a, b, a', b'\}$ induit un $2K_2$ dans $R_3 \cup T_3 \cup T_1 \cup T_5 = Y_3$. Le sous-cas où l'un des sommets a', b' appartient à R_4 est symétrique, et les cas où $j \in \{2, \dots, 5\}$ sont similaires. La preuve du lemme est ainsi complète. \square

Le lemme précédent montre que chercher un P_5 dans G peut se réduire à chercher un $2K_2$ dans quinze sous-graphes induits de G . Le lemme suivant montre que ces quinze problèmes sont faciles à résoudre.

Lemme 3.19 *Pour chaque ensemble X parmi les quinze ensembles du lemme 3.18, on peut décider en temps linéaire si $G[X]$ contient un $2K_2$.*

Démonstration. Si X est un des ensembles du lemme 3.18 autre que Y_1, \dots, Y_5 , nous avons déjà observé que $G[X]$ est biparti, et donc on peut détecter un $2K_2$ dans $G[X]$ en temps linéaire comme expliqué à la section 2.2.3.

Supposons à présent que $X = Y_i$ pour $i \in \{1, \dots, 5\}$. Dans ce cas, le sous-graphe $G[Y_i]$ n'est pas nécessairement biparti, donc nous devons procéder différemment. Rappelons que R_i est complet à T_i et T_{i-2} et complet à T_{i+2} . Soit (A, B, C, D) une permutation des quatre ensembles $R_i, T_i, T_{i-2}, T_{i+2}$; on dit que cette permutation est *acceptable* si A est complet à B et C est complet à D . Pour chaque couleur $j \in \{1, 2, 3\}$, soit A_j l'ensemble des sommets de couleur j dans A , les ensembles B_j, C_j, D_j étant définis de manière similaire. Rappelons également que dans notre 3-coloration γ , chacun des ensembles $R_1, \dots, R_4, T_1, \dots, T_5$ a seulement une couleur, et R_5 peut avoir des sommets de couleur 2 ou 3 seulement. Nous affirmons que :

$G[Y_i]$ contient un $2K_2$ si et seulement si l'une des propriétés suivantes est vérifiée pour une permutation acceptable (A, B, C, D) et une couleur $j \in \{1, 2, 3\}$:

(i) $G[A_j \cup D]$ ou $G[C_j \cup B]$ contient un $2K_2$; (ii) B^{-C_j} n'est pas complet à D^{-A_j} .

Prouvons cette affirmation. Clairement, si (i) est vérifiée, alors $G[Y_i]$ contient un $2K_2$. Maintenant, supposons que (ii) soit vérifiée. Prenons des sommets non adjacents b, d avec $b \in B^{-C_j}$ et $d \in D^{-A_j}$. Donc d a un non-voisin $a_j \in A_j$ et de manière similaire b a un non-voisin $c_j \in C_j$. Alors $\{a_j, b, c_j, d\}$ induit un $2K_2$ dans $G[Y_i]$.

Réciproquement, supposons que $G[Y_i]$ contienne un $2K_2$ de sommets u, v, x, y et d'arêtes uv, xy . Comme G est 3-coloré, on peut supposer que

u, v ont la même couleur j . Supposons que u, v soient dans le même ensemble A parmi les ensembles $R_i, T_i, T_{i-2}, T_{i+2}$. Alors x, y ne sont pas dans A , et pas non plus dans l'ensemble complet à A , et puisqu'ils sont adjacents, ils doivent être tous les deux dans l'un des deux ensembles restants. Donc il existe une permutation acceptable (A, B, C, D) telle que u, v sont dans A_j et x, y sont dans D , et (i) est vérifiée. Nous pouvons à présent supposer que u, v ne sont pas dans le même ensemble parmi $R_i, T_i, T_{i-2}, T_{i+2}$, donc il existe une permutation acceptable (A, B, C, D) telle que $u \in A$ et $v \in C$. Clairement, $x \notin A \cup D$ et $y \notin B \cup C$. Si $x \in C$, alors $y \in A$ mais alors A et C ont deux couleurs, ce qui est impossible (seul R_5 peut posséder deux couleurs). Par conséquent, on peut supposer que $x \in B$ et $y \in D$. Donc (ii) est vérifiée pour (A, B, C, D) , et l'affirmation précédente est prouvée.

D'après cette affirmation, tester si $G[Y_i]$ contient un $2K_2$ est équivalent à tester si (i) ou (ii) est vérifiée pour toute permutation acceptable et toute couleur j . Comme chacun des sous-graphes mentionnés dans (i) et (ii) est biparti, on peut utiliser l'algorithme linéaire de la section 2.2.3. L'argument ci-dessus implique qu'il y a huit permutations à tester et trois couleurs, donc le temps total de la procédure reste linéaire. En fait, étant donné notre 3-coloration, il est facile de vérifier que seulement six permutations ont à être testées : (T_1, R_1, T_3, T_4) avec $j = 2$, (T_2, R_2, T_4, T_5) avec $j = 3$, (T_3, R_3, T_1, T_5) avec $j = 2$, (T_4, R_4, T_2, T_1) avec $j = 3$, (R_5, T_5, T_2, T_3) avec $j = 3$ et (R_5, T_5, T_3, T_2) avec $j = 2$. \square

3.2.6 Cas où G contient un \overline{C}_6

Les preuves et techniques de cette section seront assez similaires à celles de la section 3.2.4. Soient $a_1, a_2, a_3, b_1, b_2, b_3$ six sommets qui induisent un \overline{C}_6 , avec pour arêtes $a_1a_2, a_1a_3, a_2a_3, b_1b_2, b_1b_3, b_2b_3, a_1b_1, a_2b_2, a_3b_3$. Nous dirons qu'un sextuplet $(A_1, A_2, A_3, B_1, B_2, B_3)$ est une \overline{C}_6 -structure dans G si $A_1, A_2, A_3, B_1, B_2, B_3$ sont des sous-ensembles non-vides et disjoints de V_G tels que (les indices étant compris modulo 3) :

- les six ensembles $(A_1, A_2, A_3, B_1, B_2, B_3)$ sont stables,
- les ensembles A_1, A_2, A_3 sont complets deux à deux, et les ensembles B_1, B_2, B_3 sont complets deux à deux,
- pour chaque $i \in \{1, 2, 3\}$, A_i est complet à B_i et anticomplet B_{i+2} ,
- pour chaque $i \in \{1, 2, 3\}$, chaque sommet de A_i a un non-voisin dans B_{i+1} et chaque sommet de B_{i+1} a un non-voisin dans A_i .

Notons que les six sommets a_1, \dots, b_3 induisent un graphe avec une \overline{C}_6 -structure $(\{a_1\}, \{a_2\}, \{a_3\}, \{b_1\}, \{b_2\}, \{b_3\})$. Notre but est de faire grossir cette structure jusqu'à la plus grande possible.

Soit H un sous-graphe induit de G avec une \overline{C}_6 -structure $(A_1, A_2, A_3, B_1, B_2, B_3)$ et x un sommet de $V_G \setminus V_H$. Nous disons que x peut être ajouté à cette \overline{C}_6 -structure si le sous-graphe induit par $V_H \cup \{x\}$ a une \overline{C}_6 -structure obtenue en plaçant x dans l'un des six ensembles.

Lemme 3.20 Soient H un sous-graphe induit de G avec une \overline{C}_6 -structure $(A_1, A_2, A_3, B_1, B_2, B_3)$, et x un sommet de $V_G \setminus V_H$ qui a un voisin dans V_H . Alors l'une des propriétés suivantes est vérifiée :

- G contient un P_5 , un C_5 ou un membre de \mathcal{F} ,
- le sous-graphe induit par $V_H \cup \{x\}$ a une \overline{C}_6 -structure,
- il existe un entier $h \in \{1, 2, 3\}$ tel que x est complet à $A_{h+1} \cup A_{h+2} \cup B_h \cup B_{h+1}$ et anticomplet à $A_h \cup B_{h+2}$.

Démonstration. La preuve est un algorithme qui détermine quelle propriété est vérifiée. En utilisant les techniques vues précédemment, il est facile de voir que cet algorithme s'exécute en $\mathcal{O}(n + m)$. Si l'algorithme trouve un P_5 ou un membre de \mathcal{F} , il s'arrête avec le message « G n'est pas dans la classe \mathcal{C} ». S'il trouve un C_5 , alors il va à l'étape précédente (section 3.2.5). L'algorithme fonctionne comme suit. Soit s_A le nombre d'ensembles parmi A_1, A_2, A_3 qui contient un voisin de x et s_B défini de manière similaire. Si $s_A = 3$ alors x a un voisin $u_i \in A_i$ pour chaque $i \in \{1, 2, 3\}$ et donc $\{x, u_1, u_2, u_3\}$ induit un K_4 (qui, pour rappel, est un membre de \mathcal{F}), donc l'algorithme retourne ce K_4 et s'arrête. Supposons à présent que $s_A \leq 2$ et $s_B \leq 2$.

Nous allons tout d'abord montré, par l'absurde, que $\max_{s_A, s_B} = 2$. Supposons en effet, au contraire, que $\max_{s_A, s_B} \leq 1$. Par hypothèse sur x , et à symétrie près, on peut supposer que x a un voisin $u_1 \in A_1$, donc $s_A = 1$ et x est anticomplet à $A_2 \cup A_3$. Choisissons des sommets $u_2 \in A_2$ et $u_3 \in A_3$. Par définition d'une \overline{C}_6 -structure, pour chaque $i \in \{1, 2, 3\}$ le sommet u_i a un non-voisin $v_{i+1} \in B_{i+1}$. Notons que $\{u_1, u_2, u_3, v_1, v_2, v_3\}$ induit un \overline{C}_6 . Si x n'est pas adjacent à v_2 alors $\{x, u_1, u_2, v_2, v_3\}$ induit soit un P_5 soit un C_5 . Donc supposons que x soit adjacent à v_2 ; s'il n'est pas adjacent à v_3 alors $\{x, u_1, u_3, v_2, v_3\}$ induit un C_5 ; et s'il est adjacent à v_3 , $s_B = 2$, ce qui contredit notre hypothèse de départ.

Comme $\max_{s_A, s_B} = 2$ et à symétrie près, on peut supposer que $s_A = 2$. Donc, à symétrie près, x a un voisin $u_i \in A_i$ pour chaque $i \in \{1, 2\}$ et est anticomplet à A_3 . Prenons un sommet $u_3 \in A_3$. On sait que, pour chaque $i \in \{1, 2, 3\}$, u_i a un non-voisin $v_{i+1} \in B_{i+1}$. Notons que $\{u_1, u_2, u_3, v_1, v_2, v_3\}$ induit un \overline{C}_6 . Si x a un non-voisin $w \in B_3$ alors $\{x, u_1, u_3, w, v_2\}$ induit un P_5 ou un C_5 . Donc nous pouvons supposer que x est complet à B_3 . Nous distinguons les trois cas suivants.

Cas 1 : x a un voisin dans B_1 . Alors l'argument ci-dessus concernant le couple (A_1, A_2) peut être répété avec le couple (B_3, B_1) , donc nous pouvons supposer que x est anticomplet à B_2 et complet à A_2 . Si x est complet à $A_1 \cup B_1$, alors la dernière propriété du lemme est vérifiée (avec $h = 3$). Si x est complet à A_1 et pas à B_1 , x peut être ajouté à A_3 . Si x est complet à B_1 et pas à A_1 , alors x peut être ajouté à B_2 . Enfin, si x a un non-voisin $u'_1 \in A_1$ et un non-voisin $v'_1 \in B_1$ alors $\{u'_1, u_2, x, v_3, v'_1\}$ induit un C_5 .

Cas 2 : x a un voisin dans B_2 . Alors l'argument ci-dessus concernant le couple (A_1, A_2) peut être répété avec le couple (B_2, B_3) , donc nous pouvons supposer que x est anticomplet à B_1 et complet à A_1 . La situation n'est toutefois pas symétrique de celle du cas 1, donc l'analysons diffé-

remment. Soit z_2 un voisin de $x \in B_2$. Nous prétendons que :

$$\text{Pour tout } j \in \{1, 2, 3\}, A_j \text{ est anticomplet à } B_{j+1}. \quad (3.1)$$

Supposons au contraire qu'il existe, pour un certain $j \in \{1, 2, 3\}$, une arête $\alpha_j \beta_{j+1}$ avec $\alpha_j \in A_j$ et $\beta_{j+1} \in B_{j+1}$. Posons tout d'abord $j = 1$. Si $x \sim \beta_2$, alors $\{x, \alpha_1, u_2, \beta_2\}$ induit un K_4 ; si $x \not\sim \beta_2$, alors $\beta_2 \neq v_2$ et par le même argument $\alpha_1 \not\sim v_2$ et donc $\{x, v_1, u_2, v_3, \alpha_1, v_2, \beta_2\}$ induit un \overline{C}_7 ($= F_9$). Posons maintenant $j = 2$. Si $x \sim \alpha_2$, alors $\{x, z_2, \alpha_2, \beta_3\}$ induit un K_4 ; si $x \not\sim \alpha_2$, alors $\alpha_2 \neq u_2$ et par le même argument $\beta_3 \not\sim u_2$, et donc $\{x, \alpha_2, u_2, \beta_3, u_1, z_2, u_3\}$ induit un \overline{C}_7 . Posons enfin $j = 3$. Alors $\{x, \alpha_3, z_2, u_1, v_3, u_2, \beta_1\}$ induit un \overline{C}_7 . Donc (3.1) est vraie.

D'après (3.1), $(A_1, A_3, A_2, B_1, B_3, B_2)$ est aussi une \overline{C}_6 . À présent la situation est entièrement similaire à celle du cas 1, et on peut conclure de la même façon.

Cas 3 : x n'a aucun voisin dans $B_1 \cup B_2$. Donc x est anticomplet à $A_3 \cup B_1 \cup B_2$ et complet à B_3 . Si x a un non-voisin $u'_1 \in A_1$, alors $\{x, u_2, u'_1, v_1, v_3\}$ induit un C_5 ; si x est complet à A_1 alors il est aussi complet à A_2 (sinon, en notant $u'_2 \in A_2$ un non-voisin de x , on sait que u'_2 a un non-voisin $v'_3 \in B_3$ et alors $\{x, u_1, u'_2, v_2, v'_3\}$ induit un C_5). En conclusion, x est complet à $A_1 \cup A_2 \cup B_3$ donc x peut être ajouté à A_3 . \square

D'un point de vue algorithmique, il est facile de tester les propriétés du lemme précédent. Initialement, nous posons $A_i = \{a_i\}$ et $B_i = \{b_i\}$ pour tout $i \in \{1, 2, 3\}$. Ensuite nous appliquons les remarques de la section 3.2.1. Le temps total d'exécution de cette partie de l'algorithme est donc $\mathcal{O}(n + m)$.

À présent, nous pouvons supposer que l'algorithme a produit un sous-graphe induit H de G qui possède une \overline{C}_6 -structure $(A_1, A_2, A_3, B_1, B_2, B_3)$ avec $a_i \in A_i$ et $b_i \in B_i$ pour tout $i \in \{1, 2, 3\}$ et la propriété que H est maximal dans le sens où aucun sommet ne peut plus être ajouté à cette structure. Le lemme précédent implique donc que, pour chaque sommet de $V_G \setminus V_H$, soit x est anticomplet à V_H soit il satisfait la dernière propriété du lemme. Pour tout $i \in \{1, 2, 3\}$ soit X_i l'ensemble des sommets de $V_G \setminus V_H$ complets à $A_{i+1} \cup A_{i+2} \cup B_i \cup B_{i+1}$ et anticomplets à $A_i \cup B_{i+2}$. Soit $Z = V_G \setminus (V_H \cup X_1 \cup X_2 \cup X_3)$. Donc $A_1, A_2, A_3, B_1, B_2, B_3, X_1, X_2, X_3, Z$ forment une partition de V_G et Z est anticomplet à V_H . Si Z contient deux sommets adjacents z et z' alors $\{z, z', a_1, b_1\}$ induit un $2K_2$ et l'algorithme va à la section 3.2.4. Ainsi, nous supposons dans la suite que Z est stable.

Lemme 3.21 G contient un P_5 ou un K_4 , ou les propriétés suivantes sont vérifiées :

- les ensembles X_1, X_2, X_3 sont stables et complets deux à deux,
- tout sommet de Z est anticomplet à au moins un des ensembles X_1, X_2, X_3 ,
- $V_G \setminus Z$ ne contient pas de sommet double,
- tout sommet double de Z est anticomplet à au moins deux des ensembles X_1, X_2, X_3 .

Démonstration. Supposons, à symétrie près, que X_1 contienne deux sommets adjacents x_1, x'_1 ; alors $\{x_1, x'_1, a, b\}$ induit un K_4 . Il en va de même pour X_2 et X_3 .

Ensuite, supposons, à symétrie près, qu'il existe deux sommets non adjacents $x_1 \in X_1$ et $x_2 \in X_2$. Alors $\{x_1, a_2, a_1, x_2, b_3\}$ induit un P_5 .

Supposons maintenant qu'il existe un sommet double u dans $V_G \setminus Z$. À symétrie près, on peut supposer que $u \in A_1 \cup X_1$. Alors $\{u, \bar{u}, a_2, b_2\}$ induit un K_4 .

Enfin, supposons qu'un sommet double $z \in Z$ ait un voisin $x_h \in X_h$ pour tout $h \in \{1, 2\}$. Alors $\{z, x_1, x_2, x_3\}$ induit un K_4 . \square

Toujours avec les techniques habituelles, l'algorithme peut tester ces propriétés en temps $\mathcal{O}(m + n)$. Donc supposons qu'elles soient vérifiées. Donc G est 3-colorable, car on peut affecter la couleur i aux sommets de $A_i \cup B_{i+2}$ pour tout $i \in \{1, 2, 3\}$, affecter à chaque sommet simple $z \in Z$ la couleur affectée à n'importe quel ensemble X_h qui ne contient aucun voisin de z , et affecter à chaque sommet double $z \in Z$ les deux couleurs affectées à deux ensembles parmi X_1, X_2, X_3 qui ne contiennent aucun voisin de z .

Lemme 3.22 *Chaque stable de G est un sous-ensemble de l'un des six ensembles suivants :*

$$A_i \cup B_{i+1} \cup B_{i+2} \cup X_i \cup Z, i \in \{1, 2, 3\},$$

$$B_i \cup A_{i+1} \cup A_{i+2} \cup X_{i-2} \cup Z, i \in \{1, 2, 3\}.$$

De plus, chacun de ces six ensembles induit un sous-graphe biparti.

Démonstration. Soit S un stable de G . Supposons tout d'abord, à symétrie près, que S contienne un sommet de A_1 . Alors la définition d'une \bar{C}_6 -structure et des X_i implique que S ne contient pas de sommet de $A_2 \cup A_3 \cup B_1 \cup X_2 \cup X_3$. Donc S est un sous-ensemble de $A_1 \cup B_2 \cup B_3 \cup X_1 \cup Z$, qui est l'un des six ensembles du lemme. De plus, cet ensemble induit un graphe biparti, parce qu'il peut être partitionné en deux stables $A_1 \cup B_3 \cup X$ et $B_2 \cup Z$. De même, si S contient un sommet de A_2, \dots, B_3 , il peut être inclus dans l'un des cinq autres ensembles du lemme, qui est biparti par un argument similaire. Enfin, supposons que S ne contienne pas de sommet de V_H . Alors le lemme précédent implique que S est un sous-ensemble de $X_h \cup Z$ pour un $h \in \{1, 2, 3\}$, qui est un sous-ensemble de l'un des six ensembles du lemme. \square

Le lemme précédent implique que pour résoudre MWIS dans G , il suffit de le résoudre pour les six graphes bipartis mentionnés et de retourner la meilleure solution. D'après les résultats de la section 2.2.3, ceci peut se faire en temps linéaire.

Si l'algorithme atteint ce point sans avoir trouvé de P_5 , cela ne signifie pas que G soit sans P_5 pour autant. Le lemme suivant éclaircit ce point.

Lemme 3.23 G est sans P_5 si et seulement si chacun des trois sous-graphes $G[A_i \cup B_{i+1} \cup X_i \cup Z]$, $i \in \{1, 2, 3\}$ est sans $2K_2$. De plus, chacun de ces trois sous-graphes est biparti.

Démonstration. La deuxième affirmation du lemme est évidente, puisque $A_i \cup X_i$ et $B_{i+1} \cup Z$ sont des stables.

Prouvons la première affirmation. Pour cela, supposons, à symétrie près, que $G[A_1 \cup B_2 \cup X_1 \cup Z]$ contienne un $2K_2$, de sommets a, b, c, d et d'arêtes ab et cd . Comme $A_1 \cup X_1$ et $B_2 \cup Z$ sont des stables, on peut supposer que $a, c \in A_1 \cup X_1$ et $b, d \in B_2 \cup Z$. Alors $b-a-a_3-c-d$ est un P_5 induit dans G , quel que soit $a_3 \in A_3$.

Réciproquement, supposons qu'aucun des trois sous-graphes ne contienne de $2K_2$. Pour tout $i \in \{1, 2, 3\}$, la définition des X_i et le lemme 3.21 impliquent que chaque sommet de $V_G \setminus Z$ est soit complet soit anticomplet à X_i . Autrement dit, tout sommet qui a un voisin et un non-voisin dans X_i appartient à Z . Posons $X = X_1 \cup X_2 \cup X_3$, et supposons que G contienne un P_5 $u_1-u_2-u_3-u_4-u_5$ avec u_1 ou $u_2 \in X$; appelons x ce sommet, disons $x \in X_1$. Si l'un des sommets u_4 ou u_5 appartient aussi à X , alors d'après ce qui précède les deux autres sommets parmi u_1, u_2, u_4, u_5 doivent être dans Z mais alors ces quatre sommets induisent un $2K_2$ dans $G[X_1 \cup Z]$, ce qui contredit l'hypothèse. D'autre part, si u_4 et u_5 ne sont pas dans X , alors comme ils ne sont pas adjacents à x , ils doivent être $A_1 \cup B_3 \cup Z$, ce qui est impossible puisque cet ensemble est stable. Ainsi, par symétrie, on peut supposer qu'aucun des sommets u_1, u_2, u_4, u_5 n'est dans X . Comme les sommets de Z sont isolés dans $G \setminus X$, les sommets u_1, u_2, u_4, u_5 doivent être dans H . Supposons que $u_3 \in X_1$. Alors u_1 et u_5 doivent être dans $A_1 \cup B_3$. Si $u_1 \in A_1$ et $u_5 \in B_3$ (ou vice versa), alors il est facile de voir que l'on ne peut pas placer u_2 et u_4 de sorte à respecter les adjacences du P_5 . Donc, à symétrie près, supposons que u_1, u_5 soient tous les deux dans A_1 . Alors u_2 et u_4 ne sont ni complets ni anticomplets à A_1 , donc ils doivent être dans B_2 , mais alors $\{u_1, u_2, u_4, u_5\}$ induit un $2K_2$ dans $G[A_1 \cup B_2]$, ce qui contredit l'hypothèse de départ. Ainsi, $u_3 \in V_H$, disons $u_3 \in A_3$; donc u_1 et u_5 sont dans $B_1 \cup B_2$. S'ils sont tous les deux dans B_1 , alors il est impossible de placer u_2 et u_4 de sorte à respecter les adjacences du P_5 . Enfin, si u_2 et u_4 sont tous les deux dans B_2 , alors u_2 et u_4 doivent être dans A_1 , mais alors $\{u_1, u_2, u_4, u_5\}$ induit un $2K_2$ dans $G[A_1 \cup B_2]$, ce qui contredit à nouveau l'hypothèse de départ. \square

CONCLUSION DU CHAPITRE

Nous avons étudié dans ce chapitre la classe des graphes sans P_5 3-colorables, et obtenu une description complète de leur structure. Nous avons ainsi pu montrer que cette structure peut toujours être obtenue en temps linéaire. De plus, en utilisant les propriétés de cette structure, nous avons pu prouver que le problème du stable de poids maximum d'un graphe sans P_5 3-colorable se réduit à le résoudre dans un nombre fixe (égal au plus à 9) de sous-graphes bipartis de G , dans lesquels on peut facilement obtenir une solution. Nous avons également montré que l'on

peut en déduire en temps linéaire des algorithmes de reconnaissance et de coloration de ces graphes.

Les conséquences de ces résultats sont importantes : d'une part nous avons répondu par l'affirmative à une question laissée ouverte par une autre équipe de chercheurs, et d'autre part nos résultats impliquent, par effet domino, que l'on a abaissé d'une unité la complexité de l'algorithme du calcul du stable maximum pour toutes les classes de graphes sans P_5 k -colorables, avec $k > 3$.

Enfin, comme perspective de ces travaux, il est légitime de se demander s'il est possible d'obtenir facilement une telle structure pour les graphes sans P_5 k -colorables pour tout k fixé, et s'il est possible d'en déduire un algorithme polynomial pour résoudre le problème du stable maximum.

CARACTÉRISATION DES GRAPHS CHAIN-PROBE

4

SOMMAIRE

4.1	PROBLÈMES DE TYPE <i>sandwich</i> ET PROBLÈMES DE TYPE <i>probe</i> . . .	153
4.1.1	Problèmes de type <i>sandwich</i>	153
4.1.2	Problèmes de type <i>probe</i>	154
4.2	CARACTÉRISATION DES GRAPHS CHAIN-PROBE	155
4.2.1	Les six graphes minimaux interdits	155
4.2.2	Preuve du théorème 4.1	157
4.2.3	Considérations algorithmiques	162
	CONCLUSION	163

LORSQUE l'on s'intéresse à une classe de graphes \mathcal{C} , il arrive que bien qu'un graphe donné n'appartienne pas à \mathcal{C} , il en soit néanmoins « proche ». On se pose alors la question de savoir s'il est possible de rajouter des arêtes dans ce graphe pour le transformer en un graphe de \mathcal{C} . Ceci donne lieu aux problèmes de type *sandwich* et aux problèmes de type *probe*.

Nous commencerons par présenter brièvement ces problèmes. Dans un second temps, nous nous intéresserons au problème de la caractérisation des graphes de type *chain-probe*, c'est-à-dire les graphes dans lesquels on peut rajouter des arêtes entre des sommets d'un stable de façon à obtenir un graphe de chaîne (c'est-à-dire, pour rappel, un graphe biparti sans $2K_2$). Ce problème a été suggéré par Martin C. Golumbic alors qu'il était en visite au laboratoire G-SCOP en juillet 2007. Nous verrons que nous avons été en mesure de répondre à cette question et d'élaborer un algorithme de reconnaissance quadratique des graphes *chain-probe*. Ces travaux ont fait l'objet d'un article paru dans *Annals of Operations Research* [170] et ont été présentés aux Journées Graphes et Algorithmes à Sophia-Antipolis en novembre 2008.

4.1 PROBLÈMES DE TYPE *sandwich* ET PROBLÈMES DE TYPE *probe*

4.1.1 Problèmes de type *sandwich*

Lorsque l'on s'intéresse au problème de la reconnaissance d'une classe de graphes, c'est le plus souvent parce que cette classe possède certaines propriétés qui rendent plus facile la résolution d'autres problèmes habituellement difficiles dans les graphes en général. Ainsi, tel que nous l'avons vu au chapitre précédent, savoir si un graphe donné est sans P_5 et 3-colorable permet, dans l'affirmative, de calculer un stable de poids maximum en temps linéaire.

Or, en pratique, il arrive que bien que le graphe que l'on donne en entrée d'un algorithme de reconnaissance n'appartienne pas à la classe désirée, il en soit néanmoins « proche » selon certains critères ; on souhaite donc parfois « relâcher » certaines conditions d'acceptation d'un graphe. Plusieurs types de relaxations ont été étudiés, telle que la **complétion** :

PROBLÈME DE LA COMPLÉTION POUR UNE CLASSE \mathcal{C}	
(\mathcal{C} -COMPLETION PROBLEM) (ou \mathcal{C} -FILL-IN PROBLEM)	
INSTANCE :	Un graphe $G = (V, E)$ et un entier k
PROBLÈME :	Est-il possible d'ajouter au plus k arêtes à G pour que le graphe obtenu appartienne à \mathcal{C} ?

Il est connu que ce problème est \mathcal{NP} -complet si \mathcal{C} est par exemple la classe des graphes hamiltoniens ou la classe des graphes d'intervalles [150, problèmes GT34 et GT35], ou des graphes cordaux [348].

Une variante, dans laquelle on impose l'ensemble d'arêtes optionnelles, donne lieu aux problèmes de type **sandwich** : il s'agit toujours d'ajouter des arêtes à un graphe G_1 mais de sorte que le graphe obtenu G soit un graphe partiel d'un graphe G_2 (autrement dit, G est « en sandwich » entre G_1 et G_2) :

PROBLÈME DU SANDWICH POUR UNE CLASSE \mathcal{C}	
(\mathcal{C} -SANDWICH PROBLEM)	
INSTANCE :	Un graphe $G_1 = (V, E_1)$ et un graphe $G_2 = (V, E_2)$ tels que $E_1 \subset E_2$
PROBLÈME :	Existe-t-il un graphe $G = (V, E)$ avec $E_1 \subseteq E \subseteq E_2$ tel que G appartienne à \mathcal{C} ?

Remarquons que le problème de la reconnaissance des graphes de la classe \mathcal{C} est équivalent au problème du sandwich particulier où $E_1 = E_2$ (*i.e.* l'ensemble d'arêtes optionnelles est vide). Les problèmes de type sandwich ont été étudiés notamment dans [171], où l'on trouve la preuve que le problème est \mathcal{NP} -complet si \mathcal{C} est la classe des graphes d'intervalles, et [168] dans lequel les auteurs donnent des algorithmes polynomiaux pour les problèmes du sandwich pour les cographes et les graphes à seuil, des preuves de \mathcal{NP} -complétudes, en particulier pour les graphes de permutation, les graphes de comparabilité et les graphes triangulés, ainsi que de nombreuses applications très intéressantes en génétique, en intelligence

artificielle, en informatique ou encore en algèbre. Des recherches récentes ont permis également de montrer que le problème est polynomial si \mathcal{C} est la classe des graphes P_4 -épars [107], et \mathcal{NP} -complet s'il s'agit de la classe des graphes de chaîne (mais à une solution linéaire si la bipartition est fixée à l'avance) [106].

4.1.2 Problèmes de type *probe*

Étant donnée une classe de graphes \mathcal{C} , on dit qu'un graphe $G = (V, E)$ est \mathcal{C} -probe si V peut être partitionné en deux ensembles, P (les sommets *probe*) et un ensemble stable N (les sommets *non-probe*), et si l'on peut ajouter des arêtes entre des sommets de N uniquement de sorte que le graphe résultant appartienne à \mathcal{C} . On distingue deux versions du problème de reconnaissance des graphes \mathcal{C} -probe, selon que la partition de V est imposée ou non :

PROBLÈME PROBE POUR UNE CLASSE \mathcal{C} (VERSION « PARTITIONNÉE »)
(PARTITIONED \mathcal{C} -PROBE PROBLEM)

INSTANCE : Un graphe $G = (V, E)$ avec $V = P \cup N$ (où N est stable)

PROBLÈME : Existe-t-il un sous-ensemble $F \subseteq N \times N$ tels que $G' = (V, E \cup F)$ appartienne à \mathcal{C} ?

PROBLÈME PROBE POUR UNE CLASSE \mathcal{C} (VERSION « NON PARTITIONNÉE »)
(NON-PARTITIONED \mathcal{C} -PROBE PROBLEM)

INSTANCE : Un graphe $G = (V, E)$

PROBLÈME : Existe-t-il une partition de V en $P \cup N$ (où N est un stable) et un sous-ensemble $F \subseteq N \times N$ tels que $G' = (V, E \cup F)$ appartienne à \mathcal{C} ?

Remarquons que le premier problème correspond au cas particulier du problème du sandwich où $E_2 \setminus E_1 = N \times N$, et est évidemment aussi un cas particulier du second problème. Notons également qu'un graphe qui vérifie la version non partitionnée n'est pas forcément solution de la version partitionnée, selon la partition imposée.

Les graphes *probe* ont été introduits par Zhang [351] qui a étudié le cas où \mathcal{C} est la classe des graphes d'intervalles, afin de modéliser un problème de biologie moléculaire, dans le cadre du projet de séquençage du génome humain. Dans ce contexte, une *sonde génétique* est un fragment d'ADN ou d'ARN utilisé pour détecter la présence de séquences de nucléotides complémentaires de la sonde ; cette sonde est appelée *hybridization probe* en anglais, d'où le nom donné à ces graphes, puis aux problèmes de ce type¹. Par la suite, les graphes d'intervalles « probe » ont fait l'objet de nombreuses recherches [172, 275, 314]. En particulier, Johnson et Spinrad [223] ont montré que l'on peut reconnaître les graphes d'intervalles probe partitionnés en $\mathcal{O}(n^2)$ en utilisant une structure de données très commune appelée *PQ-arbre* combinée à la décomposition

1. Il nous est arrivé de lire *graphes de sonde* en français, mais cet usage semble encore marginal.

modulaire ; McConnell et Spinrad [274] ont ensuite donné un algorithme de complexité $\mathcal{O}(n + m \log n)$. Si la partition $V = P \cup N$ n'est pas donnée, il est également possible de reconnaître les graphes d'intervalles probe en temps polynomial [72]. Les graphes d'intervalles probe ont rapidement été généralisés aux graphes cordaux probe par Golumbic et Lipshteyn [26, 27, 164] ; notamment, on trouvera dans [169] des preuves que l'on peut reconnaître ces graphes en $\mathcal{O}(m|P|)$ si la partition de V est donnée, et en $\mathcal{O}(m^2)$ sinon. D'autres classes ont été étudiées, comme les graphes à seuils probe [109] (reconnaisables en temps linéaire), les cographes probe [241] (eux aussi reconnaissables en temps linéaire), les graphes de permutation probe [69, 70, 71] (que l'on peut reconnaître en $\mathcal{O}(n^2)$ si la partition de V est fournie ; dans le cas contraire, la complexité est à notre connaissance encore ouverte) et bien d'autres encore².

Dans la section suivante, **nous donnons une caractérisation des graphes chain-probe** à partir de laquelle nous pouvons résoudre facilement le problème d'ajout des arêtes manquantes, ainsi qu'un **algorithme polynomial pour les reconnaître**. Ceci est bien sûr à mettre en contraste avec le résultat mentionné précédemment qui dit que le problème du sandwich est \mathcal{NP} -complet si \mathcal{C} est la classe des graphes de chaîne.

4.2 CARACTÉRISATION DES GRAPHES CHAIN-PROBE

Lors d'une invitation à Grenoble en juillet 2007, Martin C. Golumbic a suggéré d'étudier la classe des graphes chain-probe, afin de caractériser ces graphes et d'en déduire un algorithme de reconnaissance polynomial. Rappelons qu'un graphe de chaîne est un graphe biparti connexe sans P_5 , ou de manière équivalente, un graphe biparti sans $2K_2$ (cf. section 2.2.3).

Définition 4.1 *Un graphe chain-probe (chain probe graph) est un graphe biparti $G = (V, E)$ tel qu'il existe un stable $S \subseteq V$ et un ensemble d'arêtes $F \subseteq S \times S$ tels que le graphe $G' = (V, E \cup F)$ soit biparti sans $2K_2$.*

Il est facile de voir que la classe des graphes chain-probe est héréditaire, et par conséquent il existe une famille de graphes minimaux qui ne sont pas de chain-probe. Nous avons pu montrer que cette famille contient six graphes, et déduire de la construction utilisée un algorithme de reconnaissance de complexité $\mathcal{O}(n^2)$. Ces travaux ont fait l'objet d'un article paru dans Annals of Operations Research [170].

4.2.1 Les six graphes minimaux interdits

Dans toute la suite, et comme précédemment, lorsque nous dirons qu'un graphe G contient un graphe H , il faudra comprendre que H est un sous-graphe induit de G . Tout d'abord, nous allons prouver le lemme suivant, qui nous sera utile par la suite (pour alléger l'écriture, nous noterons $G + F$ le graphe $(V_G, E_G \cup F)$) :

². On pourra consulter l'excellent site <http://www.graphclasses.org> pour plus de références sur les classes probe déjà étudiées.

Lemme 4.1 Soit G un graphe chain-probe avec des ensembles S et F comme dans la définition 4.1. Alors :

1. si G contient un P_5 $v_1-v_2-v_3-v_4-v_5$, alors S contient soit v_1 et v_4 , soit v_2 et v_5 , et aucun autre sommet du P_5 , et F contient respectivement soit v_1v_4 , soit v_2v_5 ;
2. si G contient un P_6 $v_1-v_2-v_3-v_4-v_5-v_6$, alors S contient v_2 et v_5 et aucun autre sommet du P_5 , et F contient v_2v_5 .

Démonstration. Supposons tout d'abord que G contienne un P_5 $v_1-v_2-v_3-v_4-v_5$. Alors F doit contenir l'une des arêtes v_1v_4 ou v_2v_5 sinon $\{v_1, v_2, v_4, v_5\}$ induit encore un $2K_2$ dans $G + F$. Donc S contient v_1, v_4 ou v_2, v_5 , et dans chaque cas, le fait qu'il soit stable entraîne qu'il ne contient aucun autre sommet du P_5 .

Supposons à présent que G contienne un P_6 $v_1-v_2-v_3-v_4-v_5-v_6$. Alors l'argument précédent peut être appliqué aux deux P_5 $v_1-\dots-v_5$ et $v_2-\dots-v_6$, et comme S est stable il est facile de voir que la seule possibilité est qu'il contienne v_2, v_5 et aucun autre sommet du P_6 , et que F contienne v_2v_5 . \square

Remarquons que les graphes bipartis ne sont pas tous des graphes chain-probe :

Lemme 4.2 Les graphes $3K_2$, P_7 , C_6 ainsi que les trois graphes notés H_1 , H_2 , H_3 (cf. figure 4.1) ne sont pas des graphes chain-probe.

Démonstration. Soit G l'un de ces six graphes, et supposons que G soit un graphe chain-probe avec des ensembles S et F comme dans la définition. Comme G contient un $2K_2$, nous avons $F \neq \emptyset$ et $|S| \geq 2$.

Supposons que G soit un $3K_2$ de sommets $\{v_1, v_2, v_3, w_1, w_2, w_3\}$ et d'arêtes $\{v_1w_1, v_2w_2, v_3w_3\}$. Comme S est stable, pour chaque $i \in \{1, 2, 3\}$, S ne peut contenir à la fois v_i et w_i . Donc, par symétrie, on peut supposer que S ne contient pas les w_i . Par conséquent, et puisque $G + F$ doit être biparti, F ne peut pas contenir les trois paires constituées de deux v_i , par exemple, F ne contient pas v_1v_3 . Mais alors les sommets v_1, w_1, v_3, w_3 induisent toujours un $2K_2$ dans $G + F$.

Supposons à présent que G soit un P_7 $v_1-\dots-v_7$; on peut appliquer le cas 2 du lemme 4.1 au P_6 $v_1-\dots-v_6$, ce qui implique que S contient v_2 et v_5 , et au P_6 $v_2-\dots-v_7$, ce qui implique que S contient v_3 et v_6 ; mais ceci contredit le fait que S est stable.

Supposons à présent que G soit un C_6 de sommets $\{v_1, \dots, v_6\}$ et d'arêtes $\{v_i v_{i+1} \text{ modulo } 6\}$; alors comme $F \neq \emptyset$ et que $G + F$ doit être biparti, on peut supposer sans perte de généralité que S contient les sommets v_1 et v_4 , et F l'arête v_1v_4 . Alors aucun autre sommet de G ne peut appartenir à S puisque cet ensemble doit être stable. Mais $G + F$ contient encore le $2K_2$ d'arêtes v_2v_3 et v_5v_6 .

Supposons enfin que G soit l'un des trois graphes H_1 , H_2 ou H_3 . On peut appliquer le cas 1 du lemme 4.1 au P_5 $a-b-c-c'-d'$, ce qui implique que

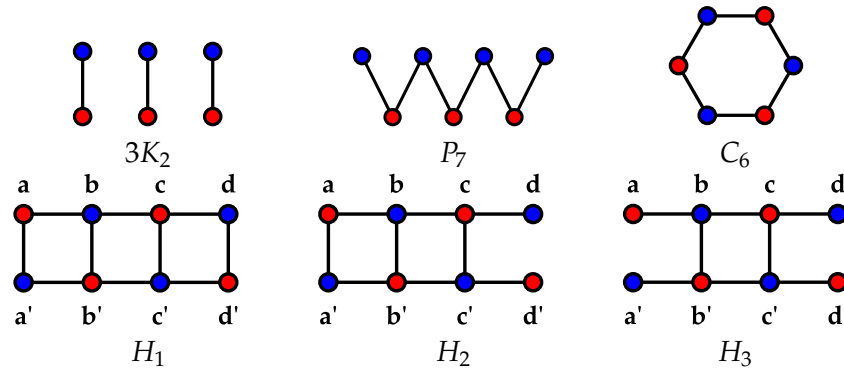


FIGURE 4.1 – Les 6 graphes minimaux qui ne sont pas chain-probe

S contient b ou c' , et au P_5 $a'-b'-c'-c-d$, ce qui implique que S contient b' ou c , et contredit le fait que S est stable. \square

Le théorème suivant prouve que ces six graphes sont les seuls graphes minimaux qui ne soient pas chain-probe.

Théorème 4.1 (Golumbic, Maffray et Morel (2009)) *Un graphe biparti est chain-probe si et seulement s'il est $\{3K_2, P_7, C_6, H_1, H_2, H_3\}$ -libre.*

La preuve de ce théorème étant assez longue, nous la donnons dans une section séparée.

4.2.2 Preuve du théorème 4.1

La condition nécessaire découle directement du lemme 4.2. Pour prouver la condition suffisante, nous allons construire le graphe étape par étape. Soit, donc, G un graphe biparti $\{3K_2, P_7, C_6, H_1, H_2, H_3\}$ -libre. Nous allons distinguer trois cas.

Premier cas : G contient un P_6

Alors il existe six ensembles stables X_1, \dots, X_6 , disjoints deux à deux et non vides, tels que pour $i = 1, \dots, 5$, tout sommet de X_i a un voisin dans X_{i+1} et tout sommet de X_{i+1} a un voisin dans X_i , et pour $i, j \in \{1, \dots, 6\}$ avec $|j - i| \geq 2$, il n'y a pas d'arête entre X_i et X_j . On pose alors $X = X_1 \cup \dots \cup X_6$. On choisit le sextuplet (X_1, \dots, X_6) tel que $X_3 \cup X_4$ est maximal, et sous cette condition l'ensemble X est aussi maximal.

Nous allons prouver que :

$$X_1 \cup X_2 \text{ et } X_5 \cup X_6 \text{ induisent des graphes sans } 2K_2. \quad (4.1)$$

En effet, supposons au contraire, et à symétrie près, qu'il y ait un $2K_2$ induit par des sommets $v_1, w_1 \in X_1$ et $v_2, w_2 \in X_2$, avec les arêtes x_1x_2 et w_1w_2 . Par définition des X_i , il existe une arête x_5x_6 entre $x_5 \in X_5$ et $x_6 \in X_6$. Mais alors les arêtes x_1x_2 , w_1w_2 et x_5x_6 forment un $3K_2$, une contradiction. Donc 4.1 est vérifiée.

D'après 4.1 et la proposition 2.4, il existe un sommet $v_1 \in X_1$ complet

à X_2 , un sommet $v_2 \in X_2$ complet à X_1 , un sommet $v_5 \in X_5$ complet à X_6 et un sommet $v_6 \in X_6$ complet à X_5 .

$$X_2 \text{ est complet à } X_3, \text{ et } X_4 \text{ est complet à } X_5. \quad (4.2)$$

En effet, supposons, à symétrie près, qu'il existe deux sommets non adjacents $w_2 \in X_2$ et $w_3 \in X_3$. Par définition des X_i , w_3 a un voisin $x_2 \in X_2$ et un voisin $x_4 \in X_4$; et x_4 a un voisin $x_5 \in X_5$ et d'après 4.1 x_5 est adjacent à v_6 . Mais alors $w_2-v_1-x_2-w_3-x_4-x_5-v_6$ induit un P_7 , une contradiction. Donc 4.2 est vérifiée.

$$X_1 \cup X_4 \text{ induit un graphe sans } 2K_2. \quad (4.3)$$

En effet, supposons que des sommets $v_3, w_3 \in X_3$ et $v_4, w_4 \in X_4$ induisent un $2K_2$ d'arêtes v_3v_4 et w_3w_4 . Alors $v_2, w_3, w_4, v_5, v_4, v_3$ induisent un C_6 , une contradiction. Donc 4.3 est vérifiée. D'après 4.3 et la proposition 2.4, il existe donc un sommet $v_3 \in X_3$ complet à X_4 , et un sommet $v_4 \in X_4$ complet à X_3 .

Notons que, d'après 4.1-4.3, l'ensemble X induit un sous-graphe connexe de G . De plus, tout sommet $x_1 \in X_1$ est contenu dans un P_6 $x_1-v_2-v_3-v_4-v_5-v_6$, et on a une propriété similaire pour X_2, \dots, X_6 . A présent, définissons $V_G \setminus X$ comme suit :

$$\begin{aligned} Y_2 &= \{v \notin X \mid v \text{ a un voisin dans } X_3 \text{ et est anticomplet à } X \setminus X_3\}. \\ Y_5 &= \{v \notin X \mid v \text{ a un voisin dans } X_4 \text{ et est anticomplet à } X \setminus X_4\}. \\ Z_2 &= \{v \notin X \mid v \text{ est complet à } X_1 \cup X_3 \cup X_5\}. \\ Z_5 &= \{v \notin X \mid v \text{ est complet à } X_2 \cup X_4 \cup X_6\}. \\ T &= \{v \notin X \mid v \text{ est anticomplet à } X\}. \end{aligned}$$

Alors :

$$X_1, \dots, X_6, Y_2, Y_5, Z_2, Z_5, T \text{ forment une partition de } V. \quad (4.4)$$

Ces onze ensembles sont clairement disjoints deux à deux. Nous considérons à présent un sommet $v \in V \setminus X$ et nous allons montrer qu'il appartient à l'un des ensembles Y_2, Y_5, Z_2, Z_5, T . Si v n'a pas de voisin dans X , il se trouve dans T . Donc supposons que v ait un voisin dans X . Comme G est biparti et X est connexe, v peut avoir un sommet dans $X_1 \cup X_3 \cup X_5$ or dans $X_2 \cup X_4 \cup X_6$, mais pas dans les deux. A symétrie près, supposons qu'il ait un voisin dans $X_1 \cup X_3 \cup X_5$.

Supposons d'abord que v n'ait pas de voisin dans X_1 . S'il a un voisin dans X_3 et un voisin dans X_5 , v peut être ajouté à X_4 , ce qui contredit alors l'hypothèse de maximalité de X ; on a la même contradiction si v a un voisin seulement dans X_5 , car il devrait se trouver dans X_6 . Donc v a un voisin dans X_3 seulement, et il appartient donc à Y_2 .

Supposons à présent que v ait un voisin x_1 dans X_1 . Si v a un non-voisin w_3 dans X_3 , alors soit $v, x_1, v_2, w_3, v_4, v_5, v_6$ induisent un P_7 , soit $v, x_1, v_2, w_3, v_4, v_5$ induisent un C_6 , selon que v est adjacent ou non à v_5 . Par conséquent, v est complet à X_3 , et s'il n'a aucun voisin dans X_5 , on peut l'ajouter à X_2 , une contradiction. Donc v a un voisin x_5 dans X_5 ; et s'il a aussi un non-voisin w_5 dans X_5 , alors $v, x_1, v_2, v_3, v_4, x_5, w_5, v_6$ induisent

un H_1 . Par conséquent, v est complet à X_5 . Enfin, si v a un non-voisin dans X_1 , le sextuplet $(X \setminus N(v), \{v_2\}, X_3 \cup (X_1 \cap N(v)), X_4 \cup \{v\}, X_5, X_6)$ contredit la maximalité de $X_3 \cup X_4$. Ainsi, v est complet à X_1 , et finalement $v \in Z_2$. Par symétrie, si v a un voisin dans $X_2 \cup X_4 \cup X_6$, alors il se trouve dans Y_5 ou Z_5 . Donc 4.4 est vérifiée.

Les relations entre X_3 et X_4 peuvent être décrites plus finement en partitionnant ces ensembles comme suit :

$$\begin{aligned} X'_3 &= \{v \in X_3 \mid v \text{ a un voisin dans } Y_2\}. \\ X''_3 &= X_3 \setminus X'_3. \\ X'_4 &= \{v \in X_4 \mid v \text{ a un voisin dans } Y_5\}. \\ X''_4 &= X_4 \setminus X'_4. \end{aligned}$$

Nous prétendons à présent que :

$$X'_3 \text{ est complet à } X_4. \quad (4.5)$$

$$X'_4 \text{ est complet à } X_3. \quad (4.6)$$

Prouvons 4.5. Soient $x_3 \in X'_3$, $x_4 \in X_4$ et y_2 un voisin de x_3 dans Y_2 . Par définition de X_3 et X_4 , x_3 est adjacent à v_4 , et x_4 est adjacent à v_3 . Maintenant, si x_3 et x_4 ne sont pas adjacents, les sommets $y_2, x_3, v_4, v_5, v_1, v_2, v_3, x_4$ induisent un H_2 . Donc 4.5 est vérifiée, et par symétrie, 4.6 aussi.

$$(X'_3 \cup Y_2) \text{ et } (X'_4 \cup Y_5) \text{ induisent des graphes sans } 2K_2. \quad (4.7)$$

Supposons que $X'_3 \cup Y_2$ contienne un $2K_2$, de sommets $x, x' \in X_3, y, y' \in Y_2$ et d'arêtes xy et $x'y'$. Alors $y', x', v_4, v_5, v_1, v_2, x, y$ induisent un H_3 . Donc $(X'_3 \cup Y_2)$ induit un graphe sans $2K_2$, et de manière similaire $(X'_4 \cup Y_5)$ induit un graphe sans $2K_2$. Donc 4.7 est vérifiée.

Notons que $Y_5 \cup Z_5$ est un stable, parce qu'il est inclus dans le voisinage de v_4 , et de manière similaire $Y_2 \cup Z_2$ est stable. Alors :

$$Z_5 \text{ est complet à } Z_2 \cup Y_2, \text{ et } Z_2 \text{ est complet à } Z_5 \cup Y_5. \quad (4.8)$$

Supposons au contraire, et à symétrie près, qu'il existe des sommets non adjacents $z_5 \in Z_5$ et $u \in Z_2 \cup Y_2$. Si $u \in Z_2$, alors $v_1, v_2, v_5, v_6, z_5, u$ induisent un C_6 . Si $u \in Y_2$, alors $v_1, v_2, v_3, v_4, v_5, v_6, z_5, u$ induisent un H_2 . Donc 4.8 est vérifiée.

$$T \cup Y_2 \cup Y_5 \text{ est stable.} \quad (4.9)$$

Supposons au contraire, et à symétrie près, qu'il existe deux sommets adjacents u et v dans $T \cup Y_2 \cup Y_5$. Alors G contient un $3K_2$ d'arêtes v_1v_2, v_5v_6, uv . Donc 4.9 est vérifiée.

Soient T_2 l'ensemble des sommets de T qui ont un voisin dans Z_5 , T_5 l'ensemble des sommets de T qui ont un voisin dans Z_2 , et T_0 l'ensemble des sommets de T qui n'ont aucun voisin dans $Z_5 \cup Z_2$. D'après 4.8, aucun sommet de T ne peut avoir de voisins à la fois dans Z_2 et Z_5 , car G est biparti. Donc T_0, T_2, T_5 sont disjoints et leur union est T . Alors :

$$Z_2 \cup T_5 \text{ et } Z_5 \cup T_2 \text{ induisent des graphes sans } 2K_2. \quad (4.10)$$

Supposons au contraire, et à symétrie près, que G contienne un $2K_2$ de sommets $z, z' \in Z_2, t, t' \in T_5$ et d'arêtes zt et $z't'$. Alors $v_1, v_2, v_5, v_6, z, t, t', z'$ induisent un H_3 . Donc 4.10 est vérifiée.

Notons que, par définition de T et 4.9, T_0 est l'ensemble des sommets isolés de G . De plus, $G \setminus T_0$ est connexe, et les deux côtés de son unique partition en deux stables sont $X_1 \cup X_3 \cup X_5 \cup Y_5 \cup Z_5 \cup T_5$ et $X_2 \cup X_4 \cup X_6 \cup Y_2 \cup Z_2 \cup T_2$, appelés respectivement côté impair et côté pair de $G \setminus T_0$. La figure 4.2 montre la structure du graphe G , où les cercles représentent les douze ensembles $X_1, \dots, X_6, Y_2, Y_5, Z_2, Z_5, T_2, T_5$ (T_0 n'est pas représenté), une ligne épaisse indique que toutes les arêtes sont présentes entre les ensembles, et une ligne fine indique un sous-graphe biparti sans $2K_2$, et l'absence de ligne indique qu'il n'y a aucune arête entre les deux ensembles.

Soit $S = X_2 \cup Y_2 \cup X_5 \cup Y_5$. Les affirmations précédentes impliquent que S est un ensemble stable de G . Soit $F = (X_2 \otimes X_5) \cup (X_2 \otimes Y_5) \cup (X_5 \otimes Y_2)$, où $A \otimes B$ désigne l'ensemble de toutes les paires $\{a, b\}$ avec $a \in A$ et $b \in B$. Alors :

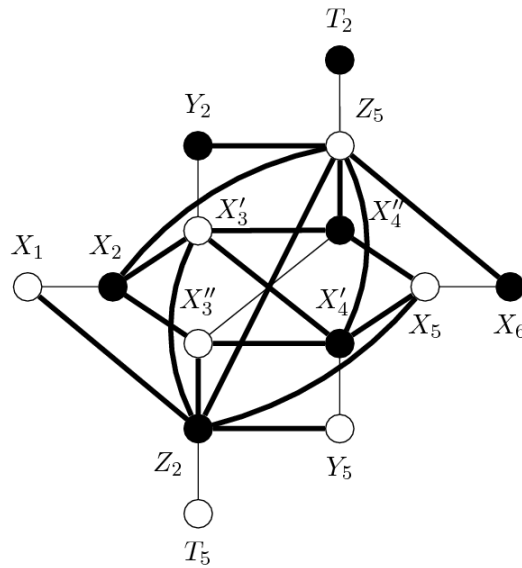
$$G + F \text{ est un graphe sans } 2K_2. \quad (4.11)$$

Pour le voir, considérons l'ordre des sommets du côté pair de $(G + F) \setminus T_0$ qui est représenté par l'inégalité $Z_2 \succeq X_2 \succeq X'_4 \succeq X''_4 \succeq Y_2 \succeq X_6 \succeq T_2$, qui signifie que chaque sommet dans l'un de ces ensembles domine tous les sommets des ensembles qui se trouvent à sa droite dans l'inégalité ; et à l'intérieur de chaque ensemble les sommets sont ordonnés comme suit : dans Z_2 , les sommets sont ordonnés selon la chaîne impliquée par 4.10 ; dans X_2 , ils sont ordonnés selon la chaîne impliquée par 4.1 ; dans X'_4 , selon 4.7 ; dans X''_4 , selon 4.3 ; dans Y_2 , selon 4.7 ; dans X_6 , selon 4.1 ; dans T_2 , selon 4.10. C'est un exercice simple de vérifier que cet ordre est un ordre total sur les sommets du côté pair de $(G + F) \setminus T_0$ selon l'inclusion de leur voisinage. Donc d'après la proposition 2.4, 4.11 est vérifiée. Ceci termine la preuve de ce premier cas.

En outre, on peut voir que les ensembles S et F mentionné dans 4.11 sont minimaux pour la propriété souhaitée. En effet, considérons deux tels ensembles S' et F' . D'après le cas 2 du lemme 4.1 appliqué aux P_6 $v_1-x_2-v_3-v_4-x_5-v_6$ pour tout $x_2 \in X_2$ et tout $x_5 \in X_5$, l'ensemble S' doit contenir $X_2 \cup X_5$ et F' doit contenir $X_2 \otimes X_5$; d'après le cas 1 du lemme 4.1 appliqué aux P_5 $v_1-x_2-v_3-v_4-y_5$ pour tout $x_2 \in X_2$ et tout $y_5 \in Y_5$, l'ensemble S' doit contenir Y_5 , et F' doit contenir $X_2 \otimes Y_5$, et par un argument symétrique S' doit contenir Y_2 et F' doit contenir $X_5 \otimes Y_2$. Donc S' contient S et F' contient F .

Second cas : G ne contient pas de P_6 mais contient un P_5

On peut commencer la construction du graphe comme précédemment : dans ce cas, il existe cinq ensembles stables X_1, \dots, X_5 , disjoints deux à deux et non vides, tels que pour $i = 1, \dots, 4$, tout sommet de X_i a un voisin dans X_{i+1} et tout sommet de X_{i+1} a un voisin dans X_i , et pour $i, j \in \{1, \dots, 5\}$ avec $|j - i| \geq 2$, il n'y a pas d'arête entre X_i et X_j . On choisit le quintuplet (X_1, \dots, X_5) tel que l'ensemble $X = X_1 \cup \dots \cup X_5$ est maximal. Nous définissons alors des sous-ensembles de $V_G \setminus X$ comme suit :

FIGURE 4.2 – Forme générale d'un graphe chain-probe contenant un P_6 .

$$Y = \{v \notin X \mid v \text{ a un voisin dans } X_3 \text{ et est anticomplet à } X \setminus X_3\}.$$

$$Z = \{v \notin X \mid v \text{ est complet à } X_1 \cup X_3 \cup X_5\}.$$

$$T = \{v \notin X \mid v \text{ est anticomplet à } X\}.$$

Les preuves des faits suivants sont similaires aux preuves du cas précédent, et nous les omettons :

$$X_1 \cup X_2 \text{ et } X_4 \cup X_5 \text{ induisent des graphes sans } 2K_2 \quad (4.12)$$

$$X_3 \text{ est complet à } X_2 \cup X_4. \quad (4.13)$$

$$Y \cup X_3 \text{ induit un graphe sans } 2K_2. \quad (4.14)$$

$$Y \cup T \text{ sont des stables.} \quad (4.15)$$

$$Z \cup T \text{ induit un graphe sans } 2K_2. \quad (4.16)$$

L'ensemble T_0 des sommets de T qui n'ont pas de voisin dans Z est l'ensemble des sommets isolés de G . De plus, $G \setminus T_0$ est connexe, et les deux côtés de son unique bipartition en deux ensembles stables sont $X_1 \cup X_3 \cup X_5 \cup (T \setminus T_0)$ et $X_2 \cup X_4 \cup Y \cup Z$, que nous appelons respectivement côté impair et côté pair de $G \setminus T_0$.

Soit $S_1 = X_1 \cup X_4$. Clairement, S_1 est stable. Soit $F_1 = X_1 \otimes X_4$. Alors :

$$G + F_1 \text{ est un graphe sans } 2K_2. \quad (4.17)$$

Pour le voir, considérons l'ordre des sommets du côté pair de $(G + F) \setminus T_0$ représenté par l'inégalité $Z \succeq X_4 \succeq X_2 \succeq Y$, qui signifie que chaque sommet dans l'un de ces ensembles domine tous les sommets des ensembles qui se trouvent à sa droite dans l'inégalité; et à l'intérieur de chaque ensemble les sommets sont ordonnés comme suit : dans Z , ils sont ordonnés selon la chaîne impliquée par 4.16, dans X_2 et X_4 , selon la chaîne impliquée par 4.13; et dans Y selon un ordre arbitraire. C'est un exercice simple de vérifier que cet ordre définit une chaîne sur les sommets du côté pair

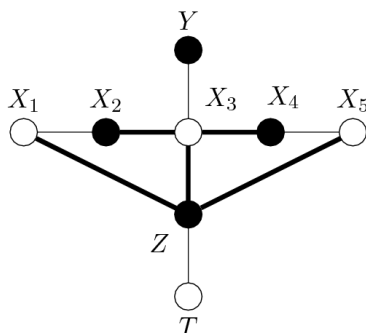


FIGURE 4.3 – Forme générale d'un graphe chain-probe contenant un P_5 mais pas de P_6 .

de $(G + F) \setminus T_0$. Ainsi, d'après la proposition 2.4, 4.17 est vérifiée.

Par symétrie, soient $S_5 = X_5 \cup X_2$ et $F_5 = X_5 \times X_2$. Alors $G + F_5$ est encore un graphe biparti sans $2K_2$. Comme pour le cas précédent, il est simple de vérifier (et nous ne redonnons pas la preuve) que F_1 et F_5 sont les seuls ensembles minimaux F tels que $G + F$ est biparti sans $2K_2$. La figure 4.3 donne la structure générale d'un tel graphe.

Troisième cas : G ne contient pas de P_5

Comme G ne contient pas de $3K_2$, il a au plus deux composantes de taille au moins 2. De plus, chaque composante de G est sans $2K_2$ (car l'existence d'un $2K_2$ dans un graphe biparti connexe implique l'existence d'un P_5). Ainsi, si G a au plus une composante de taille au moins 2, il est sans $2K_2$ et c'est donc un graphe chain-probe avec $S = \emptyset$ et $F = \emptyset$. Supposons donc que G a exactement deux composantes connexes G_1 et G_2 de taille au moins 2. Pour $i = 1, 2$, soit A_i, B_i l'unique bipartition de G_i en deux stables. Posons alors $S = A_1 \cup B_2$ (c'est bien un stable) et $F = A_1 \otimes B_2$. Alors il est facile de voir que $G + F$ est un graphe sans $2K_2$, avec pour ordre sur les sommets $A_1 \succeq A_2$, les sommets de A_i conservant le même ordre que dans G_i . G est donc un graphe chain-probe, ce qui termine la preuve du théorème. \square

4.2.3 Considérations algorithmiques

Le théorème implique que les graphes chain-probe peuvent être reconnus en temps polynomial. En effet, les six graphes interdits ont au plus huit sommets, et il existe donc un algorithme de recherche exhaustive de complexité $\mathcal{O}(n^8)$ qui décide si un graphe G d'ordre n est chain-probe. On peut améliorer cette complexité de manière significative en suivant les étapes de la preuve du théorème. Tout d'abord, on décide si chaque composante de G est sans $2K_2$ et si G a au plus deux composantes de taille au moins 2. Cette étape peut être réalisée facilement en temps $\mathcal{O}(n^2)$. Si la réponse à ce premier test est positive, alors G est chain-probe comme expliqué dans le troisième cas de la preuve. Si la réponse est négative, alors soit G a trois composantes de taille au moins 2, et on peut exhiber un $3K_2$ induit et s'arrêter, soit G contient un P_5 . Dans ce dernier cas, on commence la construction comme dans le deuxième cas de la preuve. Il est facile de voir que cette construction peut être effectuée en temps $\mathcal{O}(n^2)$. Si toutes les propriétés listées dans le deuxième cas sont satisfaites, alors G

est chain-probe ; sinon, soit G contient l'un des six sous-graphes interdits, auquel cas on exhibe ce graphe et s'arrête, soit G contient un P_6 . À nouveau, il est facile de voir que la construction donnée dans le premier cas de la preuve peut être réalisée en temps $\mathcal{O}(n^2)$. Si toutes les propriétés listées dans le premier cas sont vérifiées, G est chain-probe ; sinon G contient l'un des graphes interdits : on l'exhibe et on s'arrête. Finalement, l'algorithme dans son intégralité fonctionne en $\mathcal{O}(n^2)$:

Théorème 4.2 (Golumbic, Maffray et Morel (2009)) *On peut décider en temps $\mathcal{O}(n^2)$ si un graphe d'ordre n est chain-probe.*

CONCLUSION DU CHAPITRE

Dans ce chapitre, nous avons présenté les notions de problèmes de type *sandwich* et de type *probe*. Ce dernier problème avait été étudié à l'origine dans la classe des graphes d'intervalles, avec des applications en génétique. Nous l'avons regardé sous l'angle des graphes de chaîne, c'est-à-dire les graphes bipartis sans $2K_2$. Nous avons alors pu obtenir une caractérisation de ces graphes, sous la forme d'une liste de six graphes minimaux interdits, ainsi qu'une décomposition structurelle complète. À partir de cette structure, nous avons même pu déduire un algorithme de reconnaissance de complexité $\mathcal{O}(n^2)$, surpassant ainsi largement l'algorithme « naïf » induit par les six graphes interdits.

Ainsi que nous l'avons mentionné au début de ce chapitre, les graphes que nous appelons *chain-probe* ne sont pas tous solutions de la version *partitionnée* du problème (cela dépend en fait de la partition fournie). Très récemment, Le a caractérisé ce qu'on pourrait appeler les graphes *chain-probe partitionnés* de deux manières différentes : d'une part par une liste de six sous-graphes interdits, d'autre part *via* un certain graphe *amélioré* : G est un graphe *chain-probe partitionné* si et seulement si le graphe amélioré G^* est un graphe de chaîne.

Signalons pour terminer que récemment, nous avons été contactés par un groupe d'élèves de classes préparatoires qui, après avoir trouvé sur l'Internet la présentation donnée aux Journées Graphes et Algorithmes, ont choisi ce thème comme sujet de leur T.I.P.E.

CONCLUSION GÉNÉRALE

La puissance de la théorie des graphes est de permettre de modéliser étonnement simplement un nombre impressionnant de problèmes auxquels nous sommes quotidiennement (consciemment ou non) confrontés : problèmes de transport, de routage dans les réseaux, d'emploi du temps, de gestion de projets, etc. Même certaines sous-structures d'un graphe, comme les ensembles stables, sont d'une grande utilité : la taille d'un plus grand stable possible dans un graphe nous renseigne ainsi sur la quantité d'une certaine ressource utilisable sans créer de conflit. Malheureusement, on sait aujourd'hui qu'il peut être très difficile de calculer la valeur exacte, et même une valeur suffisamment proche de la valeur exacte, d'un stable de taille maximum dès que le nombre de sommets du graphe devient trop important.

Mais l'on sait par ailleurs que ce même problème peut être facilement résolu dans un grand nombre de classes de graphes particulières (graphes bipartis, cographes, graphes d'intervalles, etc.). Les deux résultats fondamentaux dus à Alekseev entraînent que la classe des graphes sans P_5 est la plus petite classe définie par un seul sous-graphe connexe interdit pour laquelle on ignore encore la complexité du problème du stable maximum. Cette question a été la motivation de cette thèse.

Dans ce manuscrit, nous avons, après avoir rappelé toutes les définitions de base en théorie des graphes, théorie de la complexité et théorie polyédrale, dressé un état de l'art extrêmement complet des propriétés des graphes sans P_5 (notamment les graphes sans P_5 bipartis) ainsi que de l'étude de la stabilité dans ces graphes (il suffit de regarder le nombre de références figurant dans la bibliographie pour s'en convaincre). Un nombre impressionnant de résultats existe sur le sujet, montrant ainsi la puissance de certains outils développés par les chercheurs. Nous en avons abordé quatre : la technique des graphes augmentants, la décomposition modulaire, la structure et la décomposition par cliques d'articulation ; et pour chacune nous avons montré comment elle avait été utilisée pour apporter des contributions à la résolution de cette question, qualifiée de « problème ouvert important » sur le site <http://www.graphclasses.org>. Dans ce même chapitre nous avons présenté deux de nos résultats : tout d'abord une amélioration, grâce à une meilleure utilisation des structures de données, de l'algorithme permettant d'identifier une clique dominante dans un graphe sans P_5 et sans C_5 , ramenant ainsi sa complexité de $\mathcal{O}(n^3)$ à $\mathcal{O}(m)$; ensuite, nous avons montré comment obtenir en temps polynomial un algorithme permettant de calculer un stable de poids maximum dans une certaine classe de graphes (résultat qui avait en fait été énoncé auparavant sous une autre forme par Hoàng et Reed).

En conclusion de ce chapitre, nous avons dressé un diagramme très complet des relations d'inclusion existant entre les nombreuses classes présentées précédemment, et qui nous l'espérons permettra à la fois au lecteur de mieux comprendre comment s'articulent ces classes entre elles, et qui pourra lui servir de référence s'il recherche un résultat sur une classe particulière.

Le troisième chapitre constitue notre contribution majeure à la résolution du problème : il s'agit de l'étude de la classe des graphes sans P_5 3-colorables. Nous avons été en mesure d'élaborer des algorithmes *linéaires* permettant de reconnaître, colorer, et calculer un stable de poids maximum pour les graphes de cette classe. La conséquence importante de ce résultat est qu'il implique, par effet dominant, une diminution d'une unité de la complexité d'un algorithme connu jusqu'alors pour calculer un stable de poids maximum dans un graphe sans P_5 k -colorable, quel que soit $k > 3$.

Le quatrième chapitre comprend un résultat important que nous avons obtenu : la caractérisation des graphes *chain-probe*, qui sont les graphes auxquels il est possible de rajouter des arêtes entre certains sommets d'un stable de sorte que le graphe résultant soit un graphe de chaîne, c'est-à-dire biparti connexe et sans P_5 . La résolution de ce problème répond à une question posée par Martin Golumbic lors d'un séjour au laboratoire G-SCOP, avec des applications en intelligence artificielle et plus particulièrement dans le domaine du raisonnement automatisé. Ainsi, nous avons montré qu'il existe six graphes minimaux qui ne sont pas *chain-probe*. Le plus grand de ces six graphes ayant huit sommets, on en déduit immédiatement un algorithme de complexité $\mathcal{O}(n^8)$ permettant de reconnaître les graphes *chain-probe* ; cependant, le raisonnement utilisé dans la preuve nous a conduit à élaborer un algorithme beaucoup plus efficace, de complexité $\mathcal{O}(n^2)$.

Enfin, l'annexe comprend un historique de la théorie des graphes à travers cinq problèmes majeurs et cinq mathématiciens, dont le choix a certainement une part de subjectivité. Ce qui devait être au départ un court paragraphe d'introduction dans le premier chapitre s'est rapidement transformé en un passionnant travail de recherche documentaire, et de lecture d'articles parfois très anciens (aux environs de 1870 pour certains) et dont nous donnons, pour la première fois à notre connaissance, l'ensemble des liens permettant d'accéder à ceux tombés dans le domaine public.

Ainsi s'achève cette thèse. Le problème du stable maximum dans les graphes sans P_5 n'est toujours pas résolu, mais nous espérons avoir apporté notre pierre à cette tour qui s'élève vers la solution, et qui nous permettra peut-être un jour, à nous ou à nos successeurs, d'atteindre ce but. Nous espérons en tout cas que vous, qui parcourrez ces dernières lignes, avez pris autant de plaisir à lire cette thèse que nous en avons eu à l'écrire.

ANNEXES



SOMMAIRE

A.1 HISTORIQUE À TRAVERS CINQ MATHÉMATIENS ET CINQ PROBLÈMES	169
A.1.1 Euler et le problème des 7 ponts de Königsberg	169
A.1.2 Cayley et le dénombrement des arbres	170
A.1.3 Petersen et la théorie des graphes réguliers	172
A.1.4 Guthrie et le problème des quatre couleurs	175
A.1.5 Berge et les graphes parfaits	178
A.2 HISTOIRE DE LA THÉORIE DE LA COMPLEXITÉ	182

Initialement, nous avons prévu de débiter chacune des trois parties du chapitre 1 par de brefs rappels historiques ; sans doute par goût personnel pour l'histoire des sciences, mais aussi parce que nous sommes convaincus qu'on ne peut étudier une théorie sans se poser la question de ses origines, pourquoi et dans quel contexte ces outils mathématiques ont-ils été élaborés. Ainsi voulions-nous, à travers ces lignes, répondre à des questions que pourrait légitimement se poser toute personne un peu curieuse ayant déjà rencontré des graphes ("*Qui a le premier introduit le mot graphe ?*", "*Euler a-t-il vraiment manipulé les premiers graphes ?*", "*Pourquoi Berge a-t-il qualifié certains graphes de parfaits ?*"...) ou étudié la théorie de la complexité. Nous sommes ainsi parti à la recherche des textes fondateurs de ces théories (les articles de Cayley, Sylvester, Petersen, pour les débuts de la théorie des graphes au XIX^e siècle ; prémisses de la notion de complexité algorithmique chez Lamé, aussi au XIX^e siècle). Nous encourageons d'ailleurs le lecteur à se plonger dans la lecture de ces articles riches de connaissances, grâce aux liens que nous avons fournis dans la bibliographie.

A.1 HISTORIQUE À TRAVERS CINQ MATHÉMATIENS ET CINQ PROBLÈMES

A.1.1 Euler et le problème des 7 ponts de Königsberg

Il est désormais admis de faire remonter l'origine de la théorie des graphes à l'année 1736, date à laquelle Euler¹ publia dans les *Commentarii Academiae scientiarum imperialis Petropolitanae* (Réflexions de l'Académie des Sciences de Saint-Petersbourg) un article intitulé *Solutio problematis ad geometriam situs pertinentis* (Solution d'un problème appartenant à la géométrie de position) [127] traitant de la solution au *problème des 7 ponts de Königsberg*². Bien que ce problème soit devenu célèbre, nous pensons qu'il mérite donc d'être rappelé ici.

La ville de Königsberg, en Prusse (aujourd'hui l'oblast de Kaliningrad, en Russie), était construite autour de deux îles, reliées par un pont, et reliées au continent par six autres ponts, comme l'illustre la figure suivante :

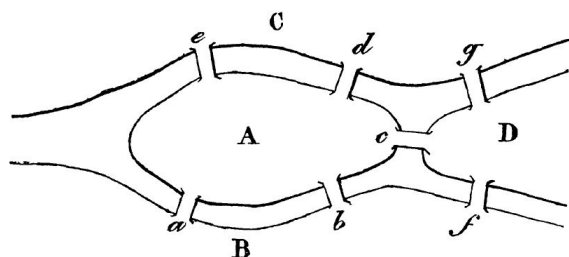


FIGURE A.1 – Les sept ponts de Königsberg (source : [100]).

La question que les habitants de Königsberg se posaient à l'époque était de déterminer s'il existe ou non une promenade à travers la ville, telle que l'on passe exactement une fois par chaque pont. Euler a trouvé une solution élégante à ce problème, mais qui, en réalité, ne repose nullement sur une modélisation « graphique ». En fait, Euler note le passage de la région A à la région B par la suite de caractères AB, puis par ABC si l'on passe ensuite de la région B à la région C, et ainsi de suite ; il remarque alors qu'une promenade solution est une suite de huit caractères, et que celle-ci doit comprendre deux fois les caractères A et B consécutivement (il en est d'ailleurs de même pour les caractères A et C). Euler note ensuite que si une région peut être atteinte par un nombre impair p de ponts, le caractère associé à la région doit apparaître $\frac{p+1}{2}$ fois dans la chaîne ; ainsi le caractère A doit en fait figurer trois fois parmi les huit caractères, et les caractères B, C et D deux fois chacun : il est donc impossible de construire la chaîne demandée, et par conséquent le problème n'admet aucune solution.

1. Leonhard Euler (1707 – 1783), mathématicien suisse, considéré comme l'un des plus grands mathématiciens de tous les temps. Il est également connu pour ses travaux en mécanique, en dynamique des fluides, en optique et en astronomie.

2. En réalité, Euler a présenté cet article le 26 août 1735 devant l'Académie des Sciences de Saint-Petersbourg. Bien que daté de 1736, il ne fut publié qu'en 1741, puis en 1752 dans une nouvelle édition des *Commentarii*. En 1851, E. Coupy en a publié la première traduction française [100].

Euler généralise ensuite ce procédé à tous les problèmes du même genre (c'est-à-dire avec un nombre arbitraire de régions et de ponts les joignant), procédé qui permet par ailleurs soit de construire une promenade si elle existe, soit d'exhiber un argument qui prouve qu'il n'existe pas de telle solution ; dans les deux cas, il obtient donc ce que l'on appelle aujourd'hui un *certificat*.

Mais l'ingéniosité d'Euler ne s'arrête pas là puisqu'il propose dans la suite de son article une méthode encore plus simple. Il note en effet que si l'on fait la somme du nombre de ponts jouxtant chaque région, on obtient toujours un nombre pair, égal à deux fois le nombre de ponts total (résultat connu aujourd'hui sous le nom de *lemme des poignées de mains*), puisque chaque pont est compté pour deux régions (ainsi dans le cas des ponts de Königsberg, on a 5 pour la région *A*, et 3 pour chacune des régions *B*, *C* et *D*, soit un total de 14) et il doit donc y avoir un nombre pair de régions desquelles partent un nombre impair de ponts. Il en déduit alors facilement qu'une condition nécessaire pour qu'il existe une solution au problème général est que chaque région possède un nombre pair de ponts permettant de la rallier (sauf éventuellement deux d'entre elles si l'on ne s'oblige pas à terminer la promenade d'où on est parti). Cette condition est en fait également suffisante, mais Euler n'en donne pas la démonstration dans son article ; la première preuve complète de ce que l'on appelle aujourd'hui *théorème d'Euler* ou *caractérisation des graphes eulériens* ne sera donnée qu'en 1873 par Carl Hierholzer [205]³.

Ainsi, Euler n'a jamais directement manipulé de graphes au sens où on l'entend aujourd'hui. On trouve cependant dans son article plusieurs notions fondamentales de la théorie des graphes (lemme des poignées de mains, graphes eulériens), et c'est pour cette raison que cet article est souvent considéré, de nos jours, comme le premier de qui allait devenir la théorie des graphes. Notons enfin que, selon [177, p. 31], c'est W.W. Rouse Ball⁴ qui, le premier, donna en 1892 une modélisation sous forme de graphe du problème des sept ponts de Königsberg [21] ; cependant, comme beaucoup de ses contemporains, Ball ne parle pas de *graphe*, mais emploie le terme *réseau*. Pourtant, le terme *graphe* commençait à être employé depuis quelques années, et contrairement à ce que l'on pourrait croire, il n'est pas issu que des mathématiques. . .

A.1.2 Cayley et le dénombrement des arbres

Le concept mathématique d'arbre (cf. figure A.2) apparaît tout d'abord implicitement dans les travaux de Kirchhoff⁵, qui a utilisé des idées proches de la théorie des graphes dans ses études des circuits électriques,

3. Plus exactement, Hierholzer (1840 – 1871) aurait expliqué sa preuve à un collègue juste avant sa mort prématurée en 1871, collègue qui aurait ensuite fait publier cette preuve en 1873.

4. Walter William Rouse Ball (1850 – 1925), mathématicien britannique.

5. Gustav Robert Kirchhoff (1824 – 1887), physicien allemand, est l'auteur des lois des nœuds et des mailles qui portent son nom, établies en 1845 alors qu'il était encore étudiant.

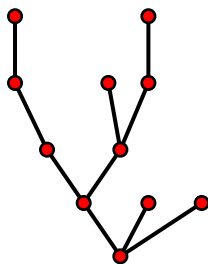


FIGURE A.2 – Un arbre au sens de la théorie des graphes.

dans les années 1840. Plus tard, à la fin des années 1850, Cayley⁶ travaille sur un problème de calcul différentiel posé par Sylvester⁷, et imagine une représentation graphique des expressions analytiques qu’il manipule, qu’il nomme *arbre*. Ce problème l’amène alors à étudier la question du dénombrement des arbres *enracinés*⁸ [62, 63].

A la même période, le chimiste Crum Brown⁹ commence à représenter graphiquement les composés chimiques, en notant dans des cercles les symboles des atomes, qu’il relie entre eux d’abord au moyen de lignes à tirets [57], puis plus simplement par des lignes pleines [58]. Il n’est cependant pas le premier : le chimiste irlandais William Higgins (1763 – 1825) s’y était déjà essayé en 1789, mais son idée n’avait pas retenu l’attention (en particulier parce que ses molécules étaient purement spéculatives) et un autre chimiste écossais, Couper¹⁰, avait introduit en 1858 un formalisme très proche de celui utilisé aujourd’hui [99].

Cayley, qui s’intéresse aussi à la chimie, est au courant de ces travaux. Il s’aperçoit donc rapidement de l’analogie entre ses arbres et les molécules de Crum Brown ; en particulier, il voit que le problème de l’énumération des isomères des alcanes peut se ramener au problème de l’énumération des arbres sans racine, et en donne une solution en 1874 [64]. Il publie ensuite, en 1875, un article unifiant et corrigeant ces résultats¹¹ [65], avant de simplifier sa formule en 1881 [67]. Enfin, en 1889, Cayley publie l’article dans lequel il énonce la célèbre formule qui porte son nom : il existe n^{n-2} arbres à n sommets étiquetés [68]¹².

Il faut remarquer que le terme *graphe* n’apparaît dans aucun des articles de Cayley précédemment cités (ni dans ceux qu’il écrira par la suite,

6. Arthur Cayley (1821 – 1895), avocat puis mathématicien britannique, est l’auteur d’un très grand nombre d’articles. De nombreuses notions portent son nom et il a notamment introduit la multiplication des matrices et les octonions.

7. James Joseph Sylvester (1814 – 1897), mathématicien anglais, est l’auteur de plusieurs centaines d’articles. Il est à l’origine des termes *matrice*, *discriminant* et *graphe*.

8. C’est-à-dire possédant un sommet distingué appelé *racine*.

9. Alexander Crum Brown (1838 – 1922), chimiste écossais.

10. Archibald Scott Couper (1831 – 1892), chimiste écossais qui développa une théorie de la structure chimique.

11. Certaines valeurs données dans l’article de 1857 étaient erronées.

12. Cayley ne fait bien qu’énoncer cette formule : il explique pourquoi elle est valide pour $n = 6$, mais ne donne pas de preuve pour le cas général. La première preuve correcte a été donnée en 1918 par le mathématicien allemand Heinz Prüfer (1896 – 1934), qui a développé pour cela le *code de Prüfer* [299].

d'ailleurs). Dans l'article de 1857, il introduit le mot *arbre*, et utilise les termes *nœud* et *branche*, qui apparaissent aussi dans les articles postérieurs, hormis celui de 1874 où il utilise respectivement les termes *diagramme*, *point* et *lien* probablement pour renforcer le lien avec la chimie.

Le problème analytique auquel s'intéressait Cayley, et qui fut au centre d'un important effort de recherche au XIX^e siècle, est celui des *formes binaires en théorie des invariants*, une branche de l'algèbre générale. Sans rentrer dans les détails, on appelle *forme binaire* un polynôme homogène (c'est-à-dire dont tous les monômes ont même degré total) à deux variables (par exemple $x^5 + 2x^3y^2 + 9xy^4$ est une forme binaire de degré 5), et un *invariant* est un polynôme qui ne « change pas » lorsqu'on lui applique une transformation linéaire.

Dès 1878, Sylvester imagine un moyen de représenter les formes binaires (qu'il appelle *binary quantics*) à l'aide de dessins semblables à des molécules chimiques, ainsi qu'il l'explique dans une note publiée dans *Nature* [324] :

L'analogie est entre les atomes et les formes binaires exclusivement. Je compare chaque forme binaire à un atome chimique. Le nombre de facteurs [...] dans une forme binaire est l'analogue de [...] la valence [...] d'un atome chimique. [...]

Chaque invariant et covariant devient ainsi exprimable par un graphe précisément identique à un diagramme kekuléen¹³, ou chemiographie.

Voici donc l'origine du mot *graphe* ! Sylvester l'emploiera de nouveau, sans le définir, dans un long article paru au même moment dans l'*American Journal of Mathematics*, qu'il venait de fonder [325]. Le terme se répand rapidement : la même année, Clifford¹⁴ commence à développer une *théorie des graphes*, mais il meurt prématurément l'année suivante sans avoir achevé ses travaux (qui seront malgré tout publiés à titre posthume en 1881 [93]) ; Spottiswoode¹⁵ écrit la même année un article intitulé *On Clifford's Graphs* [318], et Kempe¹⁶ [227] et Arthur Buchheim [60] mentionnent tous deux la *théorie des graphes* et les graphes de Clifford dans deux articles de 1885. Cependant de nombreux auteurs parlent encore de « diagramme » ou de « réseau », et c'est suite aux travaux de Petersen que le terme *graphe* s'imposera définitivement.

A.1.3 Petersen et la théorie des graphes réguliers

Les invariants d'une forme algébrique donnée génèrent un anneau, et une question qui intéressait les mathématiciens était de savoir si cet anneau

13. Friedrich Kekulé von Stradonitz (1829 – 1896), chimiste allemand célèbre pour la découverte de la tétravalence du carbone et la formule développée du benzène.

14. William Kingdon Clifford (1845 – 1879), mathématicien et philosophe anglais. Il est à l'origine, avec le mathématicien allemand Hermann Grassmann, de l'algèbre géométrique, un cas particulier d'algèbre de Clifford, nommées en son honneur.

15. William Spottiswoode (1825 – 1883), mathématicien et physicien britannique, qui fut notamment président de plusieurs sociétés scientifiques britanniques.

16. Alfred Bray Kempe (1849 – 1922), mathématicien britannique, élève de Cayley, principalement connu pour ses travaux sur le théorème des quatre couleurs.

a une base *finie*. Gordan¹⁷ prouva en 1868 que c'est effectivement le cas pour les invariants des formes binaires [173], mais sa démonstration fut jugée un peu compliquée. En 1885, il démontra un autre théorème [174] : étant données des variables x_1, x_2, \dots, x_n , on considère tous les produits de la forme

$$\prod_{i < j} (x_i - x_j)^{a_{ij}} \quad (\text{A.1})$$

où l'on suppose que toutes les variables ont même degré α ; alors on peut distinguer un nombre *fini* de tels produits, appelés *facteurs premiers*, tels que tous les autres peuvent être formés comme des produits de facteurs premiers. Par exemple, pour $n = 2$ ou 3 , il n'existe qu'un seul facteur premier, respectivement $(x_1 - x_2)$ et $(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)$; pour $n = 4$ il existe trois facteurs premiers : $(x_1 - x_2)(x_3 - x_4)$, $(x_1 - x_3)(x_2 - x_4)$ et $(x_1 - x_4)(x_2 - x_3)$.

En 1889, Hilbert¹⁸ utilise ce résultat pour simplifier la preuve de Gordan sur les formes binaires du théorème de la base finie¹⁹ [206].

En septembre 1889, alors qu'il rentre de Suède où il a reçu un traitement contre ses troubles oculaires, Sylvester fait la connaissance à Copenhague de Petersen²⁰, qui avait publié au début des années 1880 quelques résultats sur la théorie des invariants. Sylvester le met au courant de l'existence d'une preuve de Cayley qui, selon l'auteur, simplifie grandement celle de Hilbert. Mais Petersen s'aperçoit rapidement que cette « preuve » est erronée²¹, au grand étonnement de Sylvester qui refuse tout d'abord de l'admettre, n'ayant lui-même pas vu l'erreur.

Indépendamment dans un premier temps, Petersen et Sylvester examinent la prétendue preuve de Cayley et concentrent leur attention sur un élément central de l'argumentation de Hilbert, un système d'équations diophantiennes (justement ce que Cayley pensait pouvoir supprimer dans sa preuve). Par ailleurs, bien que les constructivistes reprochèrent à Hilbert d'avoir fourni une preuve du cas général purement existentielle, sa démonstration du cas binaire était, elle, constructive. Cependant, c'est le fait qu'il existe un nombre *fini* de solutions à ces équations qui intéressait Hilbert, et il ne chercha pas à expliciter ces solutions elles-mêmes. Petersen et Sylvester sont alors amenés à résoudre le même problème : déterminer les facteurs primitifs de ces équations, à partir desquels ils

17. Paul Gordan (1837 – 1912), mathématicien allemand, surnommé « le roi de la théorie des invariants ».

18. David Hilbert (1862 - 1943), célèbre mathématicien allemand, considéré comme l'un des plus grands du xx^e siècle.

19. En 1890, il généralisa ce résultat à toutes les formes algébriques [207], démontrant ainsi le théorème de la base finie, connu depuis sous le nom de *théorème de la base de Hilbert*. Cependant, sa preuve, non constructive, fut très critiquée, Gordan allant jusqu'à dire « Ce n'est pas des mathématiques, c'est de la théologie ! ».

20. Julius Petersen (1839 – 1910), mathématicien danois. On trouvera dans [258] une biographie très détaillée.

21. Cayley ne voulut pas reconnaître son erreur et insista pour que la preuve soit publiée. Cependant, bien que mentionnée, elle ne figure pas dans ses *Collected Papers*. Cayley avait d'ailleurs déjà commis une erreur en 1856, en prétendant que le cas binaire était faux à partir d'un certain ordre... Gordan prouva le contraire dans [173].

pourront déterminer toutes les autres solutions.

Petersen et Sylvester entament alors une correspondance soutenue et enthousiaste²², mais s'aperçoivent rapidement qu'ils suivent des approches différentes, auxquelles ni l'un ni l'autre ne veut renoncer. Une véritable course s'installe alors entre les deux mathématiciens pour aboutir à la décision, en décembre 1889, de publier deux articles séparés. Mais dès janvier 1890 Sylvester est, comme il l'écrit,

*incapable de tout travail de recherche mathématique en raison de mes problèmes aux yeux et certaines causes d'inquiétude mentale*²³

et ne semble plus vraiment comprendre ou vouloir faire l'effort de comprendre les résultats de Petersen, et renonce à la course. Il n'écrira plus aucun article en lien avec la théorie des invariants ou la théorie des graphes.

C'est donc seul que Petersen publie, en 1891, un article qui ne représente rien de moins que la naissance de la théorie des graphes telle que nous la connaissons aujourd'hui : *Die Theorie der regulären graphs*²⁴ [296]. Comme Petersen le reconnaît lui-même dans l'introduction, jamais cet article n'aurait pu voir le jour sans l'aide de Sylvester qui eut le premier l'idée de transformer le problème des facteurs primitifs en un problème de théorie des graphes. Petersen explique ensuite cette transformation : pour chaque x_i il dessine un point dans le plan, et lorsque le terme $(x_i - x_j)$ intervient à la puissance k , il relie les points x_i et x_j par k lignes distinctes. Comme cela a été dit plus haut, toutes les variables ont même *degré* et Petersen obtient donc ainsi un graphe régulier (c'est-à-dire où tous les sommets ont même *degré*²⁵). Avec ce formalisme, factoriser le polynôme (A.1) en deux polynômes homogènes de degrés β et γ revient à partitionner les arêtes du graphe en deux sous-graphes couvrants réguliers de degrés β et γ , appelés respectivement β -facteur et γ -facteur. Un graphe régulier qui n'admet pas une telle décomposition est logiquement qualifié de *premier*²⁶. De cette façon, le problème de la détermination des solutions des équations diophantiennes de Hilbert se ramène à la détermination de tous les graphes premiers à n sommets, et pour tout graphe régulier à trouver toutes ses factorisations en graphes premiers.

Détailler la suite de l'article de Petersen dépasserait largement le cadre de cette thèse (nous encourageons le lecteur curieux à se reporter à l'article de Mulder [288], dans lequel il pourra trouver une analyse des résultats de Petersen et des développements qui ont eu lieu au cours des décennies suivantes). Notons simplement qu'il contient les prémisses de ce qui de-

22. Toutes les lettres de Sylvester envoyées à Petersen ont été conservées, mais on n'a pas de trace de celles envoyées par Petersen. L'histoire de cette correspondance est détaillée dans [308].

23. C'est-à-dire de dépression.

24. *La théorie des graphes réguliers*. Notons que Petersen n'écrit pas *Graphen*, mais *graphs* pour souligner l'emprunt aux « auteurs anglais qui ont introduit le mot *graph* pour décrire de telles figures. »

25. Les mots *ordre*, *degré* ou *facteur* utilisés encore aujourd'hui en théorie des graphes proviennent donc directement de la théorie des invariants !

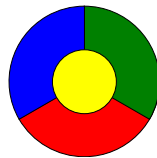
26. Nous donnerons dans la section [??] une autre définition des *graphes premiers*.

viendra la théorie des *couplages* (un *1*-facteur n'est rien d'autre qu'un couplage parfait !) et qu'on y retrouve même la notion de chaîne augmentante permettant de déterminer un couplage de cardinalité maximale. Quelques années plus tard, Petersen utilisera ces résultats pour invalider la preuve d'une conjecture qui a résisté plus de 120 ans aux mathématiciens : le théorème des quatre couleurs.

A.1.4 Guthrie et le problème des quatre couleurs

En 1852, alors qu'il essaie de colorier les comtés d'une carte d'Angleterre de sorte que deux comtés voisins aient une couleur différente, Francis Guthrie²⁷ remarque qu'il peut y parvenir avec seulement quatre couleurs. Il tente de démontrer qu'il en est de même pour n'importe quelle carte, et bien qu'il pense y être parvenu, il est insatisfait de sa preuve et en parle à son frère Frederick, étudiant en physique, qui en fait part à son tour à l'un de ses professeurs, De Morgan²⁸, dont Francis avait été lui-même l'élève. La première trace écrite que l'ont ait de ce problème est justement une lettre de De Morgan, datée du 23 octobre 1852, adressée à Hamilton²⁹ :

Un de mes étudiants m'a demandé aujourd'hui de lui justifier un fait dont je ne savais pas que c'était un fait - et je ne le sais toujours pas. Il dit que si on divise une figure de quelque manière que ce soit, et qu'on colorie les compartiments de façon que les parties ayant une frontière commune soient de couleurs différentes, on peut avoir besoin de quatre couleurs, comme dans l'exemple ci-dessous, mais jamais de plus.



Un exemple nécessitant cinq couleurs ou plus semble ne pas pouvoir être inventé. Car à ce que j'en vois pour l'instant, si on se ramène à quatre compartiments partageant deux à deux un bout de frontière commune, alors trois d'entre eux entourent complètement le quatrième, lui interdisant toute connexion avec un cinquième. Si ceci est vrai, quatre couleurs suffiront pour toute carte, sans qu'on ait besoin de colorier identiquement deux pays qui ont plus d'un point commun. [...] Et, si cela est vrai, est-ce déjà connu ? Mon étudiant dit qu'il l'a deviné en coloriant une carte d'Angleterre. Plus j'y pense, plus cela me paraît évident. Si votre réponse consiste en un exemple très simple qui me met au rang d'un animal stupide, je pense que je devrai agir comme le fit le Sphinx. . .

27. Francis Guthrie (1831 - 1899), mathématicien et botaniste sud-africain.

28. Augustus De Morgan (1806 - 1871), mathématicien et logicien britannique, ayant notamment formulé les *lois de De Morgan*. Il fut l'un des professeurs de Sylvester.

29. William Hamilton (1805 - 1865), mathématicien, physicien et astronome irlandais ; il est principalement connu pour sa découverte des *quaternions*.

Mais Hamilton est bien trop occupé à cette période à publier ses résultats sur les quaternions. Sa réponse est donc immédiate (datée du 26 octobre 1852) et montre bien son manque d'intérêt pour la question de De Morgan :

Il est peu vraisemblable que je réfléchisse à votre « quaternion » de couleurs.

La première trace du problème des quatre couleurs dans un journal semble remonter à 1860 dans la revue *Athenaeum*, et bien que le nom de l'auteur de l'article soit absent, « on est presque sûr qu'il s'agisse de De Morgan » [73, p. 5]. C'est ainsi qu'outre-Atlantique on prit connaissance de ce problème auquel personne n'avait encore apporté de solution. En particulier, le philosophe américain Charles Sanders Peirce montra un vif intérêt et prouva par exemple qu'une carte dessinée sur un tore peut nécessiter six couleurs (en réalité, il existe même une configuration imposant l'utilisation de sept couleurs).

En revanche, en Grande-Bretagne le problème tombe un peu dans l'oubli pendant 25 ans, jusqu'à ce que Cayley le découvre à son tour et lui donne une existence « officielle » en l'exposant à la communauté scientifiques, la première fois le 13 juin 1878 à la *London Mathematical Society*, puis en avril 1879 à la *Royal Geographical Society* [66]. Pour l'anecdote, Cayley écrivit :

Le théorème qui dit que quatre couleurs suffisent pour toute carte est mentionné quelque part par le Professeur De Morgan.

et quand on retrouva la lettre de De Morgan adressée à Hamilton, on ignorait le nom de cet étudiant qui avait posé le problème. Ce ne fut qu'en 1880 que son identité fut redécouverte, dans une note de Frederick Guthrie, devenu physicien, adressée à la *Royal Society of Edinburgh* :

Il y a environ trente ans, alors que je suivais les cours du Professeur De Morgan, mon frère, Francis Guthrie, qui venait d'en sortir (et qui est maintenant professeur de mathématiques à l'Université Sud-Africaine, au Cap), me fit remarquer que le plus grand nombre de couleurs nécessaires au coloriage d'une carte en évitant de colorier pareillement deux régions linéairement frontalières est quatre. Il ne serait guère sérieux, après si longtemps, de tenter de restituer ici sa démonstration, mais la figure cruciale était la suivante [...]. Avec la permission de mon frère, je soumis le théorème au professeur De Morgan, qui eut beaucoup de plaisir et l'accepta comme un résultat nouveau.

Francis Guthrie devint un éminent botaniste (plusieurs espèces de finbos, une plante sud-africaine, portent son nom), et mourut en 1899, sans avoir jamais rien publié sur le théorème des quatre couleurs.

Le 17 juillet 1879, un avocat anglais et ancien élève de Cayley, Kempe³⁰, qui était présent à l'exposé de son professeur en 1878, publie une démonstration du théorème et apporte même plusieurs améliorations qui sont

30. Il est troublant de constater que Sylvester, Cayley et Kempe ont été tous trois avocats avant de se tourner vers les mathématiques !

publiées dans diverses revues ; ce travail le rend célèbre et il obtient même une décoration. Kempe remarque notamment que si on place une feuille de papier calque sur une carte, qu'on marque un point dans chaque région et qu'on relie ces points par des traits, on obtient le diagramme d'un « mécanisme articulé » ; le problème de la coloration des cartes revient alors à donner une couleur à chaque point de sorte que deux points reliés sur le diagramme soient de couleur différente. Sans le savoir, Kempe avait donc modélisé le problème en termes de graphe planaire (un graphe est *planaire* s'il peut être dessiné dans le plan de sorte qu'aucune arête n'en croise une autre)! Une autre démonstration, « améliorée », est présentée 15 mars 1880 à la Royal Society of Edinburgh par Tait³¹ [326]. Malheureusement, chacune de ces preuves sera invalidée onze ans après sa parution.

Tout d'abord, Heawood³² démontre en 1890 qu'un argument dans la preuve de Kempe est erroné (bien que les concepts développés dans cette preuve n'en soient pas moins importants et ouvriront la route vers une preuve correcte) [197]. Il apporte en effet un contre-exemple à la méthode des *chaînes de Kempe* en exhibant une carte de 25 régions, qu'on connaît aujourd'hui sous le nom de *graphe 4-chromatique de Heawood*³³. Même si la preuve est incorrecte, Heawood montre qu'elle permet d'établir un *théorème des cinq couleurs*. Malgré de nombreuses autres contributions au problème du coloriage (il continuera à publier des articles en relation avec le problème des quatre couleurs jusqu'en 1949!), notamment son extension à des cartes dessinées sur des surfaces autres que la sphère, la renommée de Heawood n'égalera jamais celle de Kempe, et s'il finit lui aussi par obtenir une décoration, c'est pour sa participation active à la récolte de fonds pour la restauration d'un château. . .

Quant à la démonstration de Tait, Petersen montre implicitement en 1891 dans sa *Theorie der regularen graphs* qu'elle est incomplète. Cependant, elle contient des idées qui initieront l'étude de l'arête-coloration des graphes. On y trouve en effet le résultat suivant :

*La conjecture des quatre couleurs est vraie si et seulement si tout graphe cubique et sans isthme est 3-arête-colorable.*³⁴

Notons que dans cet énoncé, Tait fait implicitement référence aux graphes cubiques sans isthme *planaires*. En effet, si l'on omet cette condition, on peut trouver des contre-exemples, c'est-à-dire des graphes cubiques sans isthme qui ne sont pas 3-arête-colorables, le plus célèbre étant sans doute

31. Peter Guthrie Tait (1831 - 1901) physicien et mathématicien écossais.

32. Percy John Heawood (1861 - 1955), mathématicien britannique. Il est l'auteur de la *conjecture de Heawood*, qui donne une borne sur le nombre de couleurs nécessaires pour colorier un graphe sur une surface de genre donnée, et qui fut démontrée en 1968.

33. Les contre-exemples à la méthode des chaînes de Kempe ont continué d'intéresser les mathématiciens. Ainsi, en 1896, Vallée Poussin a donné un contre-exemple à 15 sommets (le *graphe de Poussin*), Errera un graphe à 17 sommets (le *graphe d'Errera*) [126], Kittell un graphe à 23 sommets (le *graphe de Kittell* [231]); enfin, deux contre-exemples minimaux ont été construits : le *graphe de Soifer* en 1997 [317] et le *graphe de Fritsch* en 1998 [140], tous deux d'ordre 9.

34. Un graphe *cubique* est un graphe 3-régulier ; un *isthme* est une arête telle que sa suppression déconnecte le graphe.

celui donné par Petersen en 1898³⁵ dans *L'intermédiaire des mathématiciens*³⁶ [297] :

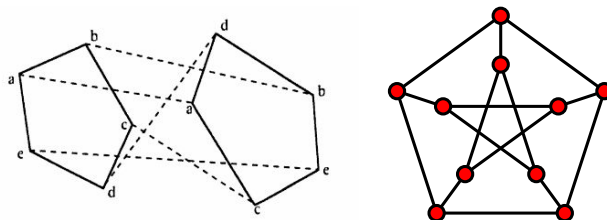


FIGURE A.3 – Le graphe de Petersen, dans sa forme originale et tel qu'on le représente aujourd'hui.

Quelques avancées ont lieu pendant la première moitié du xx^e siècle. Puis avec les progrès de l'informatique, le mathématicien allemand Heinrich Heesch (1906 - 1995) s'intéresse à la possibilité de prouver la conjecture des quatre couleurs à l'aide d'un ordinateur et invente la technique de *décharge*. Il se rend plusieurs fois aux États-Unis où il travaille avec Wolfgang Haken; malheureusement, son financement est interrompu à un moment crucial, et ce sont finalement Haken et Kenneth Appel qui annoncent, en 1976, avoir démontré le théorème des quatre couleurs. Leur démonstration partage la communauté scientifique, car c'est la première fois qu'un ordinateur est nécessaire (il faut en effet étudier 1478 configurations, ce qui représente à l'époque 1200 heures de calcul!). Depuis, leur algorithme a été simplifié et une nouvelle preuve a été publiée en 1996 par Robertson, Sanders, Seymour et Thomas, qui ne fait intervenir « que » 633 configurations [304].

L'une des maximes favorites du grand mathématicien Paul Erdős³⁷ était qu'« il faut parfois compliquer un problème pour en simplifier la solution. » Selon lui, cette maxime s'appliquait au problème des quatre couleurs, « un problème subtil et non pas un problème complexe », et une démonstration simple, et même très simple, devait exister. Mais pour cela, il faudrait peut-être formuler le problème pour un ensemble de points plus vaste qu'un graphe planaire, et incluant celui-ci. On ne dispose toujours pas, en 2011, d'une preuve du théorème des quatre couleurs ne nécessitant pas d'ordinateur.

A.1.5 Berge et les graphes parfaits

Enfin, on ne saurait clore cet historique sans évoquer le nom de Claude Berge ni sa théorie des graphes parfaits, tant ils ont été un moteur pour le développement de la théorie des graphes depuis ces cinquante dernières

35. En fait, Kempe avait déjà mentionné ce graphe en 1886, en relation avec un problème de géométrie, en tant que *configuration de Desargues* [228].

36. Ce journal, fondé par Charles-Ange Laisant en 1894, avait pour but de permettre aux mathématiciens de déposer des « questions », auxquelles d'autres mathématiciens pouvaient ensuite publier une réponse.

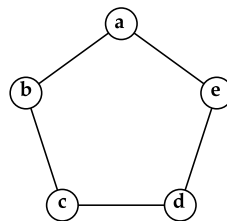
37. Paul Erdős (1913 - 1996), mathématicien hongrois célèbre pour le nombre de ses publications et de ses collaborateurs.

années.

Claude Berge est un mathématicien français né en 1926³⁸. Après un bref passage par la topologie (il a étendu un théorème de topologie dans l'espace de Banach et publiera un ouvrage en 1959), il se tourne vers la théorie des jeux sur les graphes, qu'on appelle souvent *jeux de Nim*; puis, le sujet étant vite limité, il commence à s'intéresser à d'autres problèmes sur les graphes, et en particulier à un problème lié à la théorie de l'information, que voici³⁹.

Lorsqu'on transmet un message, sous forme d'une suite de symboles, à travers un canal, il est soumis aux interférences et aux imperfections physiques de ce canal, et on n'obtient pas toujours en sortie une copie exacte de ce que l'on a envoyé. On dit ainsi que deux symboles sont *confondables* s'ils peuvent potentiellement donner le même symbole en sortie, et on généralise cette notion à des messages de longueur quelconques. Shannon⁴⁰ s'est intéressé dans [313] à déterminer, dans un alphabet donné, un ensemble de symboles deux à deux non confondables, permettant ainsi de s'assurer qu'on pourra retrouver le message d'origine lors de la réception.

Considérons par exemple l'alphabet $\Sigma = \{a, b, c, d, e\}$ tel que les paires de symboles confondables soient $\{ab, bc, cd, de, ea\}$ (c'est-à-dire que a peut être confondu avec b et e , b peut être confondu avec a et c , etc.). Si l'on assimile à chaque symbole un sommet, et qu'on relie deux sommets s'ils correspondent à des symboles confondables, on obtient le graphe G suivant :



Par conséquent, un ensemble de symboles deux à deux non confondables correspond à un ensemble de sommets deux à deux non adjacents, ou *ensemble stable*; le nombre maximum de symboles que l'on peut ainsi prendre correspond au *nombre de stabilité*⁴¹ du graphe, noté $\alpha(G)$; dans cet exemple, on a donc $\alpha(G) = 2$ (en n'utilisant par exemple que les symboles a et c). Il est ainsi possible de transmettre $\alpha(G)^m$ mots de m symboles sans

38. Pour l'anecdote, il était l'arrière-petit-fils du président de la république Félix Faure. Une photo assez célèbre le montre d'ailleurs posant devant un portrait de son aïeul.

39. Les informations qui suivent sont issues de [24], [25], et [300].

40. Claude Shannon (1916 - 2001), ingénieur et mathématicien américain, considéré comme le fondateur de la théorie de l'information. Ses travaux ont été et continuent d'être d'une très grande importance dans tous les domaines des sciences de l'information et de la communication, de la sociologie aux télécommunications.

41. Le mathématicien et physicien John Von Neumann l'appelait *coefficient de stabilité interne*, par opposition au coefficient de stabilité externe que l'on appelle aujourd'hui *nombre de domination* du graphe.

risquer d'en confondre deux lors de la réception.

Toutefois, en étant un peu astucieux, il est possible de faire mieux, en découpant le message à transmettre non pas en symboles, mais en *groupes* de symboles. Considérons par exemple l'ensemble de mots, aussi appelé *code*, $\{aa, bc, ce, db, ed\}$ défini sur l'alphabet précédent. Il est facile de vérifier que ces cinq mots sont deux à deux non confondables (par exemple, aa et bc ne sont pas confondables, car bien que a puisse être confondu avec b , il ne peut pas l'être avec c). Ainsi, en utilisant ces cinq mots de code plutôt que des symboles individuels, on peut former $5^{\lfloor \frac{m}{2} \rfloor}$ messages de m symboles, une amélioration sensible par rapport à 2^m ; et plus généralement, avec un code constitué de C mots de longueur n , on peut transmettre sans risque d'erreur $|C|^{\lfloor \frac{m}{n} \rfloor} = 2^{\lfloor \frac{m}{n} \rfloor \log_2 |C|} \approx 2^{\frac{m}{n} \log_2 |C|}$ messages à m symboles. En dessinant comme ci-dessus le graphe G_n des mots confondables de longueur n (qu'on note plus simplement G au lieu de G_1 si $n = 1$), il vient que la taille maximale d'un tel code, notée $N(n)$, est égale à $\alpha(G_n)$; et il est facile de voir que $G_n = G^n$, où G^n désigne le *produit normal*⁴² de n copies de G . Ceci a conduit Shannon à définir la *capacité zéro-erreur* d'un canal de transmission comme étant

$$C_0(G) = \lim_{n \rightarrow \infty} \frac{1}{n} \log \alpha(G^n).$$

Déterminer cette limite est relativement difficile, et même le cas le plus simple, le cycle à 5 sommets (noté C_5) décrit précédemment, a résisté plus de vingt ans, jusqu'à ce que Lovász montre en 1979 que $C_0(C_5) = \frac{1}{2} \log 5$, grâce à un nouvel outil important, la *fonction theta*⁴³ [250].

De plus, il est facile de voir que $\alpha(G^n) \geq \alpha(G)^n$ puisque pour stable S de G , S^n est un stable de G^n . Par conséquent, $C_0(G) \geq \log \alpha(G)$. Le problème de Shannon était de déterminer les graphes qui satisfont $C_0(G) = \log(\alpha(G))$.

Si l'on désigne par $\theta(G)$ le nombre minimum de cliques⁴⁴ pour partitionner les sommets de G , on a clairement $\alpha(G) \leq \theta(G)$, et il est facile de vérifier que $\theta(G^n) \leq \theta(G)^n$. Ainsi, on a

$$\log \alpha(G) \leq C_0(G) \leq \log \theta(G)$$

et par conséquent, si $\alpha(G) = \theta(G)$, $C_0(G) = \log \alpha(G)$. Etant liés à la capacité zéro-erreur de Shannon, Berge nomme *graphes α -parfaits* les graphes qui vérifient la « belle propriété », c'est-à-dire $\alpha(H) = \theta(H)$ pour tout sous-graphe induit H de G (sans cette restriction, il est facile de construire des graphes vérifiant $\alpha = \theta$).

En plus de la classe G_1 des graphes α -parfaits, les recherches de Berge le conduisent à ce moment-là à considérer deux autres classes de graphes :

42. Le produit normal (ou *produit fort*) de deux graphes $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ a pour ensemble de sommets $V_1 \times V_2$ et (u_1, u_2) est adjacent à (v_1, v_2) si et seulement si $(u_1 = v_1 \text{ et } u_2 \sim v_2)$ ou $(u_1 \sim v_1 \text{ et } u_2 = v_2)$ ou $(u_1 \sim v_1 \text{ et } u_2 \sim v_2)$, où $u_1 \sim v_1$ signifie « u_1 est adjacent à v_1 ».

43. Appelé aussi *nombre de Lovász*.

44. Une *clique* est un graphe dont les sommets sont deux à deux adjacents.

la classe G_2 des graphes vérifiant $\chi(H) = \omega(H)$ ⁴⁵ pour tout sous-graphe induit H de G (ce sont les graphes χ -parfaits), et la classe G_3 des graphes *sans trou impair ni antitrou impair* ⁴⁶, appelés aussi *graphes de Berge*.

Comme un stable dans G est une clique dans le complémentaire \overline{G} de G , et qu'une coloration de G est une partition en cliques dans son complémentaire, on a le résultat suivant :

$$G \in G_1 \Leftrightarrow \overline{G} \in G_2.$$

Berge conjecture dès 1960 que les deux classes G_1 et G_2 sont en fait les mêmes. Cette conjecture, connue sous le nom de *conjecture faible des graphes parfaits*, fut démontrée en 1972 par Lovász [249]; les graphes α -parfaits et χ -parfaits devinrent les *graphes parfaits* et la conjecture devint le *théorème des graphes parfaits* :

$$G \text{ est parfait} \Leftrightarrow \overline{G} \text{ est parfait.}$$

S'il y avait une *conjecture faible*, c'est que Berge avait également formulé une *conjecture forte* des graphes parfaits, qui impliquait la première : *les classes G_1 (et donc G_2) et G_3 sont équivalentes*. En peu de temps, les travaux de Berge ont déclenché un engouement extraordinaire, comme en atteste cet extrait de son discours inaugural aux Journées internationales d'études de la théorie des graphes (1966) :

Je remercie les deux cent cinquante participants de ce congrès venus si nombreux à Rome pour un sujet qui, il y a dix ans seulement, n'aurait attiré qu'une dizaine de personnes. Étrange évolution que celle de la théorie des graphes, qui se développe par à-coups, sous l'impulsion tour à tour du rôle de l'électricité, de la géométrie des polyèdres, de la théorie des jeux, et surtout maintenant de la recherche opérationnelle et du calcul électronique.

et plusieurs centaines d'articles ont été publiés sur les graphes parfaits. En particulier, Grötschel, Lovász et Schrijver ont montré que certains problèmes que l'on savait difficiles ⁴⁷ à résoudre en général sont plus faciles dans les graphes parfaits [178]; cependant, leur méthode repose sur l'utilisation d'ellipsoïdes, une technique employée en programmation linéaire, et n'est pas combinatoire. Enfin, en 2002, Chudnovsky, Robertson, Seymour et Thomas ont annoncé avoir démontré ce que l'on appelle désormais le *théorème fort des graphes parfaits* [77]. L'intérêt pour les graphes parfaits n'est pas retombé pour autant, bien au contraire! La recherche d'algorithmes combinatoires et d'une preuve simplifiée du théorème fort est aujourd'hui très active.

Claude Berge est mort le 30 juin 2002, quelques semaines après l'annonce de la démonstration de sa conjecture forte. Touche-à-tout, il était

45. $\chi(G)$ est le nombre minimum de stables nécessaires pour partitionner les sommets de G (on parle aussi de *coloration*); $\omega(G)$ désigne la taille de la plus grande clique de G .

46. Un *trou impair* est un cycle (sans corde) comprenant un nombre impair de sommets ≥ 5 ; un *antitrou* est le complémentaire d'un trou.

47. Nous donnerons dans la section 1.2 un sens plus précis à ce mot.

également sculpteur, collectionneur et homme de lettres (il était notamment co-fondateur aux côtés de Raymond Queneau et Georges Péric de l'OuLiPo (Ouvroir de Littérature Potentielle), un groupe de littéraires et mathématiciens). Sa plus célèbre nouvelle, *Qui a tué le Duc de Densmore ?* est une énigme policière dont la résolution repose sur le théorème de Ha-jós, un théorème de théorie des graphes, évidemment. . .

A.2 HISTOIRE DE LA THÉORIE DE LA COMPLEXITÉ

La théorie de la complexité est une théorie jeune, beaucoup plus que la théorie des graphes⁴⁸ puisque ses origines remontent à la fin des années 60, bien que l'on trouve déjà des notions d'*algorithmes efficaces* chez les Grecs anciens, ou encore chez Lamé en 1844 (cf. [311, p.50]) et qu'elle repose sur les travaux de Turing⁴⁹ en 1937 [334], [335]. Dans [135], Fortnow et Homer font coïncider l'origine de la théorie de la complexité avec la publication en 1965 d'un article de Hartmanis et Stearns intitulé *On the Computational Complexity of Algorithms*, dans lequel on trouve les premières définitions de complexité temporelle et spatiales (exprimées à l'aide de machines de Turing⁵⁰). En 1965 aussi, Edmonds [121] a explicité un algorithme en temps polynomial pour le problème du couplage maximum (cf. section 2.1.4), et affirmé qu'il s'agit d'une bonne formalisation de la notion de calcul « efficace ». Ceci est annonciateur de la fameuse question $\mathcal{P} = \mathcal{NP} ?$, sans doute le plus célèbre problème en informatique théorique ; ses implications sont telles qu'il figure sur la liste des *sept problèmes du millénaire* du Clay Mathematics Institute, chacun récompensé d'un prix d'un million de dollars pour sa résolution !

Une étape cruciale dans l'histoire de la théorie de la complexité eut lieu en 1971 quand Cook [95] aux États-Unis, et indépendamment Levin [246]⁵¹ en U.R.S.S., apportèrent la preuve de l'existence de problèmes dits *\mathcal{NP} -complets*, un résultat connu aujourd'hui sous le nom de *théorème de Cook-Levin*. Comme nous le verrons dans la section suivante, les problèmes *\mathcal{NP} -complets* sont d'une certaine manière les problèmes les plus difficiles que nous rencontrerons par la suite.

Le premier problème à avoir été démontré *\mathcal{NP} -complet* est SAT ou *problème de satisfaisabilité*. En logique propositionnelle, une variable ne peut prendre que deux valeurs : soit *vrai*, soit *faux* (c'est le principe du *tiers exclus*) ; on peut également appliquer à toute variable un opérateur de négation, « NON », noté \neg . On appelle alors *littéral* toute variable ou sa négation. Une *clause* est un ensemble de littéraux séparés par l'opérateur logique « OU », noté \vee . Enfin, une formule logique est sous *forme normale conjonctive* lorsqu'elle est écrite sous forme d'une conjonction de

48. Cet historique sera donc beaucoup plus succinct que celui de la théorie des graphes !

49. Alan Turing (1912-1954), mathématicien britannique considéré comme le fondateur de l'informatique.

50. Une machine de Turing est un modèle abstrait du fonctionnement des appareils mécaniques de calcul, tel un ordinateur et sa mémoire, imaginé par Turing.

51. Bien que publiés seulement en 1973, les travaux de Levin sont cités dans des séminaires et soumis pour publication quelques années auparavant.

clauses, c'est-à-dire sous la forme d'un ensemble de clauses séparées par l'opérateur logique « ET », noté \wedge . Le problème SAT consiste à déterminer s'il est possible d'affecter la valeur *vrai* ou *faux* à chacune des variables d'une formule écrite sous forme normale conjonctive, de sorte que ladite formule soit *satisfaite*, c'est-à-dire vraie.

Les implications du théorème de Cook-Levin sont extrêmement importantes. Tout d'abord, il signifie que tous les problèmes d'une classe appelée \mathcal{NP} (qui contient les problèmes \mathcal{NP} -complets) peuvent se ramener à SAT au moyen d'une transformation appelée *réduction polynomiale*. Grâce à cette transformation, Karp [226] a pu prouver la \mathcal{NP} -complétude de 21 problèmes réputés difficiles de combinatoire et théorie des graphes, connus aujourd'hui comme les *21 problèmes de Karp*. Depuis, on a recensé plusieurs milliers de problèmes \mathcal{NP} -complets; les plus célèbres et importants d'entre eux sont recensés dans l'ouvrage de référence de Garey et Johnson [150]. Les problèmes \mathcal{NP} -complets sont donc plus ou moins tous équivalents entre eux. Ainsi, une conséquence fondamentale du théorème de Cook-Levin est que si on arrive à mettre au point un algorithme « efficace » pour résoudre l'un d'entre eux, tous les autres pourront aussi être résolus efficacement. Le problème $\mathcal{P} = \mathcal{NP}$? que nous avons évoqué plus haut consiste justement à déterminer si un tel algorithme existe. Aujourd'hui, la très grande majorité de la communauté s'accorde à penser que ce n'est pas le cas, i.e. $\mathcal{P} \neq \mathcal{NP}$. Par conséquent, lorsque la \mathcal{NP} -complétude d'un problème est prouvée, les chercheurs se tournent vers d'autres méthodes pour le résoudre, par exemple les algorithmes d'approximation (qui comme leur nom l'indique donnent une solution approchée), les méthodes probabilistes ou encore les heuristiques.

Des « preuves » confirmant ou infirmant que $\mathcal{P} = \mathcal{NP}$ sont régulièrement annoncées (le site web <http://www.win.tue.nl/~gwoegi/P-versus-NP.htm> en recensait 72 en juin 2011 !), mais aucun de ces résultats n'est reconnu par la communauté. Néanmoins, en août 2010, un chercheur américain des laboratoires HP, Vinay Deolalikar, a publié sur l'Internet un article de 120 pages intitulé $\mathcal{P} \neq \mathcal{NP}$, présenté comme une démonstration de ce célèbre problème, et qui a suscité un grand intérêt jusque dans la presse; Cook a même reconnu, au vu de l'article, qu'il s'agissait d'une déclaration à prendre très au sérieux. Mais quelques semaines plus tard, plusieurs erreurs fatales ont été découvertes. . . La théorie de la complexité est donc un domaine extrêmement actif et tout laisse penser qu'il reste encore de nombreuses choses à y découvrir.

BIBLIOGRAPHIE

- [1] R.K. AGARWAL : An Investigation of Subgraph Isomorphism Problem. Mémoire de D.E.A., University of Toronto, Dept. of Computer Science, 1980. (Cité page 80.)
- [2] V.E. ALEKSEEV : Hereditary classes and coding of graphs (en russe). *Problems of Cybernetics*, 39:151–164, 1982. (Cité page 17.)
- [3] V.E. ALEKSEEV : On the local restrictions effect on the complexity of finding the graph independence number. *Combinatorial-Algebraic Methods in Applied Mathematics, Gorkiy Univ. Press, Gorky*, pages 3–13, 1983. (Cité pages 49 et 92.)
- [4] V.E. ALEKSEEV : On the number of maximal independent sets in graphs from hereditary classes (en russe). *Combinatorial-Algebraic Methods in Discrete Optimization, Nizhny Novgorod University Publishers, Nizhniy Novgorod*, 6:5–8, 1991. (Cité pages 69 et 104.)
- [5] V.E. ALEKSEEV : A polynomial algorithm for the independent set problem in fork-free graphs (en russe). *Discrete Anal. Oper. Res*, 6 (4):3–19, 1999. (traduction anglaise : voir [7]). (Cité pages 51, 64 et 65.)
- [6] V.E. ALEKSEEV : On easy and hard hereditary classes of graphs with respect to the independent set problem. *Discrete Applied Mathematics*, 132(1-3):17–26, 2003. (Cité pages 50 et 101.)
- [7] V.E. ALEKSEEV : Polynomial algorithm for finding the largest independent sets in graphs without forks. *Discrete Applied Mathematics*, 135(1-3), 2004. ISSN 0166-218X. (Cité pages 51 et 185.)
- [8] V.E. ALEKSEEV, R. BOLIAC, D.V. KOROBITSYN et V.V. LOZIN : \mathcal{NP} -hard graph problems and boundary classes of graphs. *Theoretical Computer Science*, 389(1-2):219–236, 2007. (Cité page 50.)
- [9] V.E. ALEKSEEV et D.V. KOROBITSYN : Complexity of some problems on hereditary classes of graphs. *Diskretnaya Matematika*, 4(4):34–40, 1992. (Cité page 50.)
- [10] V.E. ALEKSEEV et V.V. LOZIN : On local transformations of graphs that preserve the independence number (en russe). *Diskretnyi Analiz i Issledovanie Operatsii*, 5(1):3–19, 1998. (Cité page 96.)
- [11] V.E. ALEKSEEV et V.V. LOZIN : Augmenting graphs for independent sets. *Discrete Applied Mathematics*, 145(1):3–10, 2004. (Cité pages 60, 64, 65 et 106.)

- [12] V.E. ALEKSEEV et V.V. LOZIN : Local transformations of graphs preserving independence number. *Discrete applied mathematics*, 135(1-3):17–30, 2004. (Cité page 96.)
- [13] G. ALEXE, P.L. HAMMER, V.V. LOZIN et D. de WERRA : Struction revisited. *Discrete applied mathematics*, 132(1-3):27–46, 2003. (Cité page 93.)
- [14] C. ARBIB et R. MOSCA : The maximum stable set problem for particular graphs with no induced P_5 . Rapport technique, Tech. Report 79, Univ. di L'Aquila Dipartimento di Matematica Pura e Applieata, 1995. (Cité page 68.)
- [15] C. ARBIB et R. MOSCA : On $(P_5, \text{diamond})$ -free graphs. *Discrete mathematics*, 250(1-3):1–22, 2002. (Cité pages 83 et 96.)
- [16] J.C. ARDITTI et D. de WERRA : A note on a paper by D. Seinsche. *Journal of Combinatorial Theory, Series B*, 21:90, 1976. (Cité page 77.)
- [17] G. BACSÓ et Z. TUZA : A characterization of graphs without long induced paths. *Journal of graph theory*, 14(4):455–464, 1990. (Cité page 54.)
- [18] G. BACSÓ et Z. TUZA : Dominating cliques in P_5 -free graphs. *Periodica Mathematica Hungarica*, 21(4):303–308, 1990. (Cité pages 52 et 54.)
- [19] G. BACSÓ et Z. TUZA : Graphs without induced P_5 and C_5 . *Discusiones Mathematicae Graph Theory*, 24(3):503–508, 2004. (Cité page 52.)
- [20] M.L. BALINSKI : Labelling to obtain a maximum matching. In *Combinatorial Mathematics and Its Applications (Proceedings Conference Chapel Hill, North Carolina, 1967 ; RC Bose, TA Dowling, eds.)*, The University of North Carolina Press, Chapel Hill, North Carolina, pages 585–602, 1969. (Cité page 47.)
- [21] W.W.R. BALL : Mathematical Recreations and Essays. *Bull. Amer. Math. Soc.*, 2:37–46, 1892. URL <http://www.gutenberg.org/files/26839/26839-pdf.pdf>. (Cité page 170.)
- [22] L.W. BEINEKE : Characterizations of derived graphs. *Journal of Combinatorial Theory*, 9(2):129–135, 1970. (Cité page 12.)
- [23] C. BERGE : Two theorems in graph theory. *Proceedings of the National Academy of Sciences of the United States of America*, 43(9):842–844, 1957. (Cité page 46.)
- [24] C. BERGE : *Graphes et hypergraphes*. Dunod, Paris, 1970. (Cité pages 16 et 179.)
- [25] C. BERGE : Motivations and history of some of my conjectures. *Discrete Mathematics*, 165(15):70, 1997. (Cité page 179.)
- [26] A. BERRY, M.C. GOLUBIC et M. LIPSHTEYN : Two tricks to triangulate chordal probe graphs in polynomial time. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages

- 992–969. Society for Industrial and Applied Mathematics, 2004. (Cité page 155.)
- [27] A. BERRY, M.C. GOLUMBIC et M. LIPSHTEYN : Recognizing chordal probe graphs and cycle-bicolorable graphs. *SIAM Journal on Discrete Mathematics*, 21(3):573–591, 2008. (Cité page 155.)
- [28] A.A. BERTOSSI : Dominating sets for split and bipartite graphs. *Information Processing Letters*, 19(1):37–40, 1984. (Cité page 105.)
- [29] N. BLUM : A new approach to maximum matching in general graphs. *Automata, Languages and Programming*, pages 586–597, 1990. (Cité page 47.)
- [30] H. BODLAENDER, A. BRANDSTÄDT, D. KRATSCH, M. RAO et J. SPINRAD : Linear Time Algorithms for Some \mathcal{NP} -Complete Problems on (P_5, Gem) -Free Graphs (extended abstract). In *Fundamentals of Computation Theory*, Lecture Notes in Computer Science, volume 2751, pages 61–72. Springer, 2003. (Cité page 84.)
- [31] H.L. BODLAENDER, A. BRANDSTÄDT, D. KRATSCH, M. RAO et J. SPINRAD : On algorithms for (P_5, gem) -free graphs. *Theoretical Computer Science*, 349(1):2–21, 2005. (Cité page 84.)
- [32] R. BOLIAC et V.V. LOZIN : An augmenting graph approach to the stable set problem in P_5 -free graphs. *Discrete Applied Mathematics*, 131(3):567–575, 2003. (Cité pages 65 et 70.)
- [33] R. BOLIAC et V.V. LOZIN : Independent domination in finitely defined classes of graphs. *Theoretical Computer Science*, 301(1-3):271–284, 2003. (Cité pages 63 et 105.)
- [34] I. BOMZE, M. BUDINICH, P. PARDALOS et M. PELILLO : *Handbook of Combinatorial Optimization*, volume 4, chapitre The maximum clique problem, pages 1–74. Kluwer Academic Publishers, Boston, MA, 1999. (Cité page 38.)
- [35] K.S. BOOTH et G.S. LUEKER : Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976. (Cité page 24.)
- [36] E. BOROS et P.L. HAMMER : Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002. (Cité page 87.)
- [37] A. BRANDSTÄDT : $(P_5, \text{diamond})$ -free graphs revisited : structure and linear time optimization. *Discrete applied mathematics*, 138(1-2):13–27, 2004. (Cité pages 81, 83 et 96.)
- [38] A. BRANDSTÄDT et F.F. DRAGAN : On linear and circular structure of $(\text{claw}, \text{net})$ -free graphs. *Discrete applied mathematics*, 129(2-3):285–303, 2003. (Cité page 92.)
- [39] A. BRANDSTÄDT, F.F. DRAGAN, H.O. LE et R. MOSCA : New graph classes of bounded clique-width. *Theory of Computing Systems*, 38(5):623–645, 2005. (Cité page 81.)

- [40] A. BRANDSTÄDT et P.L. HAMMER : On the stability number of claw-free P_5 -free and more general graphs. *Discrete Applied Mathematics*, 95(1-3):163–167, 1999. (Cité page 94.)
- [41] A. BRANDSTÄDT et C.T. HOÀNG : Stability number of bull-and chair-free graphs revisited. *Discrete Applied Mathematics*, 131(1):39–50, 2003. (Cité page 81.)
- [42] A. BRANDSTÄDT et C.T. HOÁNG : On Clique Separators, Nearly Chordal Graphs, and the Maximum Weight Stable Set Problem. In *Integer programming and combinatorial optimization : 11th International IPCO Conference, Berlin, Germany, June 8-10, 2005 : proceedings*, Lecture Notes in Computer Science, volume 3509, pages 265–275. Springer, 2005. (Cité pages 64, 97, 101 et 106.)
- [43] A. BRANDSTÄDT et C.T. HOÁNG : On clique separators, nearly chordal graphs, and the Maximum Weight Stable Set Problem. *Theoretical Computer Science*, 389(1-2):295–306, 2007. (Cité pages 64, 97 et 106.)
- [44] A. BRANDSTÄDT et D. KRATSCH : On the structure of (P_5, gem) -free graphs. *Discrete Applied Mathematics*, 145(2):155–166, 2005. (Cité page 84.)
- [45] A. BRANDSTÄDT, H.O. LE et V.B. LE : On α -redundant vertices in P_5 -free graphs. *Information Processing Letters*, 82(3):119–122, 2002. (Cité pages 69 et 98.)
- [46] A. BRANDSTÄDT, H.O. LE et R. MOSCA : Chordal co-gem-free and (P_5, gem) -free graphs have bounded clique-width. *Discrete Applied Mathematics*, 145(2):232–241, 2005. (Cité page 84.)
- [47] A. BRANDSTÄDT, V.B. LE et H.N. de RIDDER : Efficient robust algorithms for the maximum weight stable set problem in chair-free graph classes. *Information Processing Letters*, 89(4):165–173, 2004. (Cité page 84.)
- [48] A. BRANDSTÄDT, V.B. LE et S. MAHFUD : New applications of clique separator decomposition for the Maximum Weight Stable Set problem. *Theoretical Computer Science*, 370(1-3):229–239, 2007. (Cité pages 101 et 102.)
- [49] A. BRANDSTÄDT et V.V. LOZIN : A note on α -redundant vertices in graphs. *Discrete Applied Mathematics*, 108(3):301–308, 2001. (Cité pages 68, 94 et 97.)
- [50] A. BRANDSTÄDT et R. MOSCA : On the structure and stability number of P_5 -and co-chair-free graphs. *Discrete Applied Mathematics*, 132(1-3):47–65, 2003. (Cité page 84.)
- [51] A. BRANDSTÄDT et R. MOSCA : On variations of P_4 -sparse graphs. *Discrete Applied Mathematics*, 129(2-3):521–532, 2003. (Cité pages 81 et 82.)

- [52] S. BRANDT : Computing the independence number of dense triangle-free graphs. In *Graph-Theoretic Concepts in Computer Science*, pages 100–108. Springer, 1997. (Cité page 44.)
- [53] A. BRETSCHER, D. CORNEIL, M. HABIB et C. PAUL : A simple linear time lexbfs cograph recognition algorithm. *SIAM Journal on Discrete Mathematics*, 22(4):1277–1296, 2008. (Cité page 78.)
- [54] H. BROERSMA, F.V. FOMIN, P.A. GOLOVACH et D. PAULUSMA : Three Complexity Results on Coloring P_k -Free Graphs. In *Combinatorial Algorithms : 20th International Workshop, IWOCA 2009, Hradec Nad Moravici, Czech Republic, June 28–July 2, 2009*, Lecture Notes in Computer Science, volume 5874, pages 95–104. Springer-Verlag, 2009. (Cité page 111.)
- [55] H. BROERSMA, P.A. GOLOVACH, D. PAULUSMA et J. SONG : Narrowing down the gap on the complexity of coloring P_k -free graphs. In *Proceedings of the 36th international conference on Graph-theoretic concepts in computer science, WG 2010, Zarós, Crete, Greece, June 28–30, 2010*, Lecture Notes in Computer Science, volume 6410, pages 63–74. Springer-Verlag, 2010. (Cité page 111.)
- [56] C. BRON et J. KERBOSCH : Finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973. (Cité page 43.)
- [57] A.C. BROWN : On the theory of isomeric compounds. *Trans. Roy. Soc. Edinburgh*, 23:707–720, 1864. (Cité page 171.)
- [58] A.C. BROWN : On the Use of Graphic Representations of Chemical Formulae. *Proceedings of the Royal Society of Edinburgh*, 5(3):429–31, 1865. (Cité page 171.)
- [59] D. BRUCE, C.T. HOÀNG et J. SAWADA : A Certifying Algorithm for 3-Colorability of P_5 -Free Graphs. In *Proceedings of the 20th International Symposium on Algorithms and Computation*, Lecture Notes in Computer Science, volume 5878, pages 594–604. Springer-Verlag, 2009. (Cité pages 116 et 133.)
- [60] A. BUCHHEIM : On Clifford’s Theory of Graphs. *Proceedings of the London Mathematical Society*, 1(1):80, 1885. (Cité page 172.)
- [61] M. BURLET et J.P. UHRY : Parity graphs. In C. BERGE et V. CHVÁTAL, éditeurs : *Topics on perfect graphs*, volume 21 de *Annals of Discrete Mathematics*, pages 253–277. North-Holland, 1984. (Cité page 77.)
- [62] A. CAYLEY : XXVIII. On the theory of the analytical forms called trees. *Philosophical Magazine Series 4*, 13(85):172–176, 1857. URL <http://www.archive.org/stream/collmathpapers03caylrich#page/242/mode/1up>. (Cité page 171.)
- [63] A. CAYLEY : On the analytical form called trees, Part II. *Philos. Mag.(4)*, 18:374–378, 1859. URL <http://www.archive.org/stream/collectedmathemao4cayluoft#page/112/mode/1up>. (Cité page 171.)

- [64] A. CAYLEY : On the mathematical theory of isomers. *Philosophical Magazine*, 67:444, 1874. URL <http://www.archive.org/stream/collectedmathema09cayluoft#page/202/mode/1up>. (Cité page 171.)
- [65] A. CAYLEY : On the analytical forms called Trees, with application to the theory of chemical combinations. *Rep. Brit. Assoc. Advance. Sci*, 45:257–305, 1875. URL <http://www.archive.org/stream/collectedmathema09cayluoft#page/427/mode/1up>. (Cité page 171.)
- [66] A. CAYLEY : On the colouring of maps. In *Proceedings of the Royal Geographical Society and Monthly Record of Geography*, volume 1 (4), pages 259–261. JSTOR, 1879. (Cité page 176.)
- [67] A. CAYLEY : On the analytical forms called trees. *American Journal of Mathematics*, pages 266–268, 1881. URL <http://www.archive.org/stream/collectedmathema11cayluoft#page/n388/mode/1up/>. (Cité page 171.)
- [68] A. CAYLEY : A theorem on trees. *Quart. J. Math*, 23(376-378):69, 1889. URL <http://www.archive.org/stream/collectedmathema13cayluoft#page/26/mode/1up>. (Cité page 171.)
- [69] D. CHANDLER, M.S. CHANG, A. KLOKS, J. LIU et S.L. PENG : On probe permutation graphs. In *Theory and Applications of Models of Computation*, Lecture Notes in Computer Science, volume 3959, pages 494–504. Springer, 2006. (Cité page 155.)
- [70] D. CHANDLER, M.S. CHANG, T. KLOKS, J. LIU et S.L. PENG : Partitioned probe comparability graphs. In *Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, volume 4271, pages 179–190. Springer, 2006. (Cité page 155.)
- [71] D.B. CHANDLER, M.S. CHANG, T. KLOKS, J. LIU et S.L. PENG : Partitioned probe comparability graphs. *Theoretical Computer Science*, 396(1-3):212–222, 2008. (Cité page 155.)
- [72] G.J. CHANG, A.J.J. KLOKS, J. LIU et S.L. PENG : The PIGs Full Monty—A floor show of minimal separators. In *STACS 2005*, Lecture Notes in Computer Science, volume 3404, pages 521–532. Springer, 2005. (Cité page 155.)
- [73] G. CHARTRAND et P. ZHANG : *Chromatic graph theory*. Chapman & Hall, 2008. (Cité page 176.)
- [74] J. CHEN, I.A. KANJ et W. JIA : Vertex Cover : Further Observations and Further Improvements. *Journal of Algorithms*, 41(2):280–301, 2001. (Cité page 92.)
- [75] N. CHIBA et T. NISHIZEKI : Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14(1):210–223, 1985. (Cité page 45.)
- [76] M. CHUDNOVSKY, G. CORNUÉJOLS, X. LIU, P. SEYMOUR et K. VUŠKOVIĆ : Recognizing Berge graphs. *Combinatorica*, 25(2):143–186, 2005. (Cité page 98.)

- [77] M. CHUDNOVSKY, N. ROBERTSON, P. SEYMOUR et R. THOMAS : The strong perfect graph theorem. *Annals of Math.*, 164:51–229, 2006. (Cité pages 22 et 181.)
- [78] M. CHUDNOVSKY et P. SEYMOUR : The structure of claw-free graphs. *Surveys in combinatorics*, 327:153–171, 2005. (Cité page 48.)
- [79] M. CHUDNOVSKY et P. SEYMOUR : Claw-free graphs. I. Orientable prismatic graphs. *Journal of Combinatorial Theory, Series B*, 97(6):867–903, 2007. (Cité page 48.)
- [80] M. CHUDNOVSKY et P. SEYMOUR : Claw-free graphs. II. Non-orientable prismatic graphs. *Journal of Combinatorial Theory, Series B*, 98(2):249–290, 2008. (Cité page 48.)
- [81] M. CHUDNOVSKY et P. SEYMOUR : Claw-free graphs. III. Circular interval graphs. *Journal of Combinatorial Theory, Series B*, 98(4):812–834, 2008. (Cité page 48.)
- [82] M. CHUDNOVSKY et P. SEYMOUR : Claw-free graphs. IV. Decomposition theorem. *Journal of Combinatorial Theory, Series B*, 98(5):839–938, 2008. (Cité page 48.)
- [83] M. CHUDNOVSKY et P. SEYMOUR : Claw-free graphs. V. Global structure. *Journal of Combinatorial Theory, Series B*, 98(6):1373–1410, 2008. (Cité page 48.)
- [84] M. CHUDNOVSKY et P. SEYMOUR : Claw-free graphs. VII. Quasi-line graphs. *Submitted manuscript*, 2009. (Cité page 48.)
- [85] M. CHUDNOVSKY et P. SEYMOUR : Claw-free graphs. VI. Colouring. *Journal of Combinatorial Theory, Series B*, 100(6):560–572, 2010. (Cité page 48.)
- [86] V. CHVÁTAL : On certain polytopes associated with graphs. *Journal of Combinatorial Theory, Series B*, 18(2):138–154, 1975. (Cité page 49.)
- [87] V. CHVÁTAL : Perfectly ordered graphs. In C. BERGE et V. CHVÁTAL, éditeurs : *Topics on perfect graphs*, volume 21 de *Annals of Discrete Mathematics*, pages 63–65. North-Holland, 1984. (Cité page 25.)
- [88] V. CHVÁTAL et P.L. HAMMER : Set-packing and threshold graphs. Rapport technique, Univ. Waterloo Res. Report, CORR, 1973. (Cité pages 48 et 102.)
- [89] V. CHVÁTAL et P.L. HAMMER : Aggregation of Inequalities in Integer Programming. *Annals of Discrete Mathematics*, 1:145–162, 1977. (Cité page 48.)
- [90] V. CHVÁTAL, C.T. HOÀNG, N.V.R. MAHADEV et D. de WERRA : Four classes of perfectly orderable graphs. *Journal of graph theory*, 11(4):481–495, 1987. (Cité page 83.)
- [91] V. CHVÁTAL et I. RUSU : A note on graphs with no long holes. In *Proc. of DIMACS Workshop on Perfect Graphs at Princeton, NJ*, 1993. (Cité page 95.)

- [92] V. CHVÁTAL et N. SBIHI : Recognizing claw-free perfect graphs. *Journal of Combinatorial Theory, Series B*, 44(2):154–176, 1988. (Cité page 22.)
- [93] W.K. CLIFFORD : *Mathematical Fragments, being facsimiles of his unfinished papers relating to the theory of graphs, by the late W. K. Clifford*. Macmillan, 1881. (Cité page 172.)
- [94] M. CONFORTI, G. CORNUÉJOLS et K. VUŠKOVIĆ : Decomposition of odd-hole-free graphs by double star cutsets and 2-joins. *Discrete applied mathematics*, 141(1-3):41–91, 2004. (Cité page 98.)
- [95] S.A. COOK : The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971. (Cité page 182.)
- [96] D. COPPERSMITH et S. WINOGRAD : Matrix multiplication via arithmetic progressions. *Journal of symbolic computation*, 9(3):251–280, 1990. (Cité page 45.)
- [97] D.G. CORNEIL, H. LERCHS et L. STEWART BURLINGHAM : Complement reducible graphs. *Discrete Applied Mathematics*, 3:163–174, 1981. (Cité pages 77, 78 et 80.)
- [98] D.G. CORNEIL, Y. PERL et L.K. STEWART : A linear recognition algorithm for cographs. *SIAM Journal on Computing*, 14:926–934, 1985. (Cité pages 72, 78 et 97.)
- [99] A.S. COUPER : Sur une nouvelle théorie chimique. *Annales de chimie et de physique*, 53:469–489, 1858. URL <http://gallica.bnf.fr/ark:/12148/bpt6k34794n/f468.table>. (Cité page 171.)
- [100] E. COUPY : Solution d’un problème appartenant à la géométrie de situation, par Euler. *Nouv. Ann. Math.*, 10:106–119, 1851. URL http://archive.numdam.org/ARCHIVE/NAM/NAM_1851_1_10_/NAM_1851_1_10__106_1/NAM_1851_1_10__106_1.pdf. (Cité page 169.)
- [101] B. COURCELLE, J. ENGELFRIET et G. ROZENBERG : Handle-rewriting hypergraph grammars. *Journal of computer and system sciences*, 46(2):218–270, 1993. (Cité page 80.)
- [102] B. COURCELLE, J.A. MAKOWSKY et U. ROTICS : Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000. (Cité page 81.)
- [103] B. COURCELLE et S. OLARIU : Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000. (Cité page 81.)
- [104] M. COZZENS et L. KELLEHER : Dominating cliques in graphs. *Discrete Mathematics*, 86(1-3):101–116, 1990. (Cité pages 52 et 56.)
- [105] P. DAMASCHKE : Induced subgraph isomorphism for cographs is \mathcal{NP} -complete. In *Graph-Theoretic Concepts in Computer Science*, pages 72–78. Springer, 1991. (Cité page 80.)

- [106] S. DANTAS, C.M.H. de FIGUEIREDO, M.C. GOLUMBIC, S. KLEIN et F. MAFFRAY : The chain graph sandwich problem. *Annals of Operations Research*, pages 1–7, 2010. URL <http://dx.doi.org/10.1007/s10479-010-0792-0>. (Cité page 154.)
- [107] S. DANTAS, S. KLEIN, C.P. de MELLO et A. MORGANA : The graph sandwich problem for P_4 -sparse graphs. *Discrete Mathematics*, 309(11):3664–3673, 2009. (Cité page 154.)
- [108] G.B. DANTZIG : Application of the simplex method to a transportation problem. In Tj.C. KOOPMANS, éditeur : *Activity analysis of production and allocation*, pages 359–373, 1951. Proceedings Conference on Linear Programming, Chicago, Illinois, 1949. (Cité page 31.)
- [109] H.N. de RIDDER : *On probe classes of graphs*. Thèse de doctorat, Institut für Informatik, Universität Rostock, Deutschland, 2007. (Cité page 155.)
- [110] C. DE SIMONE : On the vertex packing problem. *Graphs and Combinatorics*, 9(1):19–30, 1993. (Cité page 82.)
- [111] C. DE SIMONE et R. MOSCA : Stable set and clique polytopes of (P_5, gem) -free graphs. *Discrete Mathematics*, 307(22):2661–2670, 2007. (Cité page 84.)
- [112] C. DE SIMONE et A. SASSANO : Stability number of bull-and chair-free graphs. *Discrete Applied Mathematics*, 41(2):121–129, 1993. (Cité page 51.)
- [113] D. de WERRA et P.L. HAMMER : Weighted stability number of graphs and weighted satisfiability : The two facets of pseudo-Boolean optimization. *Annals of Operations Research*, 149(1):67–73, 2007. (Cité page 86.)
- [114] D. de WERRA et D. KOBLER : Graph coloring : foundations and applications. *RAIRO Oper. Res.*, 37:29–66, 2003. (Cité page 111.)
- [115] T. DENLEY : The independence number of graphs with large odd girth. *Electron. J. Combin*, 1(9), 1994. (Cité page 60.)
- [116] A.K. DEWDNEY et A. KEEWATIN : Fast Turing Reductions Between Problems in \mathcal{NP} : Chapter 4 : Reductions Between \mathcal{NP} -complete Problems. Rapport technique, Dept. of Computer Science, University of Western Ontario, 1981. (Cité page 44.)
- [117] G.A. DIRAC : On rigid circuit graphs. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 25, pages 71–76. Springer, 1961. (Cité page 23.)
- [118] D. DUFFUS, R.J. GOULD et M.S. JACOBSON : Forbidden subgraphs and the Hamiltonian theme. In *Proceedings of the Fourth International Conference on the Theory and Applications of Graphs, Kalamazoo, Michigan, USA*, pages 297–316, 1980. (Cité page 92.)

- [119] C. EBENEGER, P.L. HAMMER et D. de WERRA : Pseudo-Boolean functions and stability of graphs. In *Annals of Discrete Mathematics*, volume 19, pages 83–97. North-Holland, 1984. (Cité pages 89, 90 et 91.)
- [120] J. EDMONDS : Maximum matching and a polyhedron with 0, 1-vertices. *Journal of Research of the National Bureau of Standards*, 69 (1-2):125–130, 1965. (Cité pages 32 et 47.)
- [121] J. EDMONDS : Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17(3):449–467, 1965. (Cité pages 1, 46 et 182.)
- [122] J. EDMONDS et R.M. KARP : Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19 (2):248–264, 1972. (Cité page 46.)
- [123] J. EGERVÁRY : Matrixok kombinatorius tulajdonságairól [Hungarian, with German summary]. *Matematikai és Fizikai Lapok*, 38:16–28, 1931. (Cité page 41.)
- [124] M. EL-ZAHAR et P. ERDŐS : On the existence of two non-neighboring subgraphs in a graph. *Combinatorica*, 5(4):295–300, 1985. (Cité page 53.)
- [125] P. ERDŐS, A.L. RUBIN et H. TAYLOR : Choosability in graphs. In *Proc. West Coast Conf. on Combinatorics, Graph Theory and Computing, Congressus Numerantium*, volume 26, pages 125–157, 1979. (Cité page 113.)
- [126] A. ERRERA : *Du coloriage des cartes et de quelques questions d'analysis situs*. Thèse de doctorat, Brussels University, 1921. (Cité page 177.)
- [127] L. EULER : Solutio problematis ad geometriam situs pertinentis. *Opera Omnia*, 7:128–140, 1736. URL <http://www.math.dartmouth.edu/~euler/docs/originals/Eo53.pdf>. (Cité page 169.)
- [128] S. EVEN et O. KARIV : An $\mathcal{O}(n^{2.5})$ algorithm for maximum matching in general graphs. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, pages 100–112. IEEE Computer Society, 1975. (Cité page 47.)
- [129] S. EVEN, A. PNUELI et A. LEMPEL : Permutation graphs and transitive graphs. *Journal of the ACM*, 19(3):400–410, 1972. (Cité page 111.)
- [130] M. FARBER : On diameters and radii of bridged graphs. *Discrete Mathematics*, 73(3):249–260, 1989. (Cité pages 45, 84 et 104.)
- [131] M. FARBER, M. HUYTER et Z. TUZA : An upper bound on the number of cliques in a graph. *Networks*, 23(3):207–210, 1993. ISSN 1097-0037. (Cité pages 69 et 104.)
- [132] R. FAUDREE, E. FLANDRIN et Z. RYJÁČEK : Claw-free graphs—A survey. *Discrete Mathematics*, 164(1-3):87–147, 1997. (Cité page 48.)

- [133] S.W. FÖLDES et P.L. HAMMER : Split graphs. In *Proceedings of the Eighth Southeastern Conference on Combinatorics, Graph Theory, and Computing, Louisiana State University, Baton Rouge, February 28-March 3, 1977*, pages 311–315. Utilitas Mathematica, 1977. (Cité page 102.)
- [134] S.W. FÖLDES et P.L. HAMMER : Split graphs having Dilworth number two. *Canad. J. Math*, 29(3):666–672, 1977. (Cité page 102.)
- [135] L. FORTNOW et S. HOMER : A short history of computational complexity. *Bulletin of the EATCS*, 80:95–133, 2003. (Cité page 182.)
- [136] J.L. FOUQUET : A decomposition for a class of (P_5, \overline{P}_5) -free graphs. *Discrete Mathematics*, 121(1-3):75–83, 1993. (Cité pages 81 et 82.)
- [137] J.L. FOUQUET et V. GIAKOUMAKIS : On semi- P_4 -sparse graphs. *Discrete Mathematics*, 165:277–300, 1997. (Cité pages 81 et 83.)
- [138] J.L. FOUQUET, V. GIAKOUMAKIS, F. MAIRE et H. THUILLIER : On graphs without P_5 and \overline{P}_5 . *Discrete Mathematics*, 146(1-3):33–44, 1995. (Cité pages 83, 96 et 102.)
- [139] A. FRANK : Some polynomial algorithms for certain graphs and hypergraphs. In *Proceedings of the Fifth British Combinatorial Conference, University of Aberdeen, Aberdeen, July 14-18, 1975*, pages 211–226. Utilitas Mathematica Pub., 1976. (Cité page 48.)
- [140] R. FRITSCH et G. FRITSCH : *The four color theorem*. Springer, 1998. (Cité page 177.)
- [141] G. FROBENIUS : Über zerlegbare Determinanten, Sitzungsber König. Preuss. Akad. Wiss, 18:274–277, 1917. (Cité page 42.)
- [142] D.R. FULKERSON et O.A. GROSS : Incidence matrices and interval graphs. *Pacific J. Math.*, 15:835–855, 1965. (Cité page 23.)
- [143] H.N. GABOW : An Efficient Implement at ion of Edmonds' Maximum-Matching Algorithm. Rapport technique, Department of Computer Science, Stanford University, Stanford, California, 1972. (Cité page 47.)
- [144] H.N. GABOW : An efficient implementation of Edmonds' algorithm for maximum matching on graphs. *Journal of the ACM (JACM)*, 23(2):221–234, 1976. (Cité page 47.)
- [145] H.N. GABOW et R.E. TARJAN : Faster scaling algorithms for general graph matching problems. *Journal of the ACM (JACM)*, 38(4):815–853, 1991. (Cité page 47.)
- [146] T. GALLAI : Über extreme Punkt-und Kantenmengen. *Ann. Univ. Sci. Budapest, Eotvos Sect. Math*, 2:133–138, 1959. (Cité page 41.)
- [147] T. GALLAI : Graphen mit triangulierbaren ungeraden Vielecken. *Publ. Math. Inst. Hung. Acad. Sci., Ser. A*, 7:3–36, 1962. (Cité page 98.)
- [148] T. GALLAI : Transitiv orientierbare graphen. *Acta Mathematica Hungarica*, 18(1):25–66, 1967. (Cité page 72.)

- [149] M.R. GAREY et D.S. JOHNSON : The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 32(4):826–834, 1977. (Cité page 44.)
- [150] M.R. GAREY et D.S. JOHNSON : *Computers and Intractability : An Introduction to the Theory of NP-completeness*. WH Freeman, San Francisco, 1979. (Cité pages 19, 54, 153 et 183.)
- [151] M.R. GAREY, D.S. JOHNSON et L. STOCKMEYER : Some simplified \mathcal{NP} -complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63. ACM, 1974. (Cité pages 44 et 80.)
- [152] M.R. GAREY, D.S. JOHNSON et L. STOCKMEYER : Some simplified \mathcal{NP} -complete problems. *Theoretical Computer Science*, 1(3):193–267, 1976. (Cité pages 44, 80 et 111.)
- [153] F. GAVRIL : Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. *SIAM Journal on Computing*, 1:180–187, 1972. (Cité pages 48 et 111.)
- [154] F. GAVRIL : Algorithms on clique separable graphs. *Discrete Mathematics*, 19(2):159–165, 1977. (Cité page 98.)
- [155] M.U. GERBER : *Theoretical and Algorithmic Developments for Vertex Partitioning Problems*. Thèse de doctorat, École Polytechnique Fédérale de Lausanne, 2001. (Cité pages 69 et 97.)
- [156] M.U. GERBER, A. HERTZ et V.V. LOZIN : Stable sets in two subclasses of banner-free graphs. *Discrete applied mathematics*, 132(1-3):121–136, 2003. (Cité page 106.)
- [157] M.U. GERBER, A. HERTZ et V.V. LOZIN : Augmenting chains in graphs without a skew star. *Journal of Combinatorial Theory, Series B*, 96(3):352–366, 2006. (Cité pages 65 et 106.)
- [158] M.U. GERBER, A. HERTZ et D. SCHINDL : P_5 -free augmenting graphs and the maximum stable set problem. *Discrete Applied Mathematics*, 132(1-3):109–119, 2003. (Cité pages 69 et 72.)
- [159] M.U. GERBER et V.V. LOZIN : On the stable set problem in special P_5 -free graphs. *Discrete Applied Mathematics*, 125(2-3):215–224, 2003. (Cité pages 69 et 97.)
- [160] V. GIAKOUMAKIS et I. RUSU : Weighted parameters in (P_5, \overline{P}_5) -free graphs. *Discrete Applied Mathematics*, 80(2):255–262, 1998. (Cité pages 82 et 104.)
- [161] E. GIOAN et C. PAUL : Split decomposition and graph-labelled trees : characterizations and fully-dynamic algorithms for totally decomposable graphs. *Arxiv preprint arXiv :0810.1823*, 2008. (Cité page 78.)
- [162] W. GODDARD et M.A. HENNING : Clique/connected/total domination perfect graphs. *Bulletin of the ICA*, 41:20–21, 2004. (Cité page 52.)

- [163] P. GOLOVACH et P. HEGGERNES : Choosability of P_5 -free graphs. In *Mathematical Foundations of Computer Science 2009*, Lecture Notes in Computer Science, volume 5734, pages 382–391. Springer, 2009. (Cité page 113.)
- [164] M. GOLUBIC et M. LIPSHTEYN : Chordal probe graphs. In *Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, volume 2880, pages 249–260. Springer, 2003. (Cité page 155.)
- [165] M.C. GOLUBIC : Trivially perfect graphs. *Discrete Mathematics*, 24(1):105–107, 1978. (Cité page 52.)
- [166] M.C. GOLUBIC : *Algorithmic graph theory and perfect graphs*. North-Holland, 2004. (Cité page 49.)
- [167] M.C. GOLUBIC et P.L. HAMMER : Stability in circular arc graphs. *Journal of Algorithms*, 9(3):314–320, 1988. (Cité page 95.)
- [168] M.C. GOLUBIC, H. KAPLAN et R. SHAMIR : Graph sandwich problems. *Journal of Algorithms*, 19(3):449–473, 1995. (Cité page 153.)
- [169] M.C. GOLUBIC et M. LIPSHTEYN : Chordal probe graphs. *Discrete Applied Mathematics*, 143(1-3):221–237, 2004. (Cité page 155.)
- [170] M.C. GOLUBIC, F. MAFFRAY et G. MOREL : A characterization of chain probe graphs. *Annals of Operations Research*, pages 1–9, 2009. Online first. (Cité pages 151 et 155.)
- [171] M.C. GOLUBIC et R. SHAMIR : Complexity and algorithms for reasoning about time : A graph-theoretic approach. *Journal of the ACM*, 40(5):1108–1133, 1993. (Cité page 153.)
- [172] M.C. GOLUBIC et A.N. TRENK : *Tolerance Graphs*, 2004. (Cité page 154.)
- [173] P. GORDAN : Beweis, daß jede Covariante und Invariante einer binären Form eine ganze Function mit numerischen Coefficienten einer endlichen Anzahl solcher Formen ist. *Journal für die reine und angewandte Mathematik*, 69:323–354, 1868. (Cité page 173.)
- [174] P. GORDAN : *Vorlesungen über Invariantentheorie*. Teubner, Leipzig, 1885. (Cité page 173.)
- [175] M.S. GOULD et J. RONALD : Forbidden subgraphs and hamiltonian properties of graphs. *Discrete Mathematics*, 42(2-3):189–196, 1982. (Cité page 92.)
- [176] S. GRAVIER, C.T. HOÀNG et F. MAFFRAY : Coloring the hypergraph of maximal cliques of a graph with no long path. *Discrete mathematics*, 272(2-3):285–290, 2003. (Cité page 115.)
- [177] J.L. GROSS et J. YELLEN : *Handbook of Graph Theory and Applications*. CRC press, 2004. (Cité page 170.)

- [178] M. GRÖTSCHEL, L. LOVÁSZ et A. SCHRIJVER : The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981. (Cité pages 32, 49, 111 et 181.)
- [179] M. GRÖTSCHEL, L. LOVÁSZ et A. SCHRIJVER : Polynomial algorithms for perfect graphs. *Topics on perfect graphs, Ann. Disc. Math.*, 21:325–356, 1984. (Cité pages 49 et 111.)
- [180] A. GYÁRFÁS : Problems from the world surrounding perfect graphs. *Zastosow. Mat.*, 19(3-4):413–441, 1987. (Cité page 115.)
- [181] M. HABIB, R. MCCONNELL, C. PAUL et L. 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. (Cité page 24.)
- [182] M. HABIB et C. PAUL : A simple linear time algorithm for cograph recognition. *Discrete Applied Mathematics*, 145(2):183–197, 2005. (Cité page 78.)
- [183] P. HALL : On representatives of subsets. *Journal of the London Mathematical Society*, 10:26–30, 1935. (Cité page 42.)
- [184] M.M. HALLDÓRSSON : Approximating discrete collections via local improvements. In *Proceedings of the sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 160–169. Society for Industrial and Applied Mathematics, 1995. (Cité page 60.)
- [185] P.L. HAMMER et A. HERTZ : On a transformation which preserves the stability number. *Cahiers du GERAD*, 1991. (Cité page 95.)
- [186] P.L. HAMMER, N.V.R. MAHADEV et D. de WERRA : Stability in CAN-free graphs. *Journal of Combinatorial Theory, Series B*, 38(1):23–30, 1985. (Cité page 92.)
- [187] P.L. HAMMER, N.V.R. MAHADEV et D. DE WERRA : The struction of a graph : application to CN-free graphs. *Combinatorica*, 5(2):141–147, 1985. (Cité page 92.)
- [188] P.L. HAMMER, U.N. PELED et X. SUN : Difference graphs. *Discrete Appl. Math.*, 28(1):35–44, 1990. (Cité page 57.)
- [189] P.L. HAMMER, I. ROSENBERG et S. RUDEANU : On the determination of the minima of pseudo-boolean functions (en roumain). *Studii si Cercetari matematice*, 14:359–364, 1963. (Cité page 87.)
- [190] P.L. HAMMER et S. RUDEANU : *Boolean methods in operations research and related areas*. Springer-Verlag, 1968. (Cité page 87.)
- [191] A. HAMOR et P. LEORITH : Stories of the one-zero-zero-one nights : Abu Boul in Graphistan. *Regards sur la theorie des graphes (Presses Polytechniques Romandes, Lausanne)*, 1980. (Cité page 88.)
- [192] J. HÅSTAD : Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999. (Cité page 44.)

- [193] T.W. HAYNES, S.T. HEDETNIEMI et P.J. SLATER : *Domination in graphs. Advanced topics*. Marcel Decker Inc., New York, 1998. (Cité page 40.)
- [194] T.W. HAYNES, S.T. HEDETNIEMI et P.J. SLATER : *Fundamentals of domination in graphs*. Marcel Decker Inc., New York, 1998. (Cité page 40.)
- [195] R. HAYWARD, C. HOÀNG et F. MAFFRAY : Optimizing weakly triangulated graphs. *Graphs and Combinatorics*, 5(1):339–349, 1989. (Cité page 111.)
- [196] R.B. HAYWARD : Weakly triangulated graphs. *Journal of Combinatorial Theory, Series B*, 39(3):200–208, 1985. (Cité page 111.)
- [197] P.J. HEAWOOD : Map colour theorem. *Quart. J. Pure Appl. Math.*, 24:332–338, 1890. (Cité page 177.)
- [198] P. HEGGERNES, J.A. TELLE et Y. VILLANGER : Computing Minimal Triangulations in Time $\mathcal{O}(n^\alpha \log n) = o(n^{2.376})$. *SIAM Journal on Discrete Mathematics*, 19(4):900–913, 2005. (Cité page 100.)
- [199] A. HERTZ : Quelques utilisations de la structure. *Discrete Mathematics*, 59(1-2):79–89, 1986. (Cité page 92.)
- [200] A. HERTZ : On the use of Boolean methods for the computation of the stability number. *Discrete Applied Mathematics*, 76(1-3):183–203, 1997. (Cité page 96.)
- [201] A. HERTZ : On a graph transformation that preserves the stability number. *Yugoslav Journal of Operations Research*, 10(1):1–12, 2000. ISSN 0354-0243. (Cité page 96.)
- [202] A. HERTZ et D. de WERRA : A magnetic procedure for the stability number. *Graphs and Combinatorics*, 25(5):707–716, 2009. (Cité page 96.)
- [203] A. HERTZ et V.V. LOZIN : *The maximum independent set problem and augmenting graphs*, chapitre 4, pages 69–99. Springer, 2005. (Cité pages 60 et 65.)
- [204] A. HERTZ, V.V. LOZIN et D. SCHINDL : Finding augmenting chains in extensions of claw-free graphs. *Information Processing Letters*, 86(6):311–316, 2003. (Cité pages 65 et 106.)
- [205] C. HIERHOLZER et C. WIENER : Über die Möglichkeit, einen Linenzug ohne Wiederholung und ohne Unterbrechung zu umfahren. *Mathematische Annalen*, 6(1):30–32, 1873. URL <http://www.digizeitschriften.de/main/dms/img/\?PPN=GDZPPN002242176>. (Cité page 170.)
- [206] D. HILBERT : Über die Endlichkeit des Invariantensystems für binäre Grundformen. *Mathematische Annalen*, 33(2):223–226, 1888. (Cité page 173.)
- [207] D. HILBERT : Über die Theorie der algebraischen Formen. *Mathematische Annalen*, 36(4):473–534, 1890. (Cité page 173.)

- [208] C.T. HOÀNG : *Perfect graphs*. Thèse de doctorat, School of Computer Science, McGill University Montreal, 1985. (Cité page 82.)
- [209] C.T. HOÀNG : Recognition and optimization algorithms for co-triangulated graphs. Rapport technique, Tech. Report 90637-OR, Forschungsinstitut für Diskrete Mathematik, Rheinische Friedrich-Wilhelms-Univ., Bonn, Germany, 1990, 1990. (Cité pages 119 et 120.)
- [210] C.T. HOÀNG : Efficient algorithms for minimum weighted colouring of some classes of perfect graphs. *Discrete Applied Mathematics*, 55(2):133–143, 1994. (Cité page 83.)
- [211] C.T. HOÀNG, M. KAMIŃSKI, V. LOZIN, J. SAWADA et X. SHU : Deciding k -colorability of P_5 -free graphs in polynomial time. *Algorithmica*, 57(1):74–81, 2010. (Cité pages 112 et 114.)
- [212] C.T. HOÀNG, F. MAFFRAY, S. OLARIU et M. PREISSMANN : A charming class of perfectly orderable graphs. *Discrete Mathematics*, 102(1):67–74, 1992. (Cité page 83.)
- [213] C.T. HOÀNG et B.A. REED : Some classes of perfectly orderable graphs. *Journal of Graph Theory*, 13(4):445–463, 1989. (Cité page 86.)
- [214] I. HOLYER : The \mathcal{NP} -Completeness of Edge-Coloring. *SIAM Journal on Computing*, 10:718–720, 1981. (Cité page 111.)
- [215] J.E. HOPCROFT et R.M. KARP : A $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. In *12th Annual Symposium on Switching and Automata Theory*, pages 122–125. IEEE, 1971. (Cité page 46.)
- [216] J.E. HOPCROFT et R.M. KARP : An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM Journal on Computing*, 2:225–231, 1973. (Cité page 46.)
- [217] J.E. HOPCROFT et R.E. TARJAN : Isomorphism of planar graphs. *Complexity of computer computations*, pages 143–150, 1972. (Cité page 80.)
- [218] J.E. HOPCROFT et J.K. WONG : Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 172–184. ACM, 1974. (Cité page 80.)
- [219] A. JAGOTA, G. NARASIMHAN et L. ŠOLTÉS : A generalization of maximal independent sets. *Discrete applied mathematics*, 109(3):223–235, 2001. (Cité page 42.)
- [220] L.O. JAMES, R.G. STANTON et D.D. COWAN : Graph decomposition for undirected graphs. In *Proc. 3rd Southeastern International Conference on Combinatorics, Graph Theory, and Computing*, pages 281–290, 1972. (Cité page 76.)
- [221] B. JAMISON et S. OLARIU : A tree representation for P_4 -sparse graphs. *Discrete Applied Mathematics*, 35(2):115–129, 1992. (Cité page 82.)

- [222] D.S. JOHNSON, M. YANNAKAKIS et C.H. PAPADIMITRIOU : On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988. (Cité page 43.)
- [223] J.L. JOHNSON et J.P. SPINRAD : A polynomial time recognition algorithm for probe interval graphs. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 477–486. Society for Industrial and Applied Mathematics, 2001. (Cité page 154.)
- [224] H.A. JUNG : On a class of posets and the corresponding comparability graphs. *Journal of Combinatorial Theory, Series B*, 24(2):125–133, 1978. (Cité pages 77 et 80.)
- [225] N. KARMAKAR : A new polynomial-time algorithm for linear programming. *Combinatorica*, 4:373–395, 1984. (Cité page 31.)
- [226] R.M. KARP : Reducibility among combinatorial problems. *Complexity of computer computations*, 1972. (Cité pages 40, 43, 111 et 183.)
- [227] A.B. KEMPE : On the application of Clifford’s graphs to ordinary binary quantics. *Proceedings of the London Mathematical Society*, 1(1):107, 1885. (Cité page 172.)
- [228] A.B. KEMPE : A memoir on the theory of mathematical form. *Philosophical transactions of the Royal Society of London*, pages 1–70, 1886. URL <http://www.archive.org/stream/amemoirtheorymaookempgoog#page/n7/mode/1up>. (Cité page 178.)
- [229] L. KHACHIYAN : A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979. (Cité page 31.)
- [230] S. KHANNA, N. LINIAL et S. SAFRA : On the hardness of approximating the chromatic number. *Combinatorica*, 20(3):393–415, 2000. (Cité page 111.)
- [231] I. KITTELL : A group of operations on a partially colored map. *Bull. Amer. Math. Soc*, 41:407–413, 1935. URL <http://www.ams.org/journals/bull/1935-41-06/S0002-9904-1935-06104-X/S0002-9904-1935-06104-X.pdf>. (Cité page 177.)
- [232] V. KLEE et G.J. MINTY : How good is the simplex algorithm? *Inequalities*, 3, 1972. (Cité page 31.)
- [233] D. KÖNIG : Graphok és matrixok (Graphes et matrices). *Matematikai és Fizikai Lapok*, 38:116–119, 1931. (Cité pages 41 et 46.)
- [234] D. KRÁL, J. KRATOCHVÍL, Z. TUZA et G.J. WOEGINGER : Complexity of Coloring Graphs without Forbidden Induced Subgraphs. In *Proceedings of the 27th International Workshop on Graph-Theoretic Concepts in Computer Science*, Lecture Notes in Computer Science, volume 2204, pages 254–262. Springer-Verlag, 2001. (Cité page 112.)

- [235] D. KRATSCH et J. SPINRAD : Between $\mathcal{O}(nm)$ and $\mathcal{O}(n^\alpha)$. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 709–716. Society for Industrial and Applied Mathematics, 2003. (Cité page 100.)
- [236] D. KRATSCH et J. SPINRAD : Between $\mathcal{O}(nm)$ and $\mathcal{O}(n^\alpha)$. *SIAM Journal on Computing*, 36(2):310–325, 2006. (Cité page 100.)
- [237] D. KRATSCH et J. SPINRAD : Minimal fill in $\mathcal{O}(n^{2.69})$ time. *Discrete mathematics*, 306(3):366–371, 2006. (Cité page 100.)
- [238] M. LAS VERGNAS : A note on matchings in graphs. Colloque sur la Théorie des Graphes (Paris, 1974). *Cahiers Centre Études Recherche Opérationnelle*, 17:257–260, 1975. (Cité page 47.)
- [239] E.L. LAWLER : *Combinatorial optimization : networks and matroids*. Holt, Rinehart and Winston, NY, USA, 1976. (Cité page 47.)
- [240] E.L. LAWLER, J.K. LENSTRA et A.H.G. RINNOOY KAN : Generating all maximal independent sets : NP-hardness and polynomial-time algorithms. *SIAM Journal on Computing*, 9:558, 1980. (Cité page 43.)
- [241] V.B. LE et H.N. DE RIDDER : Characterisations and linear-time recognition of probe cographs. In *Proceedings of the 33rd international conference on Graph-theoretic concepts in computer science*, Lecture Notes in Computer Science, volume 4769, pages 226–237. Springer-Verlag, 2007. (Cité page 155.)
- [242] V.B. LE, B. RANDEPATH et I. SCHIERMEYER : On the complexity of 4-coloring graphs without long induced paths. *Theoretical Computer Science*, 389(1-2):330–335, 2007. (Cité pages 111 et 112.)
- [243] P.G.H. LEHOT : An optimal algorithm to detect a line graph and output its root graph. *Journal of the ACM (JACM)*, 21(4):569–575, 1974. (Cité page 47.)
- [244] H. LERCHS : On cliques and kernels. Rapport technique, Department of Computer Science, University of Toronto, 1971. (Cité page 77.)
- [245] H. LERCHS : On the clique-kernel structure of graphs. Rapport technique, Department of Computer Science, University of Toronto, 1972. (Cité page 77.)
- [246] L.A. LEVIN : Universal'nye perebornye zadachi (Universal search problems) (en russe). *Problemy Peredachi Informatsii (Problems of Information Transmission)*, 9(3):265–266, 1973. (Cité page 182.)
- [247] R. LIN, S. OLARIU et G. PRUESSE : An optimal path cover algorithm for cographs. *Computers and Mathematics with Applications*, 30(8):75–83, 1995. (Cité page 80.)
- [248] J. LIU et H. ZHOU : Dominating subgraphs in graphs with some forbidden structures. *Discrete Mathematics*, 135(1-3):163–168, 1994. (Cité page 54.)

- [249] L. LOVÁSZ : A characterization of perfect graphs. *Journal of Combinatorial Theory, Series B*, 13(2):95–98, 1972. (Cité page 181.)
- [250] L. LOVÁSZ : On the Shannon capacity of a graph. *IEEE Transactions on Information theory*, 25(1):1–7, 1979. (Cité page 180.)
- [251] L. LOVÁSZ et M.D. PLUMMER : *Matching Theory*, volume 29. Elsevier Science Ltd, 1986. (Cité pages 47, 48 et 94.)
- [252] V.V. LOZIN : Stability in P_5 -and banner-free graphs. *European Journal of Operational Research*, 125(2):292–297, 2000. (Cité pages 65, 69 et 97.)
- [253] V.V. LOZIN et M. MILANIČ : A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 26–30. ACM, 2006. (Cité page 51.)
- [254] V.V. LOZIN et M. MILANIČ : A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *Journal of Discrete Algorithms*, 6(4):595–604, 2008. (Cité pages 48, 51 et 118.)
- [255] V.V. LOZIN et M. MILANIČ : On finding augmenting graphs. *Discrete Applied Mathematics*, 156(13):2517–2529, 2008. (Cité page 60.)
- [256] V.V. LOZIN et R. MOSCA : Independent sets in extensions of $2K_2$ -free graphs. *Discrete applied mathematics*, 146(1):74–80, 2005. (Cité pages 86, 104 et 105.)
- [257] V.V. LOZIN et R. MOSCA : Maximum independent sets in subclasses of P_5 -free graphs. *Information Processing Letters*, 109(6):319–324, 2009. (Cité pages 70 et 86.)
- [258] J. LÜTZEN, G. SABIDUSSI et B. TOFT : Julius Petersen 1839–1910 a biography. *Discrete Mathematics*, 100(1-3):82, 1992. (Cité page 173.)
- [259] S. MA, W.D. WALLIS et J. WU : Optimization problems on quasi-threshold graphs. *J. Comb. Inf. Syst. Sci.*, 14:105–110, 1989. (Cité page 52.)
- [260] F. MAFFRAY : Stable sets in k -colorable P_5 -free graphs. *Information Processing Letters*, 109(23-24):1235–1237, 2009. (Cité pages 58 et 114.)
- [261] F. MAFFRAY et G. MOREL : Un algorithme polynomial pour calculer un stable de poids maximum dans les graphes (P_5 , co-A, co-Domino)-libres. Manuscrit (présenté à ROADEF 2009), 2009. (Cité page 85.)
- [262] F. MAFFRAY et G. MOREL : A linear (in the edges) time algorithm to compute a dominating clique in (P_5, C_5) graphs. Manuscrit, 2010. (Cité page 54.)
- [263] F. MAFFRAY et G. MOREL : On 3-colorable P_5 -free graphs. In *Les Cahiers Leibniz*, n° 191. Laboratoire G-SCOP, 2011. URL <https://consult-cahiersleibniz.g-scop.grenoble-inp.fr/WebObjects/ProjetCahiersLeibniz.woa/Contents/WebServerResources/CahiersLeibniz/cahiers/Cahier191.pdf>. (Cité page 109.)

- [264] F. MAFFRAY et M. PREISSMANN : Linear recognition of pseudo-split graphs. *Discrete Applied Mathematics*, 52(3):307–312, 1994. (Cité page 102.)
- [265] F. MAFFRAY et M. PREISSMANN : On the \mathcal{NP} -completeness of the k -colorability problem for triangle-free graphs. *Discrete Mathematics*, 162(1-3):313–317, 1996. (Cité page 111.)
- [266] F. MAFFRAY et M. PREISSMANN : A Translation of Gallai’s Paper : ‘Transitiv Orientierbare Graphen’. In J.L. RAMIREZ ALFONSIN ET B.A. REED : *Perfect Graphs*, chapitre 3, pages 25–66. John Wiley & Sons Inc, 2001. (Cité page 72.)
- [267] N.V.R. MAHADEV : Vertex deletion and stability number. Swiss Federal Institute of Technology, Technical Report ORWP 90/2. Rapport technique, Swiss Federal Institute of Technology, 1990. (Cité page 96.)
- [268] N.V.R. MAHADEV et U.N. PELED : *Threshold graphs and related topics*, volume 56 de *Annals of Discrete Mathematics*. Elsevier, 1995. (Cité page 58.)
- [269] D. MAIER et J.A. STORER : A note on the complexity of the superstring problem. In *Proceedings of the 12th Annual Conference on Information Sciences and Systems*, volume 5, 1978. (Cité page 44.)
- [270] K. MAKINO et T. UNO : New algorithms for enumerating all maximal cliques. *Algorithm Theory-SWAT 2004*, pages 260–272, 2004. (Cité page 45.)
- [271] R.M. McCONNELL et F. de MONTGOLFIER : Linear-time modular decomposition of directed graphs. *Discrete Applied Mathematics*, 145(2):198–209, 2005. (Cité page 76.)
- [272] R.M. McCONNELL et J.P. SPINRAD : Linear-time transitive orientation. In *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, pages 19–25. Society for Industrial and Applied Mathematics, 1997. (Cité page 72.)
- [273] R.M. McCONNELL et J.P. SPINRAD : Modular decomposition and transitive orientation. *Discrete Mathematics*, 201(1-3):189–241, 1999. (Cité pages 49 et 72.)
- [274] R.M. McCONNELL et J.P. SPINRAD : Construction of probe interval models. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 866–875. Society for Industrial and Applied Mathematics, 2002. (Cité page 155.)
- [275] F.R. McMORRIS, C. WANG et P. ZHANG : On probe interval graphs. *Discrete Applied Mathematics*, 88(1-3):315–324, 1998. (Cité page 154.)
- [276] S. MELLIN : Polynomielle Färbungsalgorithmen für P_k -freie Graphen. Rapport technique, Diplomarbeit am Institut für Informatik, Universität zu Köln, 2002. (Cité page 112.)

- [277] S. MICALI et V.V. VAZIRANI : An $\mathcal{O}(V^{1/2}E)$ algorithm for finding maximum matching in general graphs. In *Proc. 21st Symp. on Foundations of Computer Science*, pages 17–27, 1980. (Cité page 47.)
- [278] M. MILANIČ : *Algorithmic developments and complexity results for finding maximum and exact independent sets in graphs*. Thèse de doctorat, Rutgers University-Graduate School-New Brunswick, 2007. (Cité page 60.)
- [279] G.J. MINTY : On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B*, 28(3):284–304, 1980. (Cité pages 47 et 59.)
- [280] J.W. MOON et L. MOSER : On cliques in graphs. *Israel journal of Mathematics*, 3(1):23–28, 1965. (Cité page 42.)
- [281] G. MOREL : Stabilité dans les graphes sans P_5 . Mémoire de D.E.A., INP Grenoble, juin 2007. (Cité page 60.)
- [282] R. MOSCA : Polynomial algorithms for the maximum stable set problem on particular classes of P_5 -free graphs. *Information Processing Letters*, 61(3):137–143, 1997. (Cité pages 65 et 68.)
- [283] R. MOSCA : Stable sets in certain P_6 -free graphs. *Discrete Applied Mathematics*, 92:177–191, 1999. (Cité pages 64 et 106.)
- [284] R. MOSCA : Some results on maximum stable sets in certain P_5 -free graphs. *Discrete Applied Mathematics*, 132(1-3):175–183, 2004. (Cité pages 67 et 71.)
- [285] R. MOSCA : Some observations on maximum weight stable sets in certain P_5 -free graphs. *European Journal of Operational Research*, 184(3):849–859, 2008. (Cité page 69.)
- [286] R. MOSCA : Stable sets of maximum weight in (P_7, banner) -free graphs. *Discrete Mathematics*, 308(1):20–33, 2008. (Cité pages 64, 101 et 106.)
- [287] R. MOSCA : Independent sets in $(P_6, \text{diamond})$ -free graphs. *Discrete Mathematics and Theoretical Computer Science*, 11(1):125–140, 2009. (Cité page 106.)
- [288] H.M. MULDER : Julius Petersen’s theory of regular graphs. *Discrete mathematics*, 100(1-3):157–175, 1992. (Cité page 174.)
- [289] D. NAKAMURA et Tamura A. : A revision on Minty’s algorithm for finding a maximum weight stable set of a claw-free graph. *J. Oper. Res. Soc. Japan*, 44(2):194–204, 2001. (Cité page 48.)
- [290] R.Z. NORMAN et M.O. RABIN : An algorithm for a minimum cover of a graph. *Proceedings of the American Mathematical Society*, 10(2):315–319, 1959. ISSN 0002-9939. (Cité page 46.)

- [291] G. ORIOLO, U. PIETROPAOLI et G. STAUFFER : A new algorithm for the maximum weighted stable set problem in claw-free graphs. *Integer Programming and Combinatorial Optimization*, pages 77–96, 2008. (Cité pages 48 et 51.)
- [292] Y. ORLOVICH, A. DOLGUI, G. FINKE, V. GORDON et F. WERNER : On the Complexity and Inapproximability of Dissociation Set Problems in Graphs. *Manuscrit*, 2010. (Cité page 42.)
- [293] C. PAPADOPOULOS et C. VOGLIS : Drawing graphs using modular decomposition. In *Proc. 13th International Symposium on Graph Drawing (GD'05)*, Lecture Notes in Computer Science, volume 3843, pages 343–354. Springer, 2005. (Cité page 72.)
- [294] K.R. PARTHASARATHY et G. RAVINDRA : The strong perfect-graph conjecture is true for $K_{1,3}$ -free graphs. *Journal of Combinatorial Theory, Series B*, 21(3):212–223, 1976. (Cité page 47.)
- [295] M.C. PAULL et S.H. UNGER : Minimizing the number of states in incompletely specified sequential switching functions. *Electronic Computers, IRE Transactions on*, 3:356–367, 1959. (Cité page 45.)
- [296] J. PETERSEN : Die theorie der regulären graphs. *Acta Math*, 15:193–220, 1891. URL <http://www.archive.org/stream/actamathematica23leftgoog#page/n205/mode/1up>. (Cité pages 45 et 174.)
- [297] J. PETERSEN : Sur le théoreme de Tait. *L'intermédiaire des Mathématiciens*, 5:225–227, 1898. URL <http://www.cs.columbia.edu/~sanders/graphtheory/writings/translations/petersen1.ps>. (Cité page 178.)
- [298] S. POLJAK : A note on stable sets and colorings of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 15(2):307–309, 1974. (Cité page 44.)
- [299] H. PRÜFER : Neuer beweis eines satzes über permutationen. *Archiv für Mathematik und Physik*, 27:142–144, 1918. (Cité page 171.)
- [300] J. RAMIREZ-ALFONSIN et B. REED : *Perfect Graphs*. Wiley, 2001. (Cité page 179.)
- [301] B. RANDEPATH et I. SCHIERMEYER : 3-Colorability for P_6 -free graphs. *Discrete Applied Mathematics*, 136(2-3):299–313, 2004. (Cité page 111.)
- [302] B. RANDEPATH et I. SCHIERMEYER : On maximum independent sets in P_5 -free graphs. *Discrete Applied Mathematics*, 158(9):1041–1044, 2010. (Cité page 107.)
- [303] B. RANDEPATH, I. SCHIERMEYER et M. TEWES : Three-colourability and forbidden subgraphs. II : polynomial algorithms. *Discrete mathematics*, 251(1-3):137–153, 2002. (Cité page 112.)
- [304] N. ROBERTSON, D.P. SANDERS, P. SEYMOUR et R. THOMAS : A new proof of the four-colour theorem. *Electron. Res. Announc. Amer. Math. Soc*, 2(1):17–25, 1996. (Cité page 178.)

- [305] J.M. ROBSON : Finding a maximum independent set in time $\mathcal{O}(2^{n/4})$. *LaBRI, Universite BordeauxI*, pages 1251–1301, 2001. URL <http://www.labri.fr/perso/robson/mis/techrep.html>. (Cité page 43.)
- [306] D.J. ROSE, R.E. TARJAN et G.S. LUEKER : Algorithmic aspects of vertex elimination in graphs. *SIAM Journal on Computing*, 5:266–283, 1976. (Cité pages 23 et 119.)
- [307] N.D. ROUSSOPOULOS : A max $\{m, n\}$ algorithm for determining the graph H from its line graph G. *Information Processing Letters*, 2 (4):108–112, 1973. (Cité page 47.)
- [308] G. SABIDUSSI : Correspondence between Sylvester, Petersen, Hilbert and Klein on invariants and the factorisation of graphs 1889-1891. *Discrete Mathematics*, 100(1-3):99–155, 1992. (Cité page 174.)
- [309] N. SBIHI : *Étude des stables dans les graphes sans étoile*. Thèse de doctorat, Université Scientifique et Médicale de Grenoble (Mathématiques Appliquées), Grenoble, 1978. (Cité page 47.)
- [310] N. SBIHI : Algorithme de recherche d'un stable de cardinalité maximum dans un graphe sans étoile. *Discrete Mathematics*, 29(1):53–76, 1980. (Cité pages 47 et 59.)
- [311] A. SCHRIJVER : *Combinatorial optimization : polyhedra and efficiency*. Springer-Verlag, 2003. (Cité pages 46, 47 et 182.)
- [312] D. SEINSCHKE : On a property of the class of n-colorable graphs. *Journal of Combinatorial Theory, Series B*, 16(2):191–193, 1974. (Cité page 77.)
- [313] C. SHANNON : The zero error capacity of a noisy channel. *Information Theory, IRE Transactions on*, 2(3):8–19, 1956. (Cité page 179.)
- [314] L. SHENG : Cycle free probe interval graphs. *Congressus Numerantium*, pages 33–42, 1999. (Cité page 154.)
- [315] N.Z. SHOR : Ispol'zovanie operatsii rastyazheniya prostranstva v zadachakh minimizatsii vypuklykh funktsii (en russe). *Kibernetika [Kiev]*, 1:6–12, 1970. (traduction anglaise : Utilization of the operation of space dilatation in the minimization of convex functions, *Cybernetics and Systems Analysis*, 6 :7–15, 1972). (Cité page 31.)
- [316] N.Z. SHOR : O skorosti skhodimosti metoda obobshchennogo gradientnogo spuska s rastyazheniem prostranstva (en russe). *Kibernetika [Kiev]*, 2:80–85, 1970. (traduction anglaise : Convergence rate of the gradient descent method with dilatation of the space, *Cybernetics and Systems Analysis*, 6 :102–108, 1972). (Cité page 31.)
- [317] A. SOIFER : Map coloring in the victorian age : problems and history. *Mathematics Competitions*, 10:20–31, 1997. (Cité page 177.)
- [318] W. SPOTTISWOODE : On Clifford's Graphs. *Proceedings of the London Mathematical Society*, 1(1):204, 1878. (Cité page 172.)

- [319] L. STEWART : Cographs : a Class of Tree Representable Graphs. Mémoire de D.E.A., Dept. of Computer Science, University of Toronto, 1978. (Cité page 77.)
- [320] D.P. SUMNER : Graphs indecomposable with respect to the X-join. *Discrete Mathematics*, 6(3):281–298, 1973. (Cité pages 72 et 77.)
- [321] D.P. SUMNER : Dacey graphs. *Journal of the Australian Mathematical Society*, 18(04):492–502, 1974. (Cité page 77.)
- [322] D.P. SUMNER : Graphs with 1-factors. *Proceedings of the American Mathematical Society*, 42(1):8–12, 1974. (Cité page 47.)
- [323] D.P. SUMNER : Subtrees of a graph and chromatic number. In G. CHARTRAND, éditeur : *The Theory and Applications of Graphs*. John Wiley, New York, 1980. (Cité page 111.)
- [324] J.J. SYLVESTER : Chemistry and algebra. *Nature*, 17(432):284, 1878. URL <http://www.archive.org/stream/nature15unkngoog#page/n312/mode/1up>. (Cité page 172.)
- [325] J.J. SYLVESTER : On an application of the new atomic theory to the graphical representation of the invariants and covariants of binary quantics, with three appendices. *American Journal of Mathematics*, 1(1):64–104, 1878. URL <http://www.archive.org/stream/americanjournal19orgggoog#page/n81/mode/1up>. (Cité page 172.)
- [326] P.G. TAIT : Remarks on the colouring of maps. In *Proc. Roy. Soc. Edinburgh*, volume 10 (729), pages 501–503, 1880. (Cité page 177.)
- [327] R.E. TARJAN : Decomposition by clique separators. *Discrete mathematics*, 55(2):221–232, 1985. (Cité page 99.)
- [328] R.E. TARJAN et A.E. TROJANOWSKI : Finding a maximum independent set. *SIAM Journal on Computing*, 6(3):537–546, 1977. (Cité page 43.)
- [329] R.E. TARJAN et M. YANNAKAKIS : Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13:566, 1984. (Cité page 119.)
- [330] R.E. TARJAN et M. YANNAKAKIS : Addendum : Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 14:254, 1985. (Cité page 119.)
- [331] M. TEDDER, D. CORNEIL, M. HABIB et C. PAUL : Simpler linear-time modular decomposition via recursive factorizing permutations. In *Automata, Languages and Programming (35th International Colloquium, ICALP)*, volume 5125 de *Lecture Notes in Computer Science*, pages 634–645. Springer, 2008. (Cité page 76.)
- [332] C. THOMASSEN : Every planar graph is 5-choosable. *Journal of Combinatorial Theory, Series B*, 62(1):180–181, 1994. (Cité page 113.)

- [333] S. TSUKUYAMA, M. IDE, H. ARIYOSHI et I. SHIRAKAWA : A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6:505, 1977. (Cité page 45.)
- [334] A.M. TURING : On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(1):230, 1936–37. URL http://www.dna.caltech.edu/courses/cs129/caltech_restricted/Turing_1936_IBID.pdf. (Cité page 182.)
- [335] A.M. TURING : On computable numbers, with an application to the Entscheidungsproblem. A correction. *Proceedings of the London Mathematical Society*, 43(2):544–546, 1937–38. URL http://www.dna.caltech.edu/courses/cs129/caltech_restricted/Turing_1937_correction_IBID.pdf. (Cité page 182.)
- [336] P. van't HOF : *Exploiting structure to cope with NP-hard graph problems : Polynomial and exponential time exact algorithms*. Thèse de doctorat, Durham University, 2010. (Cité page 57.)
- [337] P. van't HOF et D. PAULUSMA : A new characterization of P6-free graphs. *Discrete Applied Mathematics*, 158(7):731–740, 2010. (Cité page 57.)
- [338] V.V. VAZIRANI : A theory of alternating paths and blossoms for proving correctness of the $\mathcal{O}(\sqrt{VE})$ general graph matching algorithm. Rapport technique, Cornell University, 1989. (Cité page 47.)
- [339] V.V. VAZIRANI : A theory of alternating paths and blossoms for proving correctness of the general graph maximum matching algorithm. *Combinatorica*, 14(1):71–109, 1994. (Cité page 47.)
- [340] VG VIZING : Vertex colorings with given colors (en russe). *Metody Diskret. Analiz*, 29:3–10, 1976. (Cité page 113.)
- [341] S.H. WHITESIDES : An algorithm for finding clique cut-sets. *Information Processing Letters*, 12:31–32, 1981. (Cité page 99.)
- [342] S.H. WHITESIDES : A method for solving certain graph recognition and optimization problems, with applications to perfect graphs. In C. BERGE et V. CHVÁTAL, éditeurs : *Topics on perfect graphs*, volume 21 de *Annals of Discrete Mathematics*, pages 281–297. North-Holland, 1984. (Cité page 99.)
- [343] H. WHITNEY : Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, pages 150–168, 1932. (Cité page 12.)
- [344] G.J. WOEGINGER et J. SGALL : The complexity of coloring graphs without long induced paths. *Acta Cybernetica*, 15(1):107–117, 2001. (Cité pages 111 et 112.)
- [345] E.S. WOLK : The comparability graph of a tree. In *Proc. Amer. Math. Soc*, volume 13, pages 789–795, 1962. (Cité pages 52 et 77.)
- [346] E.S. WOLK : A note on the comparability graph of a tree. In *Proc. Am. Math. Soc*, volume 16, pages 17–20, 1965. (Cité pages 52 et 77.)

- [347] J.H. YAN, J.J. CHEN et G.J. CHANG : Quasi-threshold graphs. *Discrete applied mathematics*, 69(3):245–253, 1996. (Cité page 52.)
- [348] M. YANNAKAKIS : Computing the minimum fill-in is \mathcal{NP} -complete. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):77–79, 1981. (Cité page 153.)
- [349] M. YANNAKAKIS : Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10:310–327, 1981. (Cité pages 42 et 57.)
- [350] M.S. YU et C.H. YANG : An $\mathcal{O}(n)$ time algorithm for maximum matching on cographs. *Information processing letters*, 47(2):89–93, 1993. (Cité page 80.)
- [351] P. ZHANG : Probe interval graphs and its applications to physical mapping of DNA. Manuscrit, 1994. (Cité page 154.)
- [352] I.E. ZVEROVICH : Independent domination on $2P_3$ -free perfect graphs. Rapport technique, DIMACS Technical Report 2003-22, 2003. (Cité page 106.)
- [353] I.E. ZVEROVICH et O.I. ZVEROVICH : Stability number in subclasses of P_5 -free graphs. *Applied Mathematics-A Journal of Chinese Universities Ser. B*, 19(2):125–132, 2004. (Cité page 83.)

INDEX

- A (graphe), 85
- aimant, 95
- algorithme
 - de Minty, 47
 - de Sbihi, 47
 - du couplage maximum dans un graphe biparti, 46
 - du simplexe, 31
 - robuste, 54
- $\alpha(G)$, 18
- antenne, 91
- anticomplet
 - ensemble, 15
- antitrou, voir graphe
- arbre, 21
- atome, 99
- augmentation, 60

- cerf-volant*, 82
- chaîne, 15
 - S-augmentante, 59
 - alternée, 45
 - augmentante, 45
- chaise*, 51
- classe
 - fortement héréditaire, 50
 - monogène, 105
- classe héréditaire, 16
- clique, 17
 - d'articulation, 98
- clique width, voir largeur de clique
- coarbre, 78
- cographe, 49
- coloration, 18
 - fractionnaire, 19
 - par listes, 19
 - propre, 19
 - totale, 18
- complémentaire, voir graphe

- complet
 - ensemble, 15
 - graphe, 14
- complexité
 - spatiale, 27
 - temporelle, 27
- contraction, 74
- couplage, 41
 - parfait, 41
- cycle, 15

- décomposition modulaire, 72
- degré, voir graphe
- démagnétisation, 95
- diamant*, 82
- domino*, 85
- drapeau₂, 65

- ellipsoïde, 31

- filet*, 91
- fonction
 - booléenne, 86
 - pseudo-booléenne, 86
- forêt, 21

- $\gamma(G)$, 39
- gemme*, 82
- graphe, 9
 - H-libre, 16
 - P_4 -épars, 82
 - k-colorable, 19
 - à seuil, 48
 - adjoint, 11
 - antitrou, 22
 - arc, 9
 - arête, 9
 - multiple, 9
 - augmentant, 59
 - autocomplémentaire, 20

- biparti, 15
- boucle, 9
- centre, 53
- chain-probe*, 155
- clique, 17
- coloration, 18
- complémentaire, 10
- complet, 14
- composante connexe, 13
- connexe, 12
- cordal, 22
- d'Edmonds, 47
- d'intervalle, 23
- de Berge, 22
- de chaîne, 57
- de co-comparabilité, 49
- de comparabilité, 48
- de différence, 57
- de permutation, 48
- degré, 11
- diminuant, 62
- distance, 53
- faiblement triangulé, 111
- homéomorphe, 50
- isomorphe, 20
- line graph, 11
- multigraphe, 9
- ordre, 9
- orienté, 9
- parfait, 22
 - conjecture faible, 22
 - conjecture forte, 22
- parfaitement ordonnable, 25
- pondéré, 10
- premier, 74
- pseudo-scindé, 102
- pseudographe, 9
- quotient, 75
- sans H , 16
- scindé, 58, 102
- semi- P_4 -épars, 83
- simple, 9
- sommet, 9
 - adjacent, 10
 - d'articulation, 13
 - isolé, 11
- stable, 17
- triangulé, 22
- trou, 22
- voisin, 10
 - voisinage, 10
- griffe, 12
- $i(G)$, 40
- incrément, 60
- isomorphisme de graphes, 20
- jumeaux, 73
- $\chi(G)$, 19
- largeur de clique, 80
- lemme
 - de Berge, 46
 - des poignées de mains, 11
- Lex-BFS, 23
- liste d'adjacence, 14
- maison*, 13
- maison jumelée*, 97
- matrice
 - d'adjacence, 13
 - d'incidence, 13
- méthode de l'ellipsoïde, 31
- module, 72
 - contraction, 74
 - fort, 74
 - substitution, 74
 - trivial, 74
- multigraphe, voir graphe
- \mathcal{NP} , 28
- $v(G)$, 41
- \mathcal{O} , 27
- $\omega(G)$, 18
- ordre d'élimination simplicial, 23
- \mathcal{P} , 28
- parcours en largeur lexicographique, 23
- 2-patte, 82
- polyèdre, 30
- polytope, 30
 - des couplages, 32
- posiforme, 87
- problème, 26
 - \mathcal{NP} -complet, 28
 - probe*, 154
 - d'optimisation, 26
 - d'optimisation combinatoire, 27
 - de décision, 26

- de la clique maximum, 38
- de la complétion, 153
- de séparation, 32
- du couplage maximum, 41
- du dominant minimum, 39
- du dominant stable minimum, 40
- du Sandwich, 153
- du stable de poids maximum, 39
- du stable maximum, 37
- du transversal minimum, 40
- instance, 26
- taille, 26
- programme linéaire, 30
- pseudographe, voir graphe
- réduction de voisinage, 95
- réduction polynomiale, 28
- repliage de sommets, 92
- séparation de sommet, 92
- simplexe, 31
- sommet
 - simplicial, 23
- sous-graphe, 16
 - couvrant, 16
 - induit, 16
- stable, 17
 - maximal, 17
 - maximum, 17
 - minimal, 17
 - minimum, 17
- struction, 86
 - totale, 93
- substitution, 74
- sur-graphe, 16
- taureau*, 51
- théorème
 - d'Alekseev, 49
 - de Hall, 42
 - de König-Egerváry, 41
 - de séparation/optimisation, 33
 - des graphes augmentants, 61
 - des graphes parfaits, 22
- transversal, 40
- trou, voir graphe
- voisin, voir graphe
- voisinage, voir graphe

Titre Stabilité et coloration des graphes sans P_5

Résumé La classe des graphes sans P_5 , c'est-à-dire des graphes ne contenant pas de chaîne induite à cinq sommets, est d'un intérêt particulier en théorie des graphes. Il s'agit en effet de la plus petite classe définie par un seul sous-graphe connexe interdit pour laquelle on ignore encore s'il existe un algorithme polynomial permettant de résoudre le problème du stable maximum. Or ce problème, dont on sait qu'il est difficile en général, est d'une grande importance en pratique (problèmes de planification, d'allocation de registres dans un processeur, biologie moléculaire...). Dans cette thèse, nous commençons par dresser un état de l'art complet des méthodes utilisées pour résoudre le problème dans des sous-classes de graphes sans P_5 , puis nous étudions et résolvons ce problème dans une sous-classe particulière, la classe des graphes sans P_5 3-colorables. Nous apportons également des solutions aux problèmes de la reconnaissance et de la coloration de ces graphes, chaque fois en temps linéaire. Enfin, nous définissons, caractérisons et sommes capable de reconnaître les graphes "chain-probe", qui sont les graphes auxquels il est possible de rajouter des arêtes entre certains sommets de sorte qu'ils soient bipartis et sans P_5 . Les problèmes de ce type proviennent de la génétique et ont également des applications en intelligence artificielle.

Mots-clés Théorie des graphes, Graphes sans P_5 , Stabilité, Coloration, Décomposition modulaire.

Title Independent sets and coloring of P_5 -free graphs

Abstract The class of P_5 -free graphs, namely the graphs without induced chains with five vertices, is of particular interest in graph theory. Indeed, it is the smallest class defined by only one forbidden connected induced subgraph for which the complexity of the Maximum Independent Set problem is unknown. This problem has many applications in planning, CPU register allocation, molecular biology... In this thesis, we first give a complete state of art of the methods used to solve the problem in P_5 -free graphs subclasses; then we study and solve this problem in a particular subclass, the class of 3-colorable P_5 -free graphs. We also bring solutions to recognition and coloring problems of these graphs, each time in linear time. Finally, we define, characterize, and are able to recognize "chain-probe" graphs, namely the graphs for which we can add edges between particular vertices such that the resulting graph is bipartite and P_5 -free. Problems of this type come from genetics and have application in I.A.

Keywords Graph theory, P_5 -free graphs, Independent sets, Graph coloring, Modular Decomposition.