

Ordonnancement chromatique: polyèdres, complexité et classification

Vincent JOST

Mémoire présenté pour l'obtention du grade de docteur de l'Université Joseph FOURIER

sous la direction de András SEBŐ et Nadia BRAUNER

Discipline: Mathématique, Informatique

Spécialité: Recherche Opérationnelle, Combinatoire et Optimisation

Thèse soutenue publiquement le

5 octobre 2006

Jury:

Pierre BÉRARD (président) professeur à l'UJF à Grenoble

Gérard CORNUÉJOLS (rapporteur) professeur à Aix-Marseille II

Dominique DE WERRA (rapporteur) professeur à l'EPFL à Lausanne

Philippe BAPTISTE (examineur) CR CNRS au LIX à Palaiseau

András SEBŐ (directeur) DR CNRS à Leibniz-IMAG à Grenoble

Nadia BRAUNER (co-directrice) MdC-HDR à l'UPMF à Grenoble

Menu de la semaine

0	Dejà quatre ans de mathématiques ?	5
I	Méthodes pour l'optimisation combinatoire	13
1	Tour d'horizon de la boîte à outils	15
2	Caractérisations de structures combinatoires	87
II	Théorèmes d'ordonnement chromatique	125
3	La coloration bornée	127
4	Batches avec économie d'échelle	183
5	Un objectif pour voir les rayons γ	215
	Conclusion	243
A	Classifications, SGBD et posets	247
B	Mots équilibrés et ordonnancement juste-à-temps	263
	Bibliographie	274
	Index	283

Chapitre 0

Dejà quatre ans de mathématiques ?

Dans ce mémoire, nous étudions des problèmes qui font intervenir simultanément des contraintes étudiées en théorie de l'ordonnancement et en coloration de graphes. Le paradigme que nous tentons principalement d'appliquer pour résoudre ces problèmes est celui de la combinatoire polyédrale. Celui-ci permet d'imaginer et de prouver des relations min-max pour des problèmes d'optimisation combinatoire. De manière complémentaire, la théorie de la complexité algorithmique permet de mettre à jour certaines difficultés intrinsèques aux problèmes considérés. Ce chapitre vise à donner un aperçu des questions et des résultats discutés dans ce document. Nous commençons par expliquer comment les questions ont émergé et dans quelle perspectives de recherche elles s'inscrivent. Nous expliquons ensuite comment ce mémoire est structuré et en particulier, les liens logiques entre les différentes parties. Nous décrivons finalement comment les principaux résultats s'articulent avec les outils classiques de la théorie des graphes et de l'ordonnancement.

Mots clés du mémoire: Coloration de graphes, ordonnancement, combinatoire polyédrale, complexité algorithmique, classes de graphes parfaits.

Autres objets utilisés : Matroïdes et fonctions sous-modulaires, théorie des jeux coopératifs, voyageur de commerce, approximation polynomiale.

Menu du jour

Genèse de ce mémoire	6
Structure du document et conseils de lecture	7
Morceaux choisis d'ordonnancement chromatique	8

Genèse de ce mémoire

Lors de la conférence MAPSP'03 à Aussois, j'ai assisté à une présentation de Gerd Finke qui a profondément influencé mon approche de l'optimisation combinatoire. Le problème discuté lors de cette présentation peut être vu aussi bien comme un problème de coloration de graphes que comme un problème d'ordonnancement par batchs. C'est de ce mélange de problématiques que naît l'ordonnancement chromatique¹. Le problème suivant est discuté en détail dans l'Exemple 1.63 page 77.

[Pintro] Problème introductif mais néanmoins central

Données : Un graphe d'intervalle $G = (V, E)$, une capacité $b \in \mathbb{N}$ et des temps d'opération $p : V \rightarrow \mathbb{N}$.

Résultat : Une partition en cliques $\{C_1, C_2, \dots, C_k\}$ des sommets de G telle que chaque C_i contienne au plus b sommets.

Mesure : $\sum_{i=1}^k (\max_{v \in C_i} p(v))$ à minimiser.

En un sens, ce problème est le fil conducteur de ce mémoire, car nous discutons plusieurs cas particuliers et extensions. Malgré cela, sa complexité reste ouverte.

D'un autre côté, le rôle de ce problème dans ce mémoire est surtout d'avoir initié un questionnement de combinatoire polyédrale. En effet, ce problème combine des problématiques pour lesquelles on sait donner des relations min-max. Par exemple, si l'on enlève les contraintes de capacité et qu'on se restreint à des temps d'opération unitaires, le problème se ramène à trouver une partition en un nombre minimum de cliques. Or les graphes d'intervalles sont des graphes parfaits. En conséquence, le nombre minimum de cliques qu'il faut pour partitionner un graphe d'intervalles G est égal à la taille d'un plus grand stable de G .

Le travail présenté ici est sous-tendu par deux idées directrices. Premièrement, bien que les théories classiques de la combinatoire polyédrale (graphes parfaits, matroïdes et fonctions sous-modulaires [129]) offrent des outils pertinents pour de nombreuses applications, d'autres problématiques nécessitent des ponts permettant de combiner les outils de base de ces théories. Deuxièmement, il apparaît que les formules min-max associées à ces théories classiques admettent des généralisations communes, permettant la construction de certains de ces ponts. Les résultats discutés dans ce mémoire consistent en grande partie à prouver la pertinence ainsi qu'à définir la limite de validité de ces idées directrices.

¹Le terme *Chromatic Scheduling* est promu par Dominique de Werra depuis plus de 30 ans [54, 56, 57, 58, 59, 61, 62, 63, 108].

Du point de vue de la combinatoire polyédrale, l'apport principal de ce mémoire consiste dans la construction de quelques-uns de ces ponts. De manière complémentaire, la théorie de la complexité indique qu'il n'existe probablement pas de généralisation commune aux développements les plus pointus de ces théories. Cependant, plusieurs de ces généralisations NP-complètes admettent des algorithmes polynomiaux donnant de bonnes approximations.

Nous invitons le lecteur à consulter la Figure 5.9 page 245 pour avoir une idée schématique du type de ponts que nous avons construit et de la complémentarité des différents paradigmes utilisés dans ce mémoire.

Structure du document et conseils de lecture

Ce document est découpé en trois parties. La première partie (Chapitres 1 et 2) traite de concepts, de méthodes et de résultats classiques de l'optimisation combinatoire. La deuxième partie (Chapitres 3, 4 et 5) est consacrée à la mise en œuvre et à l'hybridation des outils présentés dans la première partie, dans le but de résoudre des problèmes d'ordonnancement chromatique. La troisième partie (Annexes A et B) concerne des résultats ne portant pas directement sur l'ordonnancement chromatique.

La plupart des résultats originaux de ce mémoire se trouvent dans la deuxième partie. De plus, les trois chapitres de la deuxième partie sont indépendants et peuvent donc être lus dans un ordre arbitraire.

Chacun des 5 chapitres, commence par un résumé des résultats qui y sont traités, une liste de mots-clés et un menu détaillé. Combiné avec l'index, cela devrait permettre d'accéder rapidement aux prérequis contenus dans la première partie. Techniquement parlant, les résultats de la deuxième partie s'appuient principalement sur les Sections 1.2, 1.3 et 2.2 (même si ces sections contiennent plus de résultats que des prérequis pour la deuxième partie).

A l'intérieur de chaque chapitre, tous les énoncés (théorèmes, propositions, conjectures, exemples...) suivent une même numérotation (selon l'ordre des pages). J'ai fait ce choix de numérotation pour faciliter la recherche dichotomique d'un énoncé lorsque l'on en connaît le numéro mais pas la page. Les formules, les tables et les figures suivent cependant une numérotation indépendante.

Dans tous les chapitres, j'ai disséminé des sections nommées "Petite philosophie à l'usage des (non)-spécialistes". J'y présente mes conclusions sur ce que m'a enseigné l'étude des problèmes techniques discuté dans les sections suivantes. Un des objectifs est de faire sentir que certaines conclusions mathématiques qui sont tirées dans un cadre précis semblent pouvoir se généraliser facilement à des cadres bien plus vastes, pourvu que l'on accepte de se placer à un niveau de réflexion plus informel.

Dans l'Annexe A, je discute quelques principes informatiques qui permettent de structurer les connaissances et l'accès à celles-ci. Je souhaite appliquer ces principes aux théorèmes d'ordonnement chromatique dans les années à venir. Cependant, les deuxièmes et troisièmes thèses proposées dans L'Annexe A ont aussi une vocation transdisciplinaire. En effet, je suis persuadé que la structuration des connaissances ainsi que la facilitation de leur accès sont, actuellement, les deux enjeux essentiels du développement de notre compréhension des réalités scientifiques. Cela me semble valable autant pour la connaissance spécialisée que pour la culture générale.

Dans l'Annexe B je présente une partie de mes travaux qui portent sur l'ordonnement juste-à-temps. Plutôt que de décrire un corpus de résultats techniques, j'ai essayé de mettre en avant les questions et conclusions d'ordre général vers lesquelles j'ai convergé lors de l'étude de ce problème. Ces conclusions résultent principalement de la confrontation de points de vues issus de l'optimisation, de l'économie et de l'éthique. Elles mènent à des interprétations mathématiques élémentaires mais surprenantes.

Morceaux choisis d'ordonnement chromatique

Nous résumons ici comment les principaux résultats de ce mémoire s'articulent avec les théories des graphes, de l'ordonnement et de la combinatoire polyédrale. Pour cela, nous survolons les 5 chapitres de ce mémoire en mentionnant les résultats qui apparaissent aussi dans les trois papiers suivants.

- [76] Batch processing with interval graph compatibilities between tasks –2004– (accepté à *Discrete Applied Mathematics* avec G. Finke, M. Queyranne et A. Sebő).
- [106] Precoloring co-Meyniel graphs –2005– (soumis à *Graphs and combinatorics* avec B. Lévêque et F. Maffray).
- [84] Clique partitioning of interval graphs with submodular costs on the cliques –2006– (accepté à *RAIRO-OR* avec D. Gijswijt et M. Queyranne).
- ... Dans cette section, les résultats qui n'ont pas (encore) été soumis sont marqués du symbole \lrcorner en fin de phrase.

Chapitre 1 : Tour d'horizon de la boîte à outils

Décider si un système linéaire $\{Ax \preceq b\}$ définit un polyèdre entier est un problème co-NP-complet [126]. Nous prouvons que ça le reste même si A et b sont à coefficients $\{0, 1\}$ \lrcorner .

Nous traduisons les questions liées au cœur d'un jeu coopératif dans le langage des

fonctions d'ensemble [1]. Le cœur du jeu associé à la coloration minimum donne une caractérisation des graphes parfaits [67, 66]. Nous prouvons que le cœur du jeu associé au TSP est intimement lié aux graphes TSP-parfaits [1].

Nous présentons les problématiques de ce mémoire dans le langage de l'ordonnement. Deux aspects se distinguent par rapport aux problématiques de l'ordonnement classique. Premièrement, nous étudions des contraintes d'incompatibilité entre les tâches. Cela est équivalent aux problématiques de la coloration. Deuxièmement, les temps d'opération sont définis par des fonctions d'ensemble et permettent d'exprimer de nombreuses problématiques de recherche opérationnelle en termes d'ordonnement par batchs [84].

Chapitre 2 : Caractérisations des classes de graphes

Nous étudions diverses caractérisations (classiques pour la plupart) de certaines classes de graphes. Quelques "types de caractérisations" (sous-graphes exclus, décomposition, validité de formules min-max ou d'algorithmes...) sont présentées, puis spécialisées à quelques classes de graphes [1]. Les ordres (et graphes) d'intervalles sont caractérisés par une forme d'équivalence des problématiques de l'ordonnement chromatique et d'ordonnement avec contraintes de précédence [1]. Les line-graphes de multi-graphes bipartis sont caractérisés comme les graphes dont l'ensemble des stables peut être décrit par les indépendants communs de deux matroïdes [1].

Certains points communs entre les hypergraphes définis par les stables d'un graphe et les indépendants d'un matroïde sont décrits pour les hypergraphes héréditaires en général [1].

Finalement, nous présentons quelques classes de fonctions d'ensembles après avoir noté que certaines classes de fonctions sous-modulaires satisfont une propriété d'échange intéressante relativement au problème de partition de coût minimum du Chapitre 4 [84].

Chapitre 3 : La coloration bornée

Le problème de la coloration b -bornée consiste à trouver le nombre minimum $\chi_b(G)$ de couleurs qu'il faut pour colorier le graphe G en imposant que chaque couleur soit utilisée pour au plus b sommets. Nous étudions la borne inférieure (1) pour ce problème qui s'exprime plus aisément sur le graphe complémentaire [76]. Pour $U \subseteq V$, notons $C_1(U), C_2(U) \dots C_t(U)$ les composantes connexes du graphe $G[U]$ induit par U et $\bar{\chi}_b(G) := \chi_b(\bar{G})$.

$$(1) \quad \bar{\chi}_b(G) \geq \max_{U \subseteq V} \left(\sum_{i=1}^t \left\lceil \frac{|C_i(U)|}{b} \right\rceil \right)$$

La borne (1) généralise l'inégalité de clique pour la coloration minimum et la formule de Berge-Tutte-Gallai pour le nombre minimum d'arêtes qu'il faut pour couvrir les sommets d'un graphe [76]. Le résultat principal du Chapitre 3 est qu'un graphe G est triangulé si et seulement si cette borne inférieure donne une formule min-max pour tout b et tout graphe H provenant de la réplication de G [1]. Nous prouvons aussi la polynomialité du problème de la coloration bornée sur les compléments des graphes triangulés [1].

La coloration b -bornée peut être reformulée comme un problème d'ordonnancement avec temps d'opération unitaires, contraintes d'incompatibilité entre les tâches et minimisation du makespan. Cela s'applique soit au cas de b machines parallèles, soit au cas d'une machine par batch avec batchs de capacité au plus b .

Chapitre 4 : Batchs avec économie d'échelle

Soit $f : \mathcal{P}(V) \rightarrow \mathbb{R}$ une fonction de coût. Le problème de la partition en cliques de coût minimum consiste à trouver une partition en cliques² $\{C_1, C_2 \dots C_k\}$ des sommets de G qui minimise $\sum_{i=1}^k f(C_i)$ (le problème de la partition en un nombre minimum de cliques correspond à $f \equiv \mathbf{1}$) [84]. Notons $\bar{\chi}(G, f)$ ce coût minimum. Nous considérons la borne inférieure suivante sur $\bar{\chi}(G, f)$ donnée par la maximisation de $1^T x$ sous les contraintes (2), où $x \in \mathbb{R}^V$ [1].

$$(2) \quad x(K) \leq f(K) \text{ pour toute clique } K \text{ de } G$$

Le système (2) est TDI³ pour certaines paires de classes de graphes et de classes de fonctions d'ensemble. Cela donne un cadre théorique commun aux formules min-max associées aux graphes parfaits, à l'intersection des polymatroïdes, aux c -couplages dans les graphes bipartis, à la formule de Greene-Kleitman pour la q -coloration partielle [1]. L'intérêt principal de cette borne est toutefois qu'elle donne lieu à de nouvelles formules min-max, particulièrement pour le problème "max-coloring"⁴. De plus, cette borne peut être formulée dans le cadre de la théorie coopérative des jeux avec coopération restreinte aux cliques d'un graphe [1].

Indépendamment de cette approche polyédrale, nous proposons un algorithme de programmation dynamique pour ce problème de partition de coût minimum restreint aux graphes d'intervalles. Nous prouvons l'optimalité de cet algorithme pour plusieurs classes de fonctions de coût [76, 84]⁵.

La partition en cliques de coût minimum peut être reformulée comme un problème d'ordonnancement par batch avec temps d'opérations généralisés, contraintes de

²Une clique est un sous-graphe complet, pas nécessairement maximal par inclusion.

³*totally dual integral*.

⁴Evoqué dans [84]. Les graphes pour lesquels la formule min-max s'applique pour le problème "max-coloring" ne sont pas encore bien compris.

⁵Un premier algorithme a été proposé dans [76] pour la "max-coloration". Une autre version de cet algorithme, adaptée à une classe plus large de fonctions de coût a été proposée dans [84].

compatibilité données par un graphe et minimisation du makespan [76, 84]. Les temps d'opération généraux élargissent considérablement le champ d'application des modèles d'ordonnancement par batchs, y compris à certains problèmes de tournées de véhicules [84].

Chapitre 5 : Un objectif pour voir les rayons γ

Nous présentons des liens entre les problèmes de la coloration minimum, du stable maximum, du sous-graphe k -chromatique maximum, de la coloration par liste et de l'extension précolorée. Nous décrivons des techniques polyédrales classiques et nouvelles pour ces problèmes.

L'extension précolorée, [PrExt] consiste à trouver une coloration minimum quand certains sommets sont déjà coloriés a priori. Nous réduisons [PrExt] à la coloration classique dans un graphe auxiliaire, dans lequel chaque classe de précouleur a été contractée en un sommet [106]. Le résultat principal du Chapitre 5 est que le graphe auxiliaire est parfait si et seulement si le graphe de départ est le complément d'un graphe de Meyniel [106]. Cela implique la polynomialité du problème de l'extension précolorée sur les compléments des graphes de Meyniel [106].

Les problèmes de ce chapitre peuvent être reformulés comme des problèmes d'ordonnancement avec temps d'opération unitaires et contraintes d'incompatibilité [\\]. Les différents problèmes peuvent être reformulés en jouant sur la fonction objectif, les dates de disponibilité et d'échéance [\\]. Cela s'applique à l'ordonnancement par batch ainsi qu'à l'ordonnancement sur machines parallèles [\\].

Première partie

Méthodes pour l'optimisation
combinatoire

Chapitre 1

Tour d'horizon de la boîte à outils

Les notions classiques nécessaires dans la suite de ce mémoire sont résumées dans les mots clés ci-dessous puis présentées dans ce chapitre. Certains résultats nouveaux ou peu connus sont aussi discutés plus en détail :

- Une preuve de la co-NP-complétude de la reconnaissance des polyèdres entiers donnés par des systèmes à coefficients $\{0, 1\}$ (Section 1.3.2.3).
- Une réduction de la coloration minimum des sommets d'un graphe $G = (V, E)$ au stable maximum dans un graphe auxiliaire qui a E comme ensemble de sommets (forêts clique-connectantes, Section 1.3.3.4).
- Une caractérisation des graphes parfaits du point de vue du jeu coopératif associé au problème de la coloration minimum (Section 1.4.2.3). Un lien fort entre les graphes TSP-parfaits et le jeu coopératif associé au problème du voyageur de commerce (Section 1.4.2.4).

Mots clés :

Section 1.1 : relation binaire, ordre, diagramme de Hasse, graphe, stable $\alpha(G)$, clique $\omega(G)$, coloration $\chi(G)$, couplage $\nu(G)$, couverture par des arêtes $\rho(G)$, digraphe, hypergraphe, matroïde.

Section 1.2 : Complexité des algorithmes et des problèmes, P, NP, FP, FNP, co-NP, approximation, forte et pseudo-polynomialité, oracles, promesses et robustesse.

Section 1.3 : Système et programme linéaire, dualité, PLNE, polyèdre entier, système TDI, box-TDI et TDAU, matrice totalement unimodulaire, plans coupants, combinatoire polyédrale, formule de Berge-Tutte, formulations polyédrales associées aux graphes parfaits, forêts clique-connectantes, polymatroïdes.

Section 1.4 : Jeu coopératif, cœur, fonction (totalement) équilibrée. Jeu de la coloration minimum. Jeu du TSP graphique. Point entier dans le cœur d'une fonction totalement équilibrée.

Section 1.5 : Ordonnement (chromatique), classification $\alpha | \beta | \gamma$.

Menu du jour

1.1 Objets combinatoires	16
1.1.1 Métriques, ordres et fonctions d'ensemble	16
1.1.2 Graphes, digraphes et hypergraphes	19
1.2 Complexité, approximation, oracles et promesses	24
1.2.1 Problèmes, langages et algorithmes	24
1.2.2 Complexité algorithmique : FP, FNP, P, NP et co-NP	25
1.2.3 Oracles de résultats, réductions et complétude	27
1.2.4 Oracles de données	29
1.2.5 Problèmes numériques	30
1.2.6 Optimisation et approximation	31
1.2.7 Promesses et robustesse	32
1.3 Programmes linéaires et relations min-max	34
1.3.1 Polyèdres, programmation et dualité linéaire	34
1.3.2 Polyèdres entiers, programmation et dualité entière	37
1.3.3 Quelques théorèmes de combinatoire polyédrale	45
1.4 Coeur d'un jeu coopératif combinatoire	53
1.4.1 Jeux coopératifs : une affaire de cœur	54
1.4.2 Jeux coopératifs combinatoires	62
1.4.3 Imputations entières dans le cœur	69
1.5 Une classification pour l'ordonnancement chromatique	74
1.5.1 Introduction à l'ordonnancement chromatique par batch	75
1.5.2 Classification $\alpha \beta \gamma$ pour l'ordonnancement chromatique	80

1.1 Objets combinatoires

Cette section rappelle les définitions et notations concernant les objets combinatoires utilisés dans la suite de ce mémoire. Nous suivons Schrijver [129] lorsque le choix d'une notation s'impose.

1.1.1 Métriques, ordres et fonctions d'ensemble

1.1.1.1 Ensembles

On définit souvent des ensembles avec la notation :

$$(1.1) \ S := \{x \in X \mid x \text{ vérifie la propriété P}\}$$

$S:=T$ aussi noté $S\stackrel{\text{def}}{=}T$ signifie que l'on définit S comme identique à T . Quant à $\{x \in X \mid x \text{ vérifie } P\}$, cela se lit : "l'ensemble des éléments x de X tels que x vérifie la propriété P ". La **cardinalité** d'un ensemble X , notée $|X|$, est le nombre d'éléments de X . Elle peut être infinie, mais si nous ne le précisons pas, elle sera supposée finie. Le **produit cartésien** de $X \times Y$ est l'ensemble des paires (x, y) avec $x \in X$ et $y \in Y$. X^n est le produit cartésien de n copies de X . On note $X \subseteq Y$ l'inclusion de l'ensemble X dans l'ensemble Y . Si $X \subseteq Y$ mais $X \neq Y$, X est un **sous-ensemble propre** de Y , ce que l'on note $X \subset Y$. L'union de X et Y est notée $X \cup Y$, l'intersection est notée $X \cap Y$. La différence de X et Y , notée $X \setminus Y$ est l'ensemble des éléments de X qui ne sont pas dans Y . La différence symétrique est notée $X \Delta Y$:

$$X \Delta Y = (X \setminus Y) \cup (Y \setminus X) = (X \cup Y) \setminus (X \cap Y)$$

L'ensemble des sous-ensembles de X est noté $\mathcal{P}(X)$. On a $|\mathcal{P}(X)| = 2^{|X|}$. Une **collection** est synonyme d'un ensemble mais est utilisé pour un ensemble dont les éléments sont eux aussi des ensembles. Si X est un ensemble et v un singleton,

$$X + v := X \cup \{v\} \quad \text{et} \quad X - v := X \setminus \{v\}$$

Le **vecteur caractéristique** d'un ensemble U inclus dans un ensemble V (où V est implicite) est le vecteur noté $\mathbf{1}_U \in \mathbb{R}^V$ dont la coordonnée $v \in V$ vaut 1 si $v \in U$ et 0 sinon. Le vecteur caractéristique de V est noté simplement $\mathbf{1}$. Les vecteurs sont supposés écrits en colonne. Le produit scalaire de x et y est donc noté $x^T y$. Pour $x \in \mathbb{R}^V$, on note x_v la v -ième coordonnée de x . On pose aussi

$$(1.2) \quad x(U) := x^T \mathbf{1}_U = \sum_{v \in U} x_v$$

1.1.1.2 Nombres

L'ensemble des entiers naturels, des entiers relatifs, des nombres rationnels et des réels est noté $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ et \mathbb{R} respectivement. On note aussi $\mathbb{R}_+ := \{x \mid x \in \mathbb{R}, x \geq 0\}$.

La **partie entière inférieure** $\lfloor x \rfloor$, d'un nombre réel x est le plus grand entier inférieur ou égal à x . La **partie entière supérieure** $\lceil x \rceil$, d'un nombre réel x est le plus petit entier supérieur ou égal à x .

1.1.1.3 Relations

Une **relation** \mathcal{R} sur un ensemble X est une application $\mathcal{R} : X \times X \rightarrow \{\text{vrai}, \text{faux}\}$. De manière équivalente, une relation est un sous-ensemble de $X \times X$ et les deux notations suivantes sont interchangeables :

$$(1.3) \quad x \mathcal{R} y \iff \mathcal{R}(x, y)$$

Les propriétés suivantes vont nous permettre de définir de nombreuses structures combinatoires (comme les (di)-graphes) ou les ordres à partir des relations. Une relation \mathcal{R} sur X est dite

- *réflexive* si $x\mathcal{R}x$ pour tout $x \in X$.
- *antiréflexive* si $x\not\mathcal{R}x$ pour aucun $x \in X$.
- *transitive* si $(x\mathcal{R}y \text{ et } y\mathcal{R}z) \implies x\mathcal{R}z$ pour tout $x, y, z \in X$
- *antitransitive* si $(x\mathcal{R}y \text{ et } y\mathcal{R}z) \implies \neg(x\mathcal{R}z)$ pour tout $x, y, z \in X$
- *symétrique* si $x\mathcal{R}y \implies y\mathcal{R}x$ pour tout $x, y \in X$
- *antisymétrique* si $(x\mathcal{R}y \text{ et } y\mathcal{R}x) \implies x = y$ pour tout $x, y \in X$
- *totale* si $x\mathcal{R}y$ ou $y\mathcal{R}x$ pour tout $x, y \in X$

Soient $X \subseteq Y$, \mathcal{R}_1 une relation sur X et \mathcal{R}_2 une relation sur Y . On dit que \mathcal{R}_2 *étend* \mathcal{R}_1 , ou que \mathcal{R}_1 est une *relation partielle* de \mathcal{R}_2 si pour tout $v, w \in X$, $v\mathcal{R}_1w \implies v\mathcal{R}_2w$.

1.1.1.4 Métriques

Une *métrique* sur un ensemble V est une fonction $d : V \times V \rightarrow \mathbb{R}$ qui satisfait les conditions suivantes pour tout $x, y, z \in V$.

- (1.4) (i) $d(x, y) \geq 0$
 (ii) $d(x, y) = 0 \iff x = y$
 (iii) $d(x, y) + d(y, z) \geq d(x, z)$ *inégalité triangulaire*

1.1.1.5 Ordres

Une relation \leq sur un ensemble X est appelée *pré-ordre* si elle est réflexive et transitive. Si de plus \leq est antisymétrique, on l'appelle *ordre partiel*, ou simplement *ordre*. Si \leq est un ordre partiel sur X , on appelle *ensemble partiellement ordonné*, ou encore *poset*, la paire (X, \leq) . Finalement, un ordre est *total*, ou *linéaire*, si c'est une relation totale. Un ordre total \preceq sur X est une *extension linéaire* d'un ordre partiel \leq si la relation \preceq étend la relation \leq . Si nous référons à un poset (X, \leq) par $\{x_1, \dots, x_n\}$, cela signifie que $x_i \leq x_j$ pour tout $i \leq j$ (*i.e.* l'ordre $\{1, 2, \dots, n\}$ étend l'ordre \leq). Dans un ensemble partiellement ordonné (X, \leq) , on appelle *idéal inférieur* un sous-ensemble Y de X tel que si $y \in Y$ et $z \leq y$, alors $z \in Y$. De même, Y est un *idéal supérieur* si $y \in Y$ et $y \leq z$ impliquent $z \in Y$.

Soit (X, \geq) un poset et $x, y \in X$. On dit que x *couvre* y , et on note $x \twoheadrightarrow y$ quand $x \geq y$ et qu'il n'existe pas $z \in X$ tel que $x \geq z \geq y$. Un *diagramme de Hasse* de (X, \geq) est une représentation de (X, \geq) qui place chaque point de X dans le plan

respectant les deux règles suivantes :

- si $x \geq y$ alors x est placé “plus haut” que y ,
- il y a une flèche (ou un trait) entre x et y si et seulement si x couvre y .

Bien que plusieurs diagrammes de Hasse soient possibles pour un même poset (X, \leq) , la relation \rightarrow est unique. Par abus de langage, on dit donc que (X, \rightarrow) est le **diagramme de Hasse** de (X, \geq) .

1.1.1.6 Fonctions d'ensemble et sous-modularité

Une **fonction d'ensemble** sur V est une fonction $f : \mathcal{P}(V) \rightarrow \mathbb{R}$. La fonction f est

- **entière** si $f(U) \in \mathbb{N}$ pour tout $U \subseteq V$,
- **sous-cardinale** si $f(U) \leq |U|$ pour tout $U \subseteq V$,
- **non-décroissante** (ou **monotone**) si $f(T) \leq f(U)$ pour tout $T \subseteq U \subseteq V$,
- **sous-additive** si $f(T \cup U) \leq f(T) + f(U) \quad \forall T, U \subseteq V$ tels que $T \cap U = \emptyset$,
- **super-additive** si $f(T \cup U) \geq f(T) + f(U) \quad \forall T, U \subseteq V$ tels que $T \cap U = \emptyset$.

Théorème 1.1 ([129] page 767) Soit $f : V \rightarrow \mathbb{R}$. Les conditions suivantes sont équivalentes :

$$(1.5) \quad f(S \cup T) + f(S \cap T) \leq f(S) + f(T) \quad \text{pour tous } S, T \subseteq V$$

$$(1.6) \quad f(S + u) + f(T) \leq f(S) + f(T + u) \quad \text{pour tous } T \subseteq S \subseteq V \text{ et } u \in V \setminus S$$

$$(1.7) \quad f(S + u + v) + f(S) \leq f(S + u) + f(S + v) \quad \forall S \subseteq V \text{ et } u, v \in V \setminus S$$

Une fonction est **sous-modulaire** si elle satisfait les conditions du Théorème 1.1.

1.1.2 Graphes, digraphes et hypergraphes

1.1.2.1 Graphes

Un **graphe simple** (souvent appelé **graphe**) est une paire $G = (V, E)$ où V est un ensemble fini et E est une relation symétrique et antiréflexive sur V . Cette définition n'est pas standard, car E est d'habitude décrit comme un ensemble de paires non-ordonnées d'éléments de V . Les éléments de V sont appelés **sommets** et les paires d'éléments en relation par E sont appelés **arêtes**. On réfère à une arête soit par un nom, $e \in E$, soit par une paire de sommets, $uv \in E$. Deux sommets liés par une arête sont **adjacents**, ou encore, **voisins**. Si $e = uv$ est une arête, u et v sont les **extrémités** de e . Pour simplifier on dit que u **voit** e et que u voit v . Si deux arêtes ont une extrémité en commun, elles sont **incidentes**, sinon elles sont **disjointes**. L'ensemble des arêtes incidentes à un sommet v est noté $\delta(v)$ et $|\delta(v)|$ est le **degré** de v . Plus généralement, pour $S \subseteq V$, $\delta(S)$ est l'ensemble des

arêtes incidentes à exactement un sommet de S . Le degré maximum d'un sommet est noté $\Delta(G)$:

$$(1.8) \quad \Delta(G) := \max_{v \in V} |\delta(v)|$$

L'ensemble des voisins de v est le **voisinage** $N(v)$ de v . Le **voisinage étendu** de v est $N^+(v) = N(v) + v$. Un **sommet** est **isolé** s'il n'a pas de voisin. Le **graphe complémentaire** \overline{G} du graphe $G = (V, E)$ a le même ensemble V de sommets et est défini par

$$uv \in E(\overline{G}) \iff uv \notin E(G)$$

Le **line-graphe** de $G = (V, E)$ est le graphe $L(G) = (E, E')$ qui a E comme ensemble de sommets et tel que $ef \in E'$ si e et f sont deux arêtes incidentes dans G .

Un **multi-graphe** est un graphe dans lequel on autorise les **arêtes multiples**, c'est-à-dire incidentes à une même paire de sommets, ainsi que les **boucles**, c'est-à-dire incidentes à un seul sommet.

1.1.2.2 Quantités caractéristiques d'un graphe

Par chacune des notions présentées nous souhaitons souvent trouver des objets de cardinalité optimum, ce qui incite à définir certaines quantités caractéristiques associées à un graphe $G = (V, E)$. Un **stable**¹ est un sous-ensemble de sommets deux-à-deux non-adjacents.

$$\alpha(G) := \text{cardinalité maximum d'un stable de } G$$

Une **clique** est un sous-ensemble de sommets deux-à-deux adjacents. Autrement dit, $X \subseteq V$ est une clique de $G = (V, E)$ si et seulement si X est un stable de \overline{G} .

$$\omega(G) := \text{cardinalité maximum d'une clique de } G = \alpha(\overline{G})$$

Un ensemble de sommets X est une **couverture par des sommets**² de G si toute arête est incidente à au moins un sommet de X . Autrement dit, X est une couverture par des sommets si et seulement si $V \setminus X$ est un stable de G .

$$\tau(G) := \text{nombre minimum de sommets dans une couverture de } G = |V| - \alpha(G)$$

Une **k -coloration** est une fonction $\varphi : V \rightarrow [1..k]$ telle que $\varphi(u) \neq \varphi(v)$ pour toute arête uv . Autrement dit, une k -coloration est une partition ordonnée $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ de V en k stables.

$$\chi(G) := \text{nombre minimum de stables nécessaires pour partitionner } V$$

¹aussi appelé **indépendant**.

²*vertex cover*.

Une **partition en cliques** de G est une coloration de \overline{G} .

$$\overline{\chi}(G) := \text{nombre minimum de cliques nécessaires pour partitionner } V = \chi(\overline{G})$$

Un couplage est un sous-ensemble \mathcal{M} d'arêtes disjointes de G .

$$\nu(G) := \text{cardinalité maximum d'un couplage de } G$$

Une **couverture par des arêtes** est un ensemble d'arêtes \mathcal{F} tel que tout sommet est incident à au moins une arête de \mathcal{F} . Une couverture par des arêtes existe si et seulement si G n'a pas de sommet isolé et on a alors ([129] page 316) :

$$\rho(G) := \text{nombre minimum d'arêtes dans une couverture de } G = |V| - \nu(G)$$

Etant donné une capacité $c : V \rightarrow \mathbb{N}$, un **c -couplage simple** est un sous-ensemble \mathcal{M} d'arêtes de G tel que $\mathcal{M} \cap \delta(v) \leq c(v)$ pour tout sommet v (\mathcal{M} est donc un 1-couplage simple si et seulement si c'est un couplage).

1.1.2.3 digraphes

Un **digraphe**³ (simple) est une paire $D = (V, A)$ où V est un ensemble fini et A est une collection de paires ordonnées et distinctes d'éléments de V . Les éléments de V et A sont les sommets et les **arcs** respectivement. Alternativement, un digraphe simple est un ensemble muni d'une relation binaire antiréflexive et antisymétrique⁴. Si on "efface l'orientation" d'un digraphe $D = (V, A)$, on obtient un graphe $G = (V, E)$ tel que E est en bijection avec A , c'est le **graphe sous-jacent** à D . Pour $S \subseteq V$, $\delta^{out}(S)$ est l'ensemble des arcs uv tels que $u \in S$ et $v \notin S$. De même, $\delta^{in}(S)$ est l'ensemble des arcs uv tels que $u \notin S$ et $v \in S$.

1.1.2.4 hypergraphes

Un **hypergraphe** \mathcal{H} est une paire $\mathcal{H} = (V, \mathcal{E})$ où V est un ensemble fini et \mathcal{E} est une famille de sous-ensembles de V . Dans un hypergraphe, les éléments de V sont les **sommets** et ceux de \mathcal{E} sont les **arêtes**. Pour $v, w \in V$ et $e, f \in \mathcal{E}$ on dit que

- v et w sont **adjacents** s'il existe $g \in \mathcal{E}$ avec $v \in g$ et $w \in g$
- v et e sont **incidents** si $v \in e$.
- e et f sont **incidentes** s'il existe $x \in V$ avec $x \in e$ et $x \in f$
- e et f sont **disjointes** si elles ne sont pas incidentes.

La **taille** ou **cardinalité** d'une arête $e \in \mathcal{E}$ est le nombre $|e|$ de sommets qui lui sont incidents. Un hypergraphe peut être décrit par sa matrice d'incidence : pour un hypergraphe \mathcal{H} , sa matrice d'incidence $M(\mathcal{H})$ est la matrice à coefficients dans

³aussi appelé graphe orienté ou *directed graph*.

⁴les digraphes simples ne sont pas toujours supposés antisymétriques dans la littérature.

$\{0, 1\}$ dont les lignes sont indicées par les éléments de V et les colonnes par ceux de \mathcal{E} .

$$(1.9) \quad M_{(v,e)}(\mathcal{H}) = \begin{cases} 1 & \text{si } v \text{ et } e \text{ sont incidents} \\ 0 & \text{sinon} \end{cases}$$

Un hypergraphe est **simple** si sa matrice d'incidence ne comporte pas deux colonnes (arêtes) identiques. Un hypergraphe simple ayant V pour ensemble de sommets peut aussi être vu comme une partie de $\mathcal{P}(V)$. Les hypergraphes que nous considérons dans cette thèse sont implicitement simples. Le **transposé**⁵ d'un **hypergraphe** est l'hypergraphe $\mathcal{H}^T = (\mathcal{E}, V)$ ayant $M(\mathcal{H})^T$ pour matrice d'incidence. Notons qu'un hypergraphe peut être simple sans que son transposé le soit. La fonction de **rang** d'un hypergraphe $\mathcal{H} = (V, \mathcal{E})$ est la fonction d'ensemble $r : \mathcal{P}(V) \rightarrow \mathbb{N}$ définie par $r(U) := \max\{|E| \mid E \in \mathcal{E}, E \subseteq U\}$.

Proposition 1.2 *La fonction $r : \mathcal{P}(V) \rightarrow \mathbb{N}$ est le rang d'un hypergraphe si et seulement si r vérifie les propriétés suivantes pour tous $T, U \subseteq V$:*

$$\begin{aligned} (1.10)(i) \quad r(T) &\leq r(U) \text{ quand } T \subseteq U && \text{monotone} \\ (ii) \quad r(U) &\leq |U| && \text{sous-cardinale} \\ (iii) \quad &\text{si } r(U) > r(U - v) \text{ pour tout } v \in U \text{ alors } r(U) = |U| \end{aligned}$$

Preuve Le rang d'un hypergraphe est sous-cardinal par définition. Il est monotone car si $T \subseteq U \subseteq V$, pour $E \subseteq T$ tel que $r(T) = |E|$, on a aussi $E \subseteq U$. Si $r(U) > r(U - v)$ pour tout $v \in U$ alors il n'existe pas d'arête maximum de taille $r(U)$ incluse dans un sous-ensemble propre de U . Donc U forme une arête et $r(U) = |U|$. Pour une fonction entière qui satisfait (1.10), il existe un unique hypergraphe qui a r comme rang : ses arêtes sont données par $\mathcal{E} := \{E \subseteq V \mid r(E) = |E|\}$. \square

Des variations autour de la condition (1.10) permettent de caractériser certaines classes d'hypergraphes comme les hypergraphes héréditaires (Proposition 1.43), l'ensemble des indépendants d'un matroïde (Théorème 1.3), ou encore l'ensemble des indépendants d'un graphe.

1.1.2.5 matroïdes

Nous référons à [122] pour un traitement plus poussé des matroïdes ainsi que de nombreux exemples et à [129] pour la combinatoire polyédrale des matroïdes et des fonctions sous-modulaires.

⁵Le transposé est habituellement appelé *dual*. Nous évitons le terme *dual* car il nous paraît ambigu (voir Section 2.3.3).

Un **matroïde** est une paire $\mathcal{M} = (V, \mathcal{I})$ où V est un ensemble fini et $\mathcal{I} \subseteq \mathcal{P}(V)$ vérifie les axiomes :

- (1.11)(i) $\emptyset \in \mathcal{I}$
(ii) si $I \subseteq J$ et $J \in \mathcal{I}$ alors $I \in \mathcal{I}$
(iii) si $I, J \in \mathcal{I}$ et $|I| < |J|$ alors il existe $z \in J \setminus I$ tel que $I + z \in \mathcal{I}$

V est l'ensemble des **éléments** et \mathcal{I} l'ensemble des **indépendants**. $X \subseteq V$ est **dépendant** si $X \notin \mathcal{I}$. Un **circuit** est un ensemble dépendant minimal par inclusion. La fonction de **rang** d'un matroïde est la fonction d'ensemble $r : \mathcal{P}(U) \rightarrow \mathbb{N}$ définie par $r(U) := \max\{|I| \mid I \in \mathcal{I}, I \subseteq U\}$. On peut caractériser les matroïdes de la manière suivante :

Théorème 1.3 [129] $r : \mathcal{P}(V) \rightarrow \mathbb{N}$ est le rang d'un matroïde si et seulement si r vérifie les propriétés suivantes pour tous $T, U \subseteq V$:

- (1.12)(i) $r(T) \leq r(U)$ quand $T \subseteq U$ **monotone**
(ii) $r(U) \leq |U|$ **sous-cardinale**
(iii) $r(T \cup U) + r(T \cap U) \leq r(T) + r(U)$ **sous-modulaire**

Un **hyperplan** est un ensemble $H \subseteq V$ maximal par inclusion tel que $r(H) = r(V) - 1$. Il existe de nombreuses structures combinatoires définissant un matroïde (*i.e.* de nombreux hypergraphes dont les arêtes forment les bases d'un matroïde). Ceux que nous utilisons dans ce mémoire sont les suivants.

Matroïdes graphiques Soit $G = (V, E)$ un (multi-)graphe. Soit $\mathcal{F} \subseteq \mathcal{P}(E)$ la collection des forêts de G . Alors $\mathcal{M}(G) = (E, \mathcal{F})$ est un matroïde. Un tel matroïde est aussi appelé **matroïde des cycles** de G , car les circuits de $\mathcal{M}(G)$ sont exactement les circuits de G .

Matroïdes uniformes Soit V un ensemble fini, $n = |V|$ et $b \in \mathbb{N}$. Soit $\mathcal{F} := \{F \subseteq V \mid |F| \leq b\}$, alors $\mathcal{U}_n^b := (V, \mathcal{F})$ est un matroïde. Les circuits de \mathcal{U}_n^b sont tous les ensembles de cardinalité $b + 1$.

Matroïdes de partition Soit V un ensemble fini et $\mathcal{P} = \{P_1, P_2, \dots, P_k\}$ une partition de V . Soit $\mathcal{F} := \{F \subseteq V \mid |F \cap P_i| \leq 1\}$, alors $\mathcal{M} = (V, \mathcal{F})$ est un matroïde. Les circuits de \mathcal{M} sont les paires d'éléments appartenant à une même classe de \mathcal{P} .

1.2 Complexité, approximation, oracles et promesses

Les bases de la théorie de la complexité algorithmique sont discutées. Puis nous introduisons des raffinements utiles pour traiter de problématiques spécifiques.

1.2.1 Problèmes, langages et algorithmes

L'acronyme *Informatique* signifie : “traitement automatique de l'information”. De nombreuses (sinon la plupart) des problématiques de l'informatique théorique et des mathématiques relèvent du traitement de l'information, dans le sens suivant. On dispose de données suffisantes à la résolution d'un problème, mais ces données doivent être traitées en vue de résoudre le problème et cela s'avère d'une difficulté parfois inextricable. Le but de la théorie de la complexité algorithmique est de rendre compte de la difficulté de ces traitements à deux niveaux. Premièrement, on souhaite mesurer la rapidité d'un algorithme donné. Deuxièmement, on souhaite rendre compte de la difficulté intrinsèque à la résolution d'un problème, c'est-à-dire donner l'algorithme le plus rapide pour résoudre un problème donné. Puisque la théorie de la complexité algorithmique a pour vocation d'analyser le traitement automatique de l'information, il est nécessaire de décrire les problèmes que l'on veut considérer en précisant comment l'information est disponible et ce que l'on veut en faire. Le formalisme des problèmes spécifie une tâche à accomplir et les données accessibles pour accomplir cette tâche. Par exemple, “trouver une coloration minimum dans un graphe” peut être décrit par la spécification suivante :

[Color] Coloration minimum (version recherche)

Données : Un graphe $G = (V, E)$, un entier k .

Résultat : Une k -coloration de G (s'il en existe une, sinon renvoyer “non”).

En pratique (c'est-à-dire si on utilise un ordinateur), il est nécessaire d'être extrêmement rigoureux lorsque l'on spécifie le format exact dans lequel le graphe doit être donné et le format dans lequel la coloration doit être rendue. En théorie (c'est-à-dire du point de vue des mathématiques discrètes et de la théorie de la complexité), ces précisions n'ont souvent que peu d'intérêt. En pratique comme en théorie, un *problème* est l'équivalent d'une spécification : c'est une sorte de cahier des charges qui demande de réaliser un travail générique (dans [Color], c'est “trouver une bonne coloration des graphes”). Le problème se spécialise en une infinité de cas particuliers, précisés par un ensemble de paramètres, les *données*. Chaque jeu de valeurs possibles pour les données est appelé *instance* du problème (dans [Color], une instance est “un graphe G et un entier k ”). Pour chaque instance, on doit réaliser le travail générique, dans un format précisé dans le *résultat*. En général, il peut exister plusieurs résultats valides pour une instance donnée, et il faut en choisir

un quelconque. En revanche, s'il n'existe aucun résultat valide, on doit renvoyer "non". La formalisation classique des problèmes fait appel à la théorie des langages. La raison est la suivante. Si le paragraphe ci-dessus permet d'initier le néophyte, il sera critiqué par l'œil expert qui voit se profiler des ambiguïtés qui portent le germe de contradictions. Nous décrivons donc les bases nécessaires de théorie des langages.

Un **alphabet** est un ensemble fini. Pour un alphabet Σ (on peut considérer que $\Sigma = \{0, 1\}$), un **mot** de **longueur** n est un élément de Σ^n . On note

$$\Sigma^* := \bigcup_{n \in \mathbb{N}} \Sigma^n$$

l'ensemble des mots sur Σ . Un **langage** est une partie de Σ^* . Un **problème de décision** est un problème du type

[Decide \mathcal{L}] Reconnaissance du langage $\mathcal{L} \subseteq \Sigma^*$

Données : Un mot x sur Σ

Résultat : "oui" si $x \in \mathcal{L}$ et "non" si $x \notin \mathcal{L}$.

Mais comment peut-on savoir si $x \in \mathcal{L}$ si on ne connaît pas \mathcal{L} ? En général, on ne peut pas⁶. Plus précisément, il existe des langages qu'on ne peut pas reconnaître avec un **algorithme**, c'est-à-dire, un procédé automatique qui fonctionne en temps fini pour toute instance. Ici, "automatique" signifie qu'un algorithme s'écrit comme une suite finies d'opérations considérées élémentaires⁷.

En général, le résultat d'un problème n'est pas une réponse "oui/non". Une manière de s'affranchir de cette particularité est de considérer un problème Π comme une partie de $\Sigma_1^* \times \Sigma_2^*$. L'ensemble des instances de Π est représenté par Σ_1^* tandis que Σ_2^* contient les résultats possibles de toutes les instances. $(I, R) \in \Sigma_1^* \times \Sigma_2^*$ si et seulement si la réponse R est un résultat valide pour l'instance I ⁸. Il est alors pratique de considérer le résultat "non" comme un résultat particulier, qui n'est pas un élément de Σ_2^* , mais qui correspond au fait que pour l'instance I , il n'existe pas de $R \in \Sigma_2^*$ tel que $(I, R) \in \Pi$. Le **problème de décision associé** à un problème $\Pi \subseteq \Sigma_1^* \times \Sigma_2^*$ est le langage $\Pi^{dec} := \{I \in \Sigma_1^* \mid \exists R \in \Sigma_2^*, (I, R) \in \Pi\}$

1.2.2 Complexité algorithmique : FP, FNP, P, NP et co-NP

Comme il existe en général plusieurs algorithmes résolvant un problème donné, on souhaite souvent en trouver un qui soit aussi rapide que possible. On formalise

⁶C'est le Théorème d'indécidabilité de Turing.

⁷Voir [123] pour une formalisation des algorithmes, par exemple avec les machines de Turing. Cette formalisation ne semble toutefois pas nécessaire pour les besoins de ce mémoire.

⁸Les problèmes dont la réponse est plus compliquée que "oui/non" sont habituellement appelés *Functional problems* [123].

habituellement la rapidité d'un algorithme comme suit. On mesure la taille $|I|$ de chaque instance I . Dans le cadre des langages, si $I \in \Sigma^n$, on pose $|I| = n$. Par exemple, pour un graphe $G = (V, E)$, on définit habituellement $|G|$ en fonction de $|E|$ et $|V|$ (par défaut, c'est $|V| + |E|$). Autrement dit la taille d'un graphe c'est son nombre de sommets plus son nombre d'arêtes. On évalue ensuite, le temps d'exécution de l'algorithme sur l'instance I , ce que l'on représente la plupart du temps par le nombre d'opérations élémentaires effectuées. Pour une fonction $f : \mathbb{N} \rightarrow \mathbb{N}$, on dit que l'algorithme A a une complexité $\mathcal{O}(f(n))$ s'il existe une constante c telle que pour toute instance I le temps d'exécution de A sur I est inférieur à $c.f(|I|)$. Un algorithme A est polynomial s'il existe un polynôme p tel que A est de complexité $p(n)$. De manière équivalente, un algorithme est polynomial s'il existe un entier k tel que A est de complexité $\mathcal{O}(n^k)$. Un problème Π est de complexité $\mathcal{O}(f(n))$ s'il existe un algorithme de complexité $\mathcal{O}(f(n))$ le résolvant. Un **problème** est donc **polynomial**, s'il peut être résolu par un algorithme polynomial. **FP**⁹ est la classe des problèmes polynomiaux. **P** est la classe des problèmes de décision polynomiaux.

FNP est la classe des problèmes $\Pi \subseteq \Sigma_1^* \times \Sigma_2^*$ tels qu'il existe un polynôme p tel que $|R| \leq p(|I|)$ pour tout $(I, R) \in \Pi$ et tels que le langage $\{(x, y) \in \Pi\}$ est polynomial. **NP**¹⁰ est la classe des problèmes de décision associé à un problème de FNP. Informellement, un **certificat NP** d'un problème de décision Π est une information supplémentaire que l'on peut associer à chaque instance "oui" I de Π , qui permet de vérifier facilement (i.e. par un algorithme polynomial) que I est bien une instance "oui" de Π . Formellement, un certificat NP est obtenu en considérant un résultat valide pour chaque instance "oui" d'un problème $\Pi' \in \text{FNP}$ dont Π est le problème de décision associé. Par exemple le problème suivant est dans NP :

[Color(dec)] Coloration minimum (version décision)

Données : Un graphe $G = (V, E)$, un entier k .

Résultat : Existe-t-il une k -coloration de G (oui/non).

Commentaires : NP car une k -coloration est un certificat NP.

En effet, on vérifie facilement qu'une fonction $\varphi : V \rightarrow \{1, 2, \dots, k\}$ est une coloration de G : il suffit de vérifier que $\varphi(v) \neq \varphi(w)$ pour tout $(v, w) \in E$.

De manière similaire, un certificat co-NP est un certificat qui permet de vérifier qu'on a bien affaire à une instance non, et **co-NP** est la classe des problèmes qui ont un tel certificat.

S'il est en général aisé de calculer la complexité d'un algorithme, il est souvent

⁹Functional Polynomial.

¹⁰Une définition équivalente de NP, basée sur les machines de Turing, est à l'origine du nom NP : *non-deterministic polynomial*.

très dur d'estimer celle d'un problème. En particulier, les questions suivantes sont actuellement parmi les plus importantes à la fois en mathématiques et en informatique théorique.

Question 1.4 $P \stackrel{?}{=} NP$

Question 1.5 $NP \stackrel{?}{=} co-NP$

Question 1.6 $P \stackrel{?}{=} NP \cap co-NP$

La première de ces trois questions est la plus célèbre, mais les deux autres ont de bonnes chances d'être plus intéressantes encore. Nous éclairons l'enjeu de ces questions dans la section suivante à l'aide du concept de réduction. Par ailleurs, s'il est bien connu que $P = NP$ si et seulement si $FP = FNP$ [123], l'équivalence entre problèmes de recherches et problèmes de décisions nécessiterait une longue discussion [15, 118, 124] car elle soulève des problématiques spécifiques non-résolues aujourd'hui.

1.2.3 Oracles de résultats, réductions et complétude

L'intérêt que suscite la théorie de la complexité algorithmique tient probablement beaucoup dans le fait qu'elle permet de comparer la difficulté de problèmes issus de domaines a priori très éloignés. Grâce aux oracles et aux réductions, ces comparaisons sont souvent possibles même si l'on ne connaît pas la complexité des problèmes en question.

Un *oracle* est une opération élémentaire (virtuelle) que l'on ajoute aux opérations que peut réellement effectuer la machine. Par exemple, un oracle pour le problème [Color] est une opération virtuelle qui lit une instance (G, k) de [Color] et renvoie (comme par magie) une k -coloration de G .

Un problème Π_1 se *réduit au sens de Turing* à un problème Π_2 s'il existe un algorithme A résolvant Π_1 en ayant un oracle pour résoudre Π_2 à disposition. C'est une *réduction* de **Cook** si l'algorithme A est polynomial. Considérons par exemple le problème de factorisation des entiers.

[Factor] Factorisation des nombres entiers

Données : Un nombre entier n

Résultat : Une liste p_1, p_2, \dots, p_k de nombres premiers tels que $n = \prod_{i=1}^k p_i$

Notons que les entrées numériques sont codées en écriture binaire. La taille de n est donc $|n| = \log(n)$. Un algorithme qui essaye tous les nombres de 1 à n (ou même \sqrt{n}) comme diviseurs de n est donc de complexité exponentielle en $|n|$. Bien

que [Factor] ne soit pas un problème de décision, le problème suivant permet de comparer [Factor] avec les problèmes de décision.

[LFact] Langage de décision équivalent à [Factor]

Données : Deux nombres entiers n et a

Résultat : Existe-t-il un entier $d \leq a$ tel que d divise n ?

Commentaires : $\text{NP} \cap \text{co-NP}$. S'il existe, un diviseur $d \leq a$ est un certificat NP. Sinon, la liste des facteurs premiers de n est un certificat co-NP.

La liste des facteurs de n est bien un certificat co-NP de [LFact] car on peut certifier polynomialement qu'un entier est premier (en fait, on peut même décider polynomialement si un entier est premier [2]).

Pour trouver un facteur de n polynomialement avec un oracle pour [LFact], on procède par dichotomie sur l'intervalle $[1..n]$. Cela prend un temps $\mathcal{O}(\log(n))$. Comme n a au plus $\log_2(n)$ facteurs, on peut factoriser complètement n avec un algorithme de complexité $\mathcal{O}((\log(n))^2) = \mathcal{O}(|n|^2)$.

Si $P = \text{NP} \cap \text{co-NP}$, alors il existe un algorithme polynomial pour factoriser les nombres entiers. Le protocole de cryptographie RSA actuellement répandu pour sécuriser les transactions financières se base sur la difficulté de la factorisation des entiers. Aussi, sous l'éventualité que $P = \text{NP} \cap \text{co-NP}$, c'est peut-être bien plus que le million de dollars qu'offre la fondation Clay pour la réponse à $P = \text{NP}$ qui est en jeu.

Une autre réduction, proposée par Karp, est habituellement utilisée pour comparer les problèmes de décision entre eux. Un problème de décision Π_1 se **réduit** polynomialement (au sens de **Karp**) à un problème de décision Π_2 s'il existe une fonction φ des instances de Π_1 dans les instances de Π_2 telle que,

- i) $\varphi(I) \in \Pi_2$ si et seulement si $I \in \Pi_1$,
- ii) φ est calculable en temps polynomial.

Un problème Π_1 est **NP-dur** si tout problème $\Pi \in \text{NP}$ se réduit polynomialement au sens de Turing à Π_1 . Un problème de décision Π_1 est **NP-complet** si $\Pi_1 \in \text{NP}$ et si tout problème $\Pi \in \text{NP}$ se réduit polynomialement au sens de Karp à Π_1 ¹¹. L'existence des problèmes NP-complets est en soi un fait surprenant : pour beaucoup de classes de complexité et notamment $\text{NP} \cap \text{co-NP}$, on ne connaît pas de problèmes complets. Grâce au travail de Cook, on sait que [SAT] est NP-complet. En pratique, pour prouver qu'un problème Π est NP-dur (ou NP-complet), on réduit un problème déjà connu pour être NP-complet (par exemple [SAT]) à Π .

Si $P \neq \text{NP}$, aucun problème NP-complet ne peut être résolu par un algorithme po-

¹¹Les liens entre les réductions de Cook et de Karp (et d'autres) soulèvent des problématiques spécifiques et sont discutées dans [123].

lynomial. Mais sous cette éventualité, il reste envisageable que $\text{NP}=\text{co-NP}$, ce qui pourrait être une avancée sans précédent pour les mathématiques discrètes, car tous les problèmes NP-complets bénéficieraient d'une justification courte que les instances "non", sont bien des instances "non".

1.2.4 Oracles de données : “Comment poser la question quand la taille des données est trop grosse”

Le titre de cette section peut paraître absurde : si l'on veut résoudre un problème, il faut bien commencer par prendre connaissance des données... Cette intuition se révèle fautive quand certaines informations disponibles sont redondantes voire superflues pour résoudre le problème qui nous intéresse. Parfois même, un accès très sobre à ces informations est suffisant. Lorsque nous cherchons un mot dans un dictionnaire, par exemple, nous ne prenons pas d'abord connaissance de la liste des mots et de l'endroit où ils sont définis pour ensuite retrouver la page qui nous intéresse. La plupart des gens utilisent une forme de recherche dichotomique qui leur permet d'examiner une infime quantité d'information contenue dans le dictionnaire avant de trouver le mot qu'ils cherchent. Dans certains problèmes, il sera donc plus naturel de considérer l'accès aux données comme une opération élémentaire supplémentaire, car l'information disponible est démesurément grande pour résoudre le problème qui nous concerne. Pour donner un sens à la complexité des problèmes et des algorithmes, il semble cependant nécessaire de mesurer la taille des instances. Pour cela on sépare l'information disponible en deux parties :

- Les données “véritables”, dont la taille sert de mesure pour la taille de l'instance.
- Un oracle, apte à délivrer des informations complémentaires sur demande. L'information contenue dans l'oracle n'est pas directement prise en compte dans les calculs de complexité

Considérons le problème suivant :

[Dictionnaire] Recherche d'un mot dans un dictionnaire

Données : Un entier n (le nombre de pages du dictionnaire), un mot m .

Oracle : Données: un mot m et un numéro de page t . Résultat: le mot m est-il défini à la page t , avant ou après ?

Résultat : Le numéro de la page où est défini le mot m .

Le problème [Dictionnaire] est soluble en temps $\mathcal{O}(\log(n))$, c'est à dire en temps polynomial dans la taille $|n| = \log(n)$ du nombre n de pages d'un dictionnaire (notons que ce nombre de pages est nécessaire pour démarrer un procédé de recherche dichotomique). Si cet exemple est trivial, d'autres le sont beaucoup moins, comme les algorithmes d'optimisation dans les matroïdes [129], la minimisation des fonctions sous-modulaires [91, 129] ou la méthode de l'ellipsoïde pour résoudre certains

programmes linéaires ayant un grand nombre de contraintes [91].

Evidemment, c'est la structure très particulière de ces problèmes qui permet d'avoir un algorithme polynomial. Faut de reposer sur une structure particulière, la nature du problème nous contraint à poser beaucoup de questions à l'oracle pour avoir suffisamment d'information. On obtient donc souvent des résultats de non-polynomialité (indépendamment de $P \stackrel{?}{=} NP$) pour des problèmes de ce type. Comme exemples citons la taille d'un plus court circuit dans un matroïde [103], la recherche d'un point fixe dans le théorème de Brouwer-Sperner [95], ou encore une affectation dans le cœur d'un jeu coopératif (Section 1.4).

1.2.5 Problèmes numériques

Les problèmes numériques sont ceux dont les données contiennent des nombres, [Factor] est un exemple. Il existe au moins trois modèles permettant d'apprécier l'influence des nombres dans la complexité. Le *modèle binaire* consiste à considérer un nombre comme une donnée binaire quelconque : on considère que le nombre n a une taille $|n| = \log(n + 1)$ dans les données. Par défaut, c'est dans ce modèle que les notions de polynomialité et NP-complétude sont définies.

Dans le *modèle unaire* les nombres sont codés en unaire (i.e. le nombre n est représenté par n bâtons). La taille du nombre n dans les données est alors n . Un *algorithme* (respectivement, un *problème*) polynomial dans le modèle unaire est *pseudo-polynomial*. Un problème NP-dur (respectivement, NP-complet) dans le modèle unaire est *fortement NP-dur* (respectivement, *fortement NP-complet*).

Un troisième modèle de calcul, le *modèle arithmétique* considère la taille de tout nombre comme constante ($= 1$). De plus les quatre opérations $+$, $-$, $*$, $/$ sont considérées comme élémentaires (et prennent un temps constant).

La polynomialité dans le modèle binaire implique la polynomialité dans le modèle unaire, mais la polynomialité du modèle arithmétique est incomparable avec les deux autres modèles [91]. Un algorithme (respectivement, un problème) est *fortement polynomial* s'il est polynomial à la fois dans le modèle binaire et dans le modèle arithmétique.

L'opération partie entière, (i.e. $x \mapsto [x]$) n'est pas polynomiale dans le modèle arithmétique (elle ne peut pas être effectuée avec un nombre constant d'appels aux quatre opérations élémentaires). Un algorithme (respectivement, un problème) est *semi-fortement polynomial* s'il est polynomial à la fois dans le modèle binaire et dans le modèle arithmétique en considérant la partie entière comme une opération élémentaire.

1.2.6 Optimisation et Approximation : “Comment faire bien quand on ne peut pas faire parfaitement”

Pour de nombreux problèmes NP-durs, il existe des algorithmes rapides donnant des solutions satisfaisantes. La théorie de l’approximation permet de juger de la qualité des solutions que fournit un algorithme pour un problème d’optimisation.

Un **problème d’optimisation** est un problème auquel on ajoute un critère pour juger de la qualité des solutions. Par exemple, le problème [Color] peut être transformé en problème d’optimisation :

[Color(opt)] Coloration minimum (version optimisation)

Données : Un graphe $G = (V, E)$.

Résultat : Une k -coloration de G .

Mesure : k , à minimiser.

En passant du problème [Color] au problème [Color(opt)], on augmente l’ensemble des réponses possibles que peut renvoyer l’algorithme (n’importe quelle coloration est maintenant une réponse valide), ce que l’on compense par un critère à optimiser jugeant de la qualité des solutions. Pour comparer deux algorithmes traitant d’un problème d’optimisation, on utilise non seulement leur complexité respectives mais aussi la qualité des solutions qu’ils renvoient. De même que pour la complexité, on juge souvent de la qualité dans le pire cas. Plusieurs critères sont toutefois possibles.

Considérons un problème de minimisation¹² Π , une fonction $\rho : \mathbb{N} \rightarrow \mathbb{R}$ et un algorithme A pour Π . Pour toute instance I , soit $sol_A(I)$ la mesure de la solution renvoyée par A sur I et $opt(I)$ la mesure d’une solution optimale. A est une **ρ -approximation** de Π si :

- A est un algorithme polynomial
- $sol_A(I) \leq opt(I)\rho(|I|)$ pour toute instance I

Un problème est **ρ -approximable** s’il a une ρ -approximation. De nombreux problèmes sont approximables à rapport constant, c’est-à-dire qu’ils sont c -approximable pour une certaine constante c . Le **rapport d’approximation** $\rho^*(\Pi)$ d’un problème Π est l’infimum des $c \in \mathbb{R} \cup \{+\infty\}$ tel qu’il existe une c -approximation pour Π .

Une notion moins courante consiste à mesurer l’écart entre l’optimum et la solution fournie, de manière non pas multiplicative mais additive. A est une **approximation à différence au plus δ** pour Π si

- A est un algorithme polynomial
- $sol_A(I) \leq opt(I) + \delta$ pour toute instance I

On définit $\delta^*(A)$ et $\delta^*(\Pi)$ de manière analogue à $\rho^*(A)$ et $\rho^*(\Pi)$.

¹²Le cas de la maximisation est traitable de manière similaire [123], mais ne sera pas utilisé dans ce mémoire.

L'approximation à différence bornée est particulièrement utile pour les problèmes de coloration des arêtes d'un graphe comme illustré par le Théorème de Vizing.

Théorème 1.7 [129] *Pour tout graphe simple G ,*

$$(1.13) \quad \Delta(G) \leq \chi(L(G)) \leq \Delta(G) + 1$$

De plus une $\Delta(G) + 1$ -coloration de $L(G)$ peut être trouvée en temps polynomial.

Autrement dit, le problème de coloration minimum est approximable à différence au plus 1 dans les line-graphes. D'un autre côté, si $P \neq NP$, comme $[\text{Color}(\text{dec})]$ est NP-complet pour $k = 3$ dans les line-graphes [97, 129], $\delta^*[\text{Color}(\text{Line-graphs})] = 1$ et $[\text{Color}(\text{opt})]$ n'est pas approximable avec un ratio inférieur à $4/3 - \epsilon$ pour $\epsilon > 0$. Quand l'algorithme de Vizing donne une k -coloration, on a une $k/(k-1)$ approximation de $\chi(L(G))$. Comme on peut décider si $\chi(L(G)) = 1$ ou 2, on a en fait $\rho^*[\text{Color}(\text{Line-graphs})] = 4/3$. Le fait que le problème de coloration des arêtes ne soit pas mieux que $4/3$ approximable cache sa très bonne approximabilité pour les graphes ayant un grand degré maximum.

1.2.7 Promesses et robustesse : “Résoudre un problème seulement sur des instances particulières”

Le cadre des promesses donne un sens simple et naturel et précis à la restriction d'un problème à une classe d'instances particulières.

Dans de nombreuses applications des problèmes d'ordonnement chromatique, on sait a priori que le graphe que l'on doit colorier possède certaines propriétés (line-graphe, graphe d'intervalle...). Il apparaît que des phrases comme “résoudre un problème sur les graphes qui ont la propriété p ” prête à confusion au niveau de la théorie de la complexité.

Si la propriété p est facilement décidable (i.e. p est une propriété polynomiale), on peut se permettre de la vérifier avant de résoudre le problème de coloration qui nous intéresse. Les cas ambigus apparaissent quand la propriété p est complexe à reconnaître, car parfois, le problème de coloration est quand même facile à résoudre sur tous les graphes qui ont la propriété p ¹³.

Un survey sur les problèmes de promesses est disponible dans [86]. Une **promesse** est une propriété (vraie ou fausse) définie sur les instances d'un problème. Le résultat d'un problème de promesse est défini seulement sur les instances qui satisfont la promesse. Autrement dit, un algorithme A résout un problème de promesse Π si

- A renvoie quelque chose pour toute instance de Π ,
- A renvoie un résultat (correct) pour toute instance de Π satisfaisant la promesse.

¹³La coloration des graphes parfaitement ordonnables (sous-classe de graphe parfaits dont la reconnaissance est NP-complète) est un bon exemple.

Si un algorithme A résout un problème de promesse Π , le problème spécifiant A est une *réalisation* de Π . Dans le cas particulier des problèmes de décision, un problème de promesse est équivalent à une partition des instances en trois classes : les instances “oui”, les instances “non” et les instances non-spécifiées (pour lesquelles on peut répondre soit oui soit non). Soit **NPP** (respectivement **co-NPP**) la classe des problèmes de promesses qui ont une réalisation NP (respectivement co-NP).

Le concept de problèmes de promesses est dû à Even, Selman et Yacobi [73] qui ont observé que le problème suivant est NP-dur.

[ExSAT] Satisfaisabilité Exclusive

Données : Deux instances I_1 et I_2 de SAT

Promesse : Exactement une des deux instances I_1 et I_2 est satisfaisable

Résultat : I_1 est-elle satisfaisable ?

Commentaires : NP-dur. $\text{NPP} \cap \text{co-NPP}$

En un sens, [ExSAT] semble bien caractérisé : si I_1 est satisfaisable, on le prouve avec une affectation des variables de I_1 , sinon, avec une affectation des variables de I_2 . Argumentons en quoi ce problème *n'est pas* bien caractérisé. [ExSAT] est dans NPP : $\{(I_1, I_2) \mid I_1 \text{ est satisfaisable}\}$ est une réalisation NP. [ExSAT] est dans co-NPP : $\{(I_1, I_2) \mid I_2 \text{ est satisfaisable}\}$ est une réalisation co-NP. Cependant, nous ne connaissons pas de réalisation $\text{NP} \cap \text{co-NP}$ pour [ExSAT] (et comme [ExSAT] est NP-dur, il n'y en a pas à moins que $\text{NP} = \text{co-NP}$). La preuve de NP-difficulté de [ExSAT]¹⁴ suggère que le fait de devoir répondre pour toute instance n'est pas aussi anodin qu'il y paraît à première vue. On pourrait essayer de remettre le paradigme des promesses en question, mais à notre connaissance, il n'existe pas d'autre formalisation pour les questions du genre : “Quelle est la complexité du problème de coloration pour les graphes G tels que $\chi(G) = \omega(G)$ ” (nous traitons cela dans la Proposition 2.5).

La robustesse est un concept informel équivalent à celui des promesses. Le terme *robustesse* semble particulièrement utilisé quand la promesse est dans co-NP¹⁵.

¹⁴Pour résoudre une instance I de [SAT], on considère les deux instances I' et I'' obtenues en fixant la première variable de I à vrai et faux respectivement. Si l'on dispose d'un oracle pour résoudre [ExSAT], on l'applique à (I', I'') (cela permet de déterminer le bon choix pour la première variable, lorsque ce choix s'impose). Dans tous les cas, la réponse de l'oracle permet de choisir une des deux instances entre I' et I'' qui est satisfaisable si et seulement si I l'est. On itère ensuite ce procédé pour éliminer l'ensemble des variables.

¹⁵L'existence de sous-classes de TFNP ayant des problèmes complets est le meilleur indice de la profondeur des questions liées à l'existence d'algorithmes robustes. L'importance de certains problèmes dans la classe TFNP [15, 118, 124, 123] (en particulier, la factorisation des entiers et la recherche de point fixes pour le Théorème de Brouwer-Sperner) incite à une étude plus profonde de ce type de questions. En particulier, la réductibilité entre les problèmes de TFNP et les problèmes de $\text{NP} \cap \text{co-NP}$ semble quasiment incomprise (le problème de la factorisation mis-à-part). Le concept d'*existentially polynomial theorem* [40] est orienté vers les mêmes problématiques, mais semble ne pas avoir atteint la même maturité que l'étude de TFNP.

Cette observation nous incite à proposer la définition (informelle) suivante. Un problème de *robustesse* est un problème de promesse avec un format de résultat alternatif. Un algorithme A résout un problème de promesse Π de manière robuste si

- A renvoie un résultat pour toute instance de Π satisfaisant la promesse.
- A renvoie un résultat ou une alternative pour toute instance de Π .

Par exemple, le Théorème fort des graphes parfaits implique que le problème suivant peut être résolu de manière robuste.

[Colorparf] Coloration des graphes parfaits (version robuste)

Données : Un graphe $G = (V, E)$.

Promesse : G est parfait.

Résultat : Une $\omega(G)$ -coloration de G .

Alternative : un trou impair ou un anti-trou impair de G .

Ce problème est *robuste polynomial*, c'est-à-dire peut être résolu de manière robuste en temps polynomial. En effet, on peut combiner l'algorithme de coloration des graphes parfaits de [91] et l'algorithme de reconnaissance des graphes parfaits de [45], pour toujours donner une réponse valide. Il est cependant envisageable que le problème puisse être résolu de manière plus simple qu'en passant par la reconnaissance des graphes parfaits, comme remarqué dans [40]. Notons par ailleurs qu'il n'est pas nécessaire a priori de savoir reconnaître la promesse pour avoir un algorithme robuste polynomial. La programmation linéaire en nombres entiers sur les polyèdres entiers illustre cette remarque à merveille (voir Section 1.3.2.3).

1.3 Programmes linéaires et relations min-max

1.3.1 Polyèdres, programmation et dualité linéaire

1.3.1.1 Description par des générateurs

Une *combinaison linéaire* d'une famille v_1, v_2, \dots, v_m de vecteurs de \mathbb{R}^n est un vecteur de la forme $v = \lambda_1 v_1 + \lambda_2 v_2 + \dots + \lambda_m v_m$ avec $\lambda_i \in \mathbb{R}$. Si $\lambda_i \geq 0$ pour tout i , alors v est une *combinaison cônica*. Si $\sum_i \lambda_i = 1$, alors v est une *combinaison affine*. Une *combinaison convexe* est une combinaison à la fois affine et cônica. L'*enveloppe convexe*, $\text{conv}(v_1, v_2, \dots, v_m)$ (respectivement, *cônica*, $\text{cone}(v_1, v_2, \dots, v_m)$) est l'ensemble des combinaisons convexes (respectivement, coniques) des vecteurs v_1, v_2, \dots, v_m .

1.3.1.2 Description par des contraintes

Un *demi-espace* de \mathbb{R}^n est un ensemble de la forme : $\{x \mid a^T x \leq b\}$, où $a \in \mathbb{R}^n$ et $b \in \mathbb{R}$. Un *polyèdre* est l'intersection d'un nombre fini de demi-espaces, autrement dit un polyèdre est un ensemble P de la forme $P = \{Ax \leq b\}$ où A est une matrice réelle de taille $m \times n$ et $b \in \mathbb{R}^m$. L'inégalité $c^T x \leq \beta$ est *conséquence* des inégalités $\{Ax \leq b\}$ s'il existe $y \geq 0$ tel que $c^T = y^T A$ et $y^T b \leq \beta$. On considère souvent des systèmes de la forme $\{Ax \leq b, x \geq 0\}$, ce que l'on transforme facilement en système de la forme $\{A'x \leq b'\}$, et réciproquement.

1.3.1.3 Programmation linéaire

La programmation linéaire consiste à optimiser une fonction linéaire sur un polyèdre. Autrement dit, un *programme linéaire* s'écrit sous la forme :

$$(1.14) \begin{aligned} \text{(i)} \quad & \max c^T x \\ \text{(ii)} \quad & Ax \leq b \\ \text{(iii)} \quad & x \geq 0 \end{aligned}$$

Une *solution réalisable* est un vecteur x qui satisfait (1.14)(ii). Une *solution optimale* est une solution réalisable x^* qui est argument du maximum dans (1.14).

1.3.1.4 Farkas-Minkowski-Weil et dualité

Le théorème suivant est habituellement attribué à Motzkin :

Théorème 1.8 (de double représentation des polyèdres) $P \subseteq \mathbb{R}^n$ est un polyèdre si et seulement s'il existe $\{v_1, v_2, \dots, v_t\}$ et $\{w_1, w_2, \dots, w_s\}$ tels que $P = \text{cone}(v_1, v_2, \dots, v_t) + \text{conv}(w_1, w_2, \dots, w_s)$.

Autrement dit, un ensemble est finiment contraint si et seulement s'il est finiment généré. Malgré sa beauté, le Théorème 1.8 ne donne pas d'explications à la solubilité en pratique de la programmation linéaire. En revanche, le lemme suivant donne une caractérisation co-NP de la solubilité d'un système linéaire :

Lemme 1.9 (Farkas) Pour toute matrice réelle A ayant m lignes et tout vecteur $b \in \mathbb{R}^m$, exactement une des deux propriétés suivantes est vraie :

- i) il existe $x \geq 0$ tel que $Ax \leq b$;
- ii) il existe $y \geq 0$ tel que $y^T A \geq 0$ et $y^T b < 0$

De manière analogue, à tout programme linéaire, on peut associer un programme dual dont les solutions optimales donnent une caractérisation NP de l'optimalité des solutions réalisables primales. Par exemple, le *programme dual* de (1.14) est

$$(1.15) \begin{aligned} \text{(i)} \quad & \min b^T y \\ \text{(ii)} \quad & y^T A \geq c \\ \text{(iii)} \quad & y \geq 0 \end{aligned}$$

Le terme de dual vient du fait qu'en passant deux fois au dual, on retrouve le programme originel.

Théorème 1.10 (de dualité de la programmation linéaire) *Pour toute matrice réelle A ayant m lignes et n colonnes, tout $b \in \mathbb{R}^m$ et tout $c \in \mathbb{R}^n$, on a*

$$(1.16) \max\{c^T x \mid Ax \leq b, x \geq 0\} = \min\{b^T y \mid y^T A \geq c, y \geq 0\}$$

pourvu qu'il existe un x et un y réalisables.

Chacun des programmes du Théorème 1.10 a un optimum infini si et seulement si son dual n'a pas de solution réalisable.

1.3.1.5 Complexité de la programmation linéaire

Khachiyan a montré que la méthode de l'ellipsoïde permet de résoudre le problème suivant en temps polynomial (les données sont en composantes rationnelles) :

[LinearProg] Programmation linéaire (Optimisation)

Données : Une matrice A et deux vecteurs b et c .

Résultat : Une solution optimale de $\max\{c^T x \mid Ax \leq b\}$

En fait, l'analyse de l'algorithme de l'ellipsoïde indique qu'il n'est pas nécessaire que le système soit explicitement donné [91, 129]. A la place, il suffit de disposer d'un oracle de séparation : c'est-à-dire, une procédure qui, étant donné un vecteur x , soit affirme que x satisfait le système linéaire, soit fournit une inégalité valide pour le système mais violée par x . D'un autre côté, il est clair que si on peut optimiser pour tout objectif, alors on peut aussi séparer. On obtient donc que "séparer est équivalent à optimiser" [91, 129].

1.3.2 Polyèdres entiers, programmation et dualité entière

1.3.2.1 Programmation linéaire en nombres entiers

Un *programme linéaire en nombres entiers* ou **PLNE**¹⁶ consiste à maximiser une fonction linéaire sur l'ensemble des vecteurs entiers d'un polyèdre. Ce modèle permet de modéliser aisément la plupart des problèmes d'optimisation combinatoire. Cependant, la programmation linéaire en nombres entiers est un problème NP-complet. Plus formellement,

[ILP] Programmation linéaire en nombres entiers

Données : Une matrice A , deux vecteurs b, c et $k \in \mathbb{Q}$

Résultat : Est-ce que $\max\{c^T x \mid Ax \leq b, x \in \mathbb{N}^m\} \geq k$?

Commentaires : NP-complet, même si $c = \mathbf{1}$ et A n'a qu'une contrainte [128].

Notons qu'optimiser une fonction linéaire sur un ensemble X est équivalent à optimiser cette fonction sur l'enveloppe convexe¹⁷ de X :

Proposition 1.11 *Pour $X \subseteq \mathbb{R}^n$ et $c \in \mathbb{R}^n$, on a*

$$(1.17) \quad \sup\{c^T x \mid x \in X\} = \sup\{c^T x \mid x \in \text{conv}(X)\}$$

Pour un polyèdre $P \subseteq \mathbb{R}^n$, notons P_I l'enveloppe convexe de ses points entiers.

$$(1.18) \quad P_I := \text{conv}(P \cap \mathbb{Z}^n)$$

Malgré (ou plutôt grâce à) sa difficulté, la programmation linéaire en nombres entiers a participé au développement de riches théories. Nous en survolons quelques-unes dans les sections suivantes.

1.3.2.2 Dualité faible et relaxation linéaire

Proposition 1.12 (Dualité faible pour les PLNE) *Pour toute matrice A et tous vecteurs b, c , on a*

$$(1.19) \quad \max\{c^T x \mid Ax \leq b, x \in \mathbb{N}^n\} \leq \min\{b^T y \mid y^T A \geq c, y \in \mathbb{N}^m\}$$

Il y a un *saut de dualité*¹⁸ lorsque l'inégalité (1.19) est stricte. Pour prouver la dualité faible, on utilise généralement le concept suivant, qui permet de prouver un

¹⁶ *integer linear program* ou *ILP*.

¹⁷ En fait, la linéarité de la fonction objectif n'est pas nécessaire dans la Proposition 1.11 car l'observation se généralise à la maximisation d'une fonction convexe.

¹⁸ *duality gap*.

peu plus. Pour un programme linéaire en nombres entiers,

$$(1.20) \quad \max\{c^T x \mid Ax \leq b, x \in \mathbb{N}^n\}$$

le **programme linéaire relaxé** associé, $\max\{c^T x \mid Ax \leq b, x \geq 0\}$ est obtenu en oubliant la contrainte d'intégralité des solutions. La Proposition 1.13 donne les relations entre programmes entiers et relaxés.

Proposition 1.13 *Pour toute matrice A et tous vecteurs b, c , on a*

$$(1.21) \quad \max\{c^T x \mid Ax \leq b, x \in \mathbb{N}^n\} \stackrel{(a)}{\leq} \max\{c^T x \mid Ax \leq b, x \geq 0\} \\ \stackrel{(b)}{=} \min\{b^T y \mid y^T A \geq c, y \geq 0\} \stackrel{(c)}{\leq} \min\{b^T y \mid y^T A \geq c, y \in \mathbb{N}^m\}$$

Preuve

- (a) Lors de la relaxation, on maximise sur un ensemble plus grand ($\mathbb{N}^n \subset \mathbb{R}_+^n$) et la solution du programme relaxé est donc supérieure ou égale à celle du programme en nombres entiers.
- (b) est le Théorème de dualité de la programmation linéaire.
- (c) L'argument est similaire à celui de (a).

□

S'il est aisé d'obtenir une borne inférieure pour (1.20) (car tout x entier tel que $Ax \leq b$ donne une telle borne), il est moins trivial d'obtenir une borne supérieure. C'est l'une des motivations pour étudier le programme

$$(1.22) \quad \min\{b^T y \mid y^T A \leq c, y \geq 0\}$$

En effet, toute solution de (1.22) donne une borne supérieure à l'optimum de (1.20). En particulier, on peut résoudre (1.22) à l'optimum en temps polynomial (au moins lorsque A, b et c sont donnés de manière explicite). Toutefois la qualité d'une telle borne supérieure dépend du système considéré car l'écart entre (1.20) et (1.22) peut s'avérer arbitrairement grand. Il est souvent intéressant d'interpréter le programme (1.22) avec des contraintes d'intégralité : lorsque toutes les inégalités de (1.21) sont serrées, on obtient une relation min-max. Dans les prochaines sections, nous discutons les théories associées aux cas où l'une, voire les deux inégalités de (1.21) sont serrées.

1.3.2.3 Polyèdres entiers

Un **polyèdre** est **entier** s'il est égal à l'enveloppe convexe des points entiers qu'il contient. Autrement dit, P est entier si $P = P_I$. Le Théorème suivant, dû à Edmonds et Giles [128, 129] caractérise les polyèdres entiers comme ceux pour lesquels l'inégalité (1.21)(a) est serrée indépendamment de la fonction objectif. Plus précisément :

Théorème 1.14 Soit $P = \{x \mid Ax \leq b, x \geq 0\}$. Les conditions suivantes sont équivalentes

- i) P est un polyèdre entier.
- ii) $\max\{c^T x \mid x \in P\}$ est atteint par un point entier pour tout $c \in \mathbb{R}^n$ (pourvu que le maximum soit fini).
- iii) La valeur de $\max\{c^T x \mid x \in P\}$ est entière pour tout $c \in \mathbb{N}^n$ (pourvu que le maximum soit fini).

On peut résoudre la programmation linéaire en nombres entiers en temps polynomial quand le système d'inégalités définit un polyèdre entier (cela se réduit essentiellement à résoudre le programme linéaire relaxé).

[ILPinIntegerPolyhedra] PLNE dans les polyèdres entiers

Données : Une matrice A , deux vecteurs b et c .

Promesse : $\{x \mid Ax \leq b, x \geq 0\}$ est un polyèdre entier (non-vide et borné).

Résultat : Une solution optimale entière de $\max\{c^T x \mid Ax \leq b, x \geq 0\}$.

Alternative : Un sommet non-entier de $\max\{c^T x \mid Ax \leq b, x \geq 0\}$ (ou une inégalité conséquence de $\{Ax \leq b, x \geq 0\}$ montrant que ce système n'a pas de solution)

Commentaires : Robuste Polynomial [128].

Malheureusement, aucune méthode générale n'est connue pour reconnaître les polyèdres entiers, et pour cause : reconnaître les polyèdres entiers est un problème fortement co-NP-complet [126]. Nous proposons quelques raffinements du résultat de [126] et de sa preuve. Le problème que nous considérons est :

[IntegerPolyhedra] Reconnaissance des polyèdres entiers

Données : Une matrice A et un vecteur b

Résultat : Est-ce que $\{x \mid Ax \leq b\}$ est un polyèdre entier ?

Théorème 1.15 [IntegerPolyhedra] est un problème co-NP-complet, même si l'on se restreint à un système de la forme

$$(1.23) \quad \{A_1 x \leq 0, A_2 x \geq 0, a_3^T x = 1\}$$

où les matrices A_1, A_2 et le vecteur a_3 sont à coefficients $\{0, 1\}$.

Papadimitriou et Yannakakis [126] commentent qu'il aurait été surprenant que l'on puisse reconnaître les polyèdres entiers car cela aurait permis de savoir pour quels systèmes on peut utiliser une technique de programmation linéaire pour résoudre un PLNE *indépendamment de la fonction objectif*. Notons que pour une fonction objectif donnée, c'est non-seulement l'existence d'un optimum entier, mais aussi l'optimisation sous promesse d'optimalité entière qui sont NP-dures :

[IntegerVertex] Sommet entier d'un système linéaire**Données :** Une matrice A , un vecteur b **Résultat :** Est-ce que $\{x \mid Ax \leq b\}$ a un sommet entier ?**[ILPinIntegerVertex]****Données :** Une matrice A , deux vecteurs b, c **Promesse :** $\max\{c^T x \mid Ax \leq b, x \geq 0\}$ est atteint par un sommet entier**Résultat :** Un sommet optimal entier pour $\max\{c^T x \mid Ax \leq b, x \geq 0\}$.**Théorème 1.16** *[IntegerVertex] est NP-complet et [ILPinIntegerVertex] est NP-dur, même si l'on se restreint à un système de la forme*

$$(1.24) \quad \{A_1 x \leq 0, A_2 x \geq 0, a_3^T x = 1\}$$

et que les matrices A_1, A_2 et les vecteurs a_3 et c sont à coefficients $\{0, 1\}$.

Le reste de la Section 1.3.2.3 est consacré aux preuves des Théorèmes 1.15 et 1.16, dont nous commençons par donner les grandes lignes :

Idée de preuve L'existence d'un circuit Hamiltonien dans un digraphe se réduit aux problèmes considérés. Pour cela, on associe un polytope P à un digraphe D de manière à ce que les sommets de P soient en bijection avec les circuits de D . On construit d'abord P de manière à ce que seuls les sommets de P associés aux circuits Hamiltoniens de D ne soient pas des vecteurs entiers. Une construction similaire permet que seuls les circuits Hamiltoniens donnent des vecteurs entiers. \square

Soit $D = (V, A)$ un digraphe. Une *circulation* sur D est une fonction $x : A \rightarrow \mathbb{R}_+$ telle que

$$(1.25) \quad x(\delta^{in}(v)) - x(\delta^{out}(v)) = 0 \quad \text{pour tout } v \in V$$

Autrement dit, x est une circulation si pour chaque sommet du graphe, la somme des valeurs entrantes est égale à la somme des valeurs sortantes. L'ensemble des circulations étant défini par un ensemble fini de contraintes linéaires, c'est un cône polyédral. L'ensemble des générateurs de ce cône est donné par un résultat de base en théorie des flots (voir [129], Théorème 11.1).**Proposition 1.17** *Les rayons extrêmes du cône des circulations d'un digraphe D sont les vecteurs caractéristiques des circuits de D (à multiplication par un scalaire près).*Notons \mathcal{C}_D le cône des circulations de D et pour $\alpha > 0$ considérons le polyèdre

$$(1.26) \quad \mathcal{C}_D^\alpha := \mathcal{C}_D \cap \{x \in \mathbb{R}^A \mid \mathbf{1}^T x = \alpha\}$$

Une conséquence immédiate de la Proposition 1.17 est :

Proposition 1.18 *Pour tout digraphe D et tout $\alpha > 0$, l'ensemble des circuits de D est en bijection avec les sommets du polyèdre \mathcal{C}_D^α de la manière suivante :*

$$C \longleftrightarrow \alpha \frac{\mathbf{1}_C}{|C|}$$

(où $\mathbf{1}_C$ est le vecteur caractéristique du circuit C). □

Les preuves des Théorèmes 1.15 et 1.16 utilisent une réduction à partir du problème suivant.

[HamiltonOriented] Circuit Hamiltonien dans les digraphes

Données : Un digraphe simple $D = (V, A)$

Résultat : D a-t-il un circuit Hamiltonien ?

Commentaires : NP-complet.

Il nous faut toutefois supposer que le digraphe utilisé dans la réduction a un nombre premier de sommets. La Proposition 1.19 montre que cette supposition ne rend pas le problème d'Hamiltonicité plus facile.

[HamiltonOrientedPrime]

Données : Un digraphe simple $D = (V, A)$

Promesse : $|V|$ est un nombre premier.

Résultat : D a-t-il un circuit Hamiltonien ?

Proposition 1.19 *[HamiltonOrientedPrime] est NP-dur.*

Preuve On réduit [HamiltonOriented] à [HamiltonOrientedPrime]. Soit $D = (V, A)$ un digraphe. Soit p un nombre premier¹⁹ tel que $|V| \leq p < 2|V|$. On construit un digraphe D' qui a p sommets et qui est Hamiltonien si et seulement si D l'est. Si $|V|$ est premier, on prend $D' := D$. Sinon, soit $v \in V$ un sommet arbitraire. D' est obtenu en éclatant v en deux sommets (l'un contenant les arcs de $\delta_D^{in}(v)$ et l'autre contenant les arcs $\delta_D^{out}(v)$). Les deux sommets sont liés par un chemin P_t ayant $t := p - |V|$ arcs (voir Figure 1.1). □

Proposition 1.20 *Soit $D = (V, A)$ un digraphe ayant un nombre premier $p = |V|$ de sommets.*

i) $\mathcal{C}_D^{(p-1)!}$ est un polyèdre entier si et seulement si D n'est pas Hamiltonien.

ii) \mathcal{C}_D^p a un sommet entier si et seulement si D est Hamiltonien.

Preuve Les coordonnées non-nulles des sommets de \mathcal{C}_D^α sont de la forme $\alpha/|C|$ où C est un circuit de D . Il suffit de noter que pour tout circuit de D :

¹⁹un tel nombre existe toujours [3] (et on peut le trouver en temps polynomial par recherche exhaustive, car on veut trouver p (et certifier sa primalité) en temps polynomial en $|V|$ et non en $\log(|V|)$).

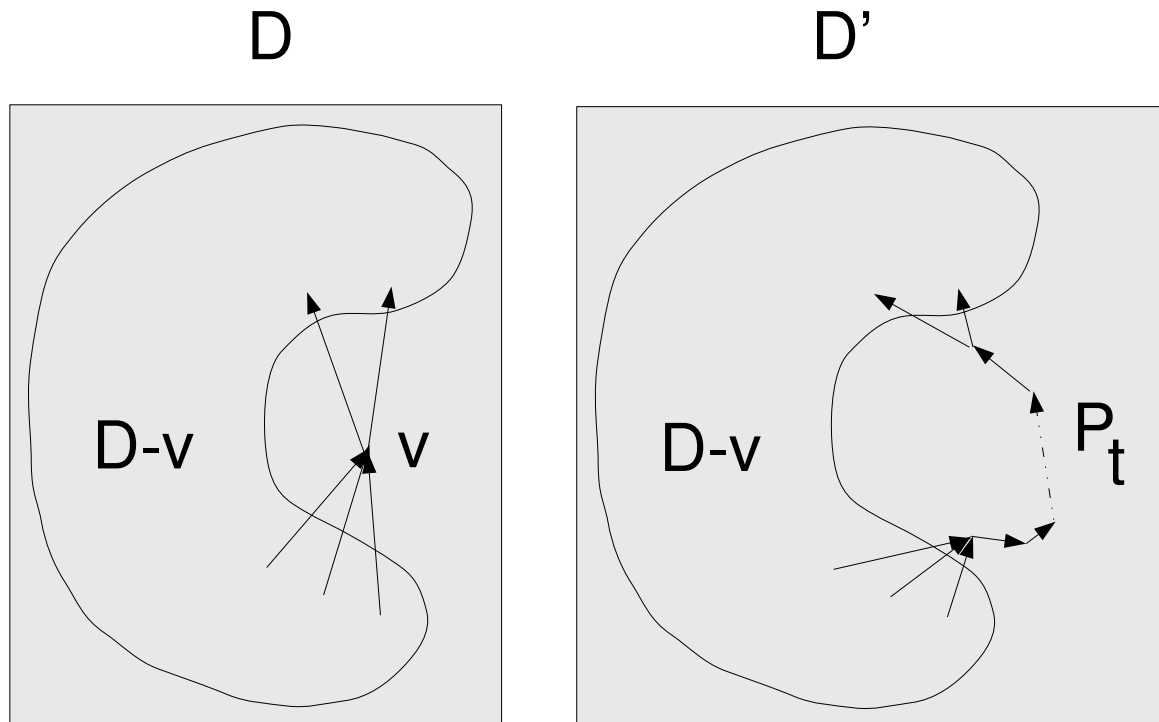


Figure 1.1 – Réduction de [HamiltonOriented] à [HamiltonOrientedPrime]

- i)* $(p-1)!/|C|$ est entier si et seulement si $|C| \leq p-1$, c'est à dire si C n'est pas un circuit Hamiltonien.
- ii)* $p/|C|$ est entier si et seulement si $|C| = p$, c'est à dire si C est un circuit Hamiltonien.

□

Preuve du Théorème 1.15 Le système suivant décrit le polyèdre \mathcal{C}_D^α :

$$\begin{aligned}
 (1.27)(i) \quad & x(\delta^{in}(v)) - x(\delta^{out}(v)) = 0 && \text{pour tout } v \in V \\
 (ii) \quad & \mathbf{1}^T x = \alpha \\
 (iii) \quad & x \geq 0
 \end{aligned}$$

La Proposition 1.20 *i)* donne donc la co-NP-complétude de [IntegerPolyhedra] (la réduction est polynomiale car la taille du nombre $(p-1)!$ est $\log((p-1)!) \leq p \log(p)$). Les deux astuces suivantes permettent de se ramener à un système en coefficients $\{0, 1\}$.

– Pour $v \in V$ on pose une nouvelle variable y_v et on remplace la contrainte (1.27)(i)

associée à v par deux nouvelles contraintes :

$$(1.28) \quad x(\delta^{in}(v)) + y = 0 \text{ et } x(\delta^{out}(v)) + y = 0$$

- On peut remplacer la contrainte (1.27)(ii) par le système suivant qui force la contrainte $z_{i,j} = (-1)^i \cdot i!$ pour toute paire i, j (et donc $z_{p-1,1} = i!$ pour $p \neq 2$ premier).

$$(1.29) \begin{aligned} \text{(i)} \quad & w = 1 \\ \text{(ii)} \quad & w + z_{1,j} = 0 \quad \text{pour } j \in \{1, 2\} \\ \text{(iii)} \quad & z_{i+1,j} + \sum_{k=1}^{i+1} z_{i,k} = 0 \quad \text{pour tout } i \in [1..p-2] \text{ et tout } j \in [1..i+1] \\ \text{(iv)} \quad & z_{p-1,1} + \sum_{a \in A} x_a = 0 \end{aligned}$$

□

La démonstration du Théorème 1.16 se termine de manière équivalente à celle du Théorème 1.15. Notons que l'on peut facilement adapter la preuve des Théorèmes 1.16 et 1.15 pour changer la contrainte $a^T x = 1$ par une inégalité dans un sens ou dans l'autre.

1.3.2.4 Systèmes TDI et box-TDI

Un **systeme** $Ax \leq b$ est **TDI**²⁰ si pour tout c entier, le minimum dans l'équation de dualité

$$\min\{y^T b \mid y^T A = c, y \geq 0\} = \max\{c^T x \mid Ax \leq b\}$$

est atteint par un vecteur y entier, dès que ce minimum est fini. De la caractérisation (iii) du Théorème 1.14, on déduit donc :

Corollaire 1.21 *Si $Ax \leq b$ est un système TDI et b est entier, alors $\{x \mid Ax \leq b\}$ est un polyèdre entier.*

De plus tout polyèdre entier peut être décrit par un système TDI. Un problème ouvert majeur en combinatoire polyédrale est de reconnaître (en temps polynomial) les systèmes TDI. En effet, la plupart des résultats de combinatoire polyédrale (voir [129]) consistent à établir qu'un système linéaire représentant un problème combinatoire est TDI. Une compréhension commune de ces phénomènes (au travers d'un algorithme de reconnaissance polynomial) serait sans doute une avancée majeure. Edmonds et Giles (cf. [128]) conjecturent cependant que cette reconnaissance est co-NP-complète. Notons de plus que de nombreux systèmes TDI ont une taille

²⁰*totally dual integral.*

exponentielle dans la taille du problème combinatoire qu'ils représentent. L'éventuelle polynomialité de la reconnaissance des systèmes TDI, n'implique donc pas directement la polynomialité de la reconnaissance de cette structure sous-jacente. Les graphes parfaits sont un bon exemple de structure décrivant un système TDI de taille exponentielle dans la taille du graphe. Notons que la reconnaissance des graphes parfaits est connue pour être polynomiale [45].

De nombreux systèmes TDI le restent quand on ajoute des bornes sur les valeurs des variables. Un **système** $Ax \leq b$ est **box-TDI** si le système $\{Ax \leq b, d_1 \leq x \leq d_2\}$ est TDI pour tous $d_1, d_2 \in \mathbb{R}^n$.

1.3.2.5 Matrices donnant des polyèdres entiers

L'existence de solutions optimales entières d'un programme linéaire s'explique parfois à partir de la matrice de contraintes.

Une matrice A est **totalelement unimodulaire** ou **TTU** si toute sous-matrice carrée de A a un déterminant dans $\{-1, 0, +1\}$. Cette propriété est fondamentale en optimisation combinatoire, elle admet plusieurs autres définitions (notons qu'une matrice TTU est toujours à coefficients dans $\{-1, 0, +1\}$) :

Théorème 1.22 [128] *Soit A une matrice à coefficients dans $\{-1, 0, +1\}$, les conditions suivantes sont équivalentes :*

- i) A est totalelement unimodulaire.*
- ii) Le polyèdre $\{x \mid Ax \leq b, x \geq 0\}$ est entier pour tout vecteur entier b .*
- iii) Le système $Ax \leq b$ est box-TDI pour tout vecteur entier b .*
- iv) Chaque sous-ensemble de lignes de A peut être partitionné en deux groupes tels que la somme des lignes d'un groupe moins la somme des lignes de l'autre est un vecteur à coefficients dans $\{-1, 0, +1\}$.*

iv) est le **critère de Ghouila-Houri**.

Voyons quelques exemples importants de matrices TTU. Une **chaîne** de sous-ensembles d'un ensemble V est une famille $V_1 \subset V_2 \subset \dots \subset V_t$ avec $V_i \subseteq V$ pour tout i . Une **famille** $V_1, V_2 \dots V_t$ est **laminaire** si $V_i \cap V_j = \emptyset$ ou $V_i \subseteq V_j$ pour tout i, j . Notons que les familles laminaires généralisent à la fois les partitions et les chaînes. Une **matrice** à coefficients $\{0, 1\}$ est **d'intervalles** si pour chaque ligne, tous les 1 sont consécutifs.

1.3.2.6 Systèmes TDAU

Une méthode extrêmement puissante qui permet de prouver les relations min-max utilise les matrices TTU de manière plus fine qu'en regardant simplement la matrice

de contraintes (voir le Théorème 5.35 de [129]).

Soit A une matrice $m \times n$ entière. Le système $Ax \leq b$ est **totalelement dualement activement unimodulaire**²¹ ou **TDAU** si pour tout $c \in \mathbb{R}^n$, $\max\{c^T x \mid Ax \leq b\}$ a une solution duale optimale $y \in \mathbb{R}_+^m$ (lorsque l'optimum est fini) telle que les lignes de la matrice A correspondant aux composantes non-nulles de y forment une matrice totalement unimodulaire.

Théorème 1.23 [129, 128] *Un système TDAU est box-TDI.*

Un des intérêts d'énoncer qu'un système est TDAU plutôt que box-TDI est qu'il est parfois plus aisé de manipuler les opérations de compositions sur les matrices totalement unimodulaires pour dériver de nouveaux cas de systèmes TDAU (cela pourrait être utile pour prolonger les résultats des Sections 4.2 et 5.2). Un autre intérêt est que certaines classes de systèmes linéaires semblent être box-TDI si et seulement si ils sont TDAU (Sections 4.2 et 5.2). Une telle équivalence pourrait aider à la reconnaissance de ces systèmes box-TDI car on dispose de caractérisations puissantes des matrices TTU [128] (particulièrement depuis le Théorème de décomposition de Seymour).

1.3.2.7 Plans coupants

Soit $P = \{x \mid Ax \leq b\}$ un polyèdre. Soit c un vecteur entier et $\delta \in \mathbb{Q}$. Clairement, si $P \subseteq \{x \mid c^T x \leq \delta\}$ ²² alors $P_I \subseteq \{x \mid c^T x \leq \lfloor \delta \rfloor\}$. L'inégalité $\{x \mid c^T x \leq \lfloor \delta \rfloor\}$ est un **plan coupant**²³. Gomory et Chvátal [128] ont observé que pour tout polyèdre $P = \{x \mid Ax \leq b\}$, on peut obtenir un ensemble de contraintes décrivant P_I en calculant itérativement des plans coupant à partir du système $\{x \mid Ax \leq b\}$.

1.3.3 Quelques théorèmes de combinatoire polyédrale

1.3.3.1 Petite philosophie à l'usage des (non)-spécialistes : Combinatoire polyédrale et relations min-max

La **combinatoire polyédrale** consiste dans la mise en œuvre du paradigme suivant²⁴ :

L'ensemble des solutions d'un problème d'optimisation combinatoire peut souvent être représenté par l'ensemble des arêtes d'un hypergraphe $\mathcal{H} = (V, \mathcal{E})$ et donc

²¹C'est un néologisme.

²²c'est-à-dire si $c^T x \leq \delta$ est conséquence de $Ax \leq b$.

²³aussi appelé **coupe de Gomory-Chvatal**.

²⁴Nous donnons ici une vision caricaturale, car ce domaine nous semble difficile à définir de manière à la fois simple et précise.

par l'ensemble $X \subseteq \mathbb{R}^V$ des vecteurs caractéristiques des éléments de \mathcal{E} . L'enveloppe convexe de X peut être décrite par des générateurs (i.e. par des combinaisons convexes d'éléments de X). En vertu du Théorème de double représentation des polyèdres, $\text{conv}(X)$ peut aussi être décrite par des contraintes (i.e. les facettes de $\text{conv}(X)$). La combinatoire polyédrale consiste à décrire ces contraintes, pour acquérir une compréhension "duale" du problème étudié.

Ce choix de présentation peut laisser penser qu'il existe un plongement naturel d'une famille X d'objets combinatoires dans un espace linéaire et incite à chercher les facettes (voire un système TDI) décrivant l'enveloppe convexe de X . Il faut toutefois tempérer l'idée que cette représentation est canonique car il existe plusieurs manières de plonger X dans un espace linéaire. L'espace des flots par exemple peut être décrit par des variables sur les arêtes, mais aussi par des variables sur les chemins. C'est à partir de cette deuxième formulation que Lehman et Seymour ont donné des généralisation de la propriété "flot-max-coupe-min" pour certaines classes d'hypergraphes [129, 49]. Le problème de la coloration se prête aussi à des représentations dans différents espaces. Or si choisir un bon espace pour plonger les objets combinatoires peut s'avérer une question primordiale, il ne semble pas exister de méthode générale pour effectuer ce choix. Pire, les nouvelles formulations semblent souvent être découvertes "par hasard". A l'heure actuelle, la mise en œuvre de la combinatoire polyédrale s'appuie non-seulement sur la méthode scientifique, mais aussi sur la création artistique.

Une *relation min-max* est une formule du type

$$\min\{f(x) \mid x \in X\} = \max\{g(y) \mid y \in Y\}$$

où X, Y sont des ensembles et f et g des fonctions à valeurs réelles. Un des intérêts d'avoir une formule min-max pour le problème $\min\{f(x) \mid x \in X\}$ est que l'on peut certifier qu'un $x \in X$ est un argument du minimum : il suffit de donner un y maximum "dual". Historiquement²⁵, la combinatoire polyédrale et l'étude des relations min-max sont intimement liées. En effet, la plupart des relations min-max peuvent être prouvées en plongeant les éléments de X dans un certains espace \mathbb{R}^n puis en interprétant (Y, g) comme un ensemble de contraintes décrivant $\text{conv}(X)$ ²⁶. Evidemment pour qu'une telle approche aboutisse il semble *nécessaire de se restreindre* à des fonctions objectifs f linéaires²⁷. De plus, pour que la relation min-max prenne

²⁵Cela ne semble pas démenti aujourd'hui.

²⁶cela se ramène souvent à vérifier que (X, f) décrit un ensemble de contraintes valides pour $\text{conv}(Y)$ puis à prouver que le PLNE ainsi formé a de bonnes propriétés d'intégralités (polyèdres entier, système TDI...). Mais l'intégralité du polyèdre ainsi défini n'est parfois pas suffisante. Un exemple est donné par la μ -coloration (le graphe P_4 page 228 donne une obstruction à la μ -colorabilité bien que le système (5.11) soit box-TDI et réalisable pour ce graphe). Un autre exemple est donné par le TSP graphique (le graphe M_1 page 68 admet, pour le système (1.61), une solution entière qui n'est pas dans l'enveloppe convexe des tours de Steiner).

²⁷La convexité de f garantit que l'optimalité locale implique l'optimalité globale, ce qui mène à des théories de dualité et permet d'appliquer des paradigmes de résolution comme la méthodes

tout son sens dans le cadre de la combinatoire polyédrale (et que l'on puisse appliquer des paradigmes comme l'équivalence séparation-optimisation [91]), il semble *nécessaire que la relation min-max soit satisfaite* pour tout f linéaire. Le sens de cette linéarité dépend encore une fois grandement de l'espace dans lequel X est plongé. Deux plongements différents pour les colorations sont discutés dans la Section 1.3.3.4 et un troisième plongement est discuté dans la Section 5.1.1. Chaque plongement permet une plus ou moins grande expressivité (*i.e.* une palette de problèmes “pondérés” associés) grâce à l'espace des fonctions de coûts linéaires qu'il impose implicitement. Bien évidemment, une plus grande expressivité se traduit par une plus grande complexité algorithmique. L'aspect artistique de la combinatoire polyédrale consiste donc à imaginer des plongements qui permettent un compromis entre l'expressivité et la complexité algorithmique. Cette philosophie est illustrée pour les problèmes d'ordonnancement chromatiques dans les Sections 4.2, 4.4.2, 5.1.1, 5.2, et 5.3.

Pour certaines relations min-max, on peut donner une formulation polyédrale dans laquelle la relation min-max s'exprime à l'aide du Théorème de dualité, mais dont les versions pondérées définies par cette formulation ne sont pas valides. On peut parfois “réparer” la formule min-max pondérée d'un des côtés (par exemple $\max\{f(x) \mid x \in X\}$) en ajoutant des inégalités valides (ce faisant, on modifie l'ensemble Y). Mais parfois, les versions pondérées sont NP-dures et la formulation linéaire ne semble pas d'un grand intérêt pour l'étude de la formule min-max en question (voir Sections 3.4.3 et 4.4.2 pour un tel exemple). Rien ne nous empêche cependant de présenter une formule min-max dans le langage de la combinatoire polyédrale : un programme linéaire a la **propriété de dualité entière**²⁸, si lui et son dual ont une solution optimale entière.

1.3.3.2 Couplages

Le cas d'école de la combinatoire polyédrale est le problème des couplages dans les graphes. Soit $G = (V, E)$, un graphe. Plongeons les couplages de $G = (V, E)$ dans \mathbb{R}^E . Plus précisément, considérons l'ensemble $\mathcal{M}(G)$ des vecteurs caractéristiques des couplages de $G = (V, E)$. Les inégalités suivantes sont valides pour $\mathcal{M}(G)$ et donc pour $\text{conv}(\mathcal{M}(G))$.

$$(1.30)(i) \quad x_e \geq 0 \quad \text{pour tout } e \in E$$

$$(ii) \quad x(\delta(v)) \leq 1 \quad \text{pour tout } v \in V$$

des ellipsoïdes [91]. La concavité garantit en revanche que l'optimum est atteint sur un sommet de $\text{conv}(X)$ et donc par un élément de X .

²⁸c'est un néologisme.

En effet, (ii) traduit le fait qu'un sommet est adjacent à au plus une arête d'un couplage. x est le vecteur d'incidence d'un couplage si et seulement s'il est entier et satisfait (ii). Si le graphe est biparti, la matrice de contraintes du système (1.30) est totalement unimodulaire [129] et on en déduit une formule min-max pour le poids maximum d'un couplage.

Si le graphe n'est pas biparti, le système (1.30) ne décrit pas complètement $\text{conv}(\mathcal{M}(G))$. Pour un triangle (K_3) par exemple, le vecteur constant $\mathbf{1}/2$ vérifie (1.30) mais n'est pas dans l'enveloppe convexe des couplages.

On peut quand même s'en sortir en ajoutant des plans coupants : le système suivant est TDI pour tout graphe.

$$\begin{aligned}
 (1.31)(i) \quad & x_e \geq 0 && \text{pour tout } e \in E \\
 (ii) \quad & x(\delta(v)) \leq 1 && \text{pour tout } v \in V \\
 (iii) \quad & x(E[U]) \leq \left\lfloor \frac{1}{2}|U| \right\rfloor && \text{pour tout } U \subseteq V
 \end{aligned}$$

Des extensions aux c -couplages ainsi qu'une formulation équivalente pour la couverture par des arêtes sont présentées dans [129].

1.3.3.3 Clique maximum et inégalité de stables

Pour décrire les cliques d'un graphe avec la méthode polyédrale, on peut tenter une approche similaire à celle que nous avons décrite pour les couplages en étudiant le polytope de \mathbb{R}_+^V suivant :

$$P_{\text{clique}}(G) := \text{conv}(\{\text{vecteurs caractéristiques des cliques de } G\})$$

Mais tout le travail reste à faire. Et comme le problème de la clique maximum est NP-dur, le polytope des cliques est défini par des *contraintes difficiles* [129], c'est-à-dire dont on ne peut pas prouver la validité en temps polynomial à moins que $\text{NP} = \text{co-NP}$. Cependant, certaines inégalités *simples* (*i.e.* dont on peut certifier la validité en temps polynomial) sont suffisantes pour de larges classes de graphes :

$$\begin{aligned}
 (1.32)(i) \quad & x_v \geq 0 && \text{pour tout } v \in V \\
 (ii) \quad & x(S) \leq 1 && \text{pour tout stable } S \text{ de } G
 \end{aligned}$$

Bien qu'il soit NP-dur de séparer sur le système 1.32, les contraintes (1.32)(ii) sont quand même des contraintes simples, car il est facile de vérifier que $S \subseteq V$ est

bien un stable. La théorie des graphes parfaits traite des graphes pour lesquels le système (1.32) a de bonnes propriétés d'intégralité (voir Théorème 2.4, page 101). Le problème dual associé au système (1.32) s'interprète en termes de coloration et est discuté dans la section suivante.

1.3.3.4 Mais où est passé le polytope des colorations ?

Contrairement aux couplages et même aux cliques, il n'existe pas de manière "naturelle" de plonger les colorations dans un espace vectoriel. Cependant, il existe plusieurs formulations intéressantes. Nous en présentons deux dans cette section. La première formule la coloration comme un problème de couverture (ou de partition) des sommets d'un graphe par des stables. C'est la formulation la plus connue et celle qui semble avoir les meilleures propriétés algorithmiques pour le problème de la coloration usuelle. C'est elle qui sert de base dans la deuxième partie de ce mémoire. La deuxième formulation exprime les partitions en cliques de $G = (V, E)$ en termes des stables d'un graphe auxiliaire, qui a E comme ensemble de sommets. Le potentiel de cette nouvelle formulation reste quasiment inexploré, bien qu'elle ait déjà fait ses preuves empiriquement pour la coloration usuelle des graphes "denses" [47] et que son expressivité permette de modéliser le problème [Max-coloring] (Section 4.4.2).

Couverture et partition par des stables La formulation classique pour le problème de la coloration minimum consiste à choisir l'ensemble des stables (pas seulement maximum par inclusion) de G comme variables. On exprime alors la coloration comme un problème de partition des sommets en stables :

$$(1.33)(i) \quad x_S \geq 0 \quad \text{pour tout } S \text{ stable de } G$$

$$(ii) \quad x(v) = 1 \quad \text{pour tout } v \in V$$

Le programme $\min \mathbf{1}^T x$ sujet à (1.33) a la propriété de dualité entière pour tout graphe satisfaisant $\chi = \omega$, mais n'est (box-)TDI que pour les graphes co-bipartis. Pour certaines classes de fonctions objectifs sous-modulaires et certaines classes de graphes parfaits, le système a aussi la propriété de dualité entière (voir Section 4.2). En particulier résoudre le problème de coloration minimum, revient à choisir la fonction objectif $\mathbf{1}$. Dans tous les cas, c'est le dual de (1.33) qui décrit un système TDI et qui exprime les formules min-max pour le coût minimum d'une coloration de G de manière plus élégante.

Forêts clique-connectantes L'idée suivante a été proposée par Denis Cornaz [47] : soit $\mathcal{B} = \{B_1, B_2, \dots, B_k\}$ une partition en cliques de G (\mathcal{B} est donc

une coloration de \overline{G}). Soit G' l'union disjointe des graphes $G[B_i]$, pour $i \in [1..k]$. G' est une union disjointe de cliques et a k composantes connexes. Toute forêt maximale par inclusion de G' est composée de $|V| - k$ arêtes. Cela motive la définition suivante : une forêt $F \subseteq E$ de G est **clique-connectante** si l'ensemble des sommets de chacune de ses composantes connexes induit une clique de G . Soit $\varphi(G)$ la cardinalité maximum d'une forêt clique-connectante.

Proposition 1.24 [47] *Pour tout graphe G , on a $\overline{\chi}(G) + \varphi(G) = |V|$.*

On peut alors plonger les forêts clique-connectantes dans \mathbb{R}^E et étudier leur enveloppe convexe. Toutefois, les symétries de la modélisation pénalisent l'efficacité de l'implémentation [47]. En effet, en considérant seulement les chemins Hamiltoniens, on voit qu'une clique de taille n peut être décrite par plus que $n!/2$ forêts différentes. Plusieurs restrictions permettent d'éliminer ces symétries [47]. L'astuce qui semble la plus prometteuse nous amène à la définition suivante : une forêt (clique-connectante) est **stellaire** si tous ses arbres sont des étoile [47].

Bien qu'il existe plusieurs forêts stellaires définissant une partition en cliques, nous avons observé que l'utilisation d'un ordre total sur V permet d'obtenir une bijection. Soit \prec un ordre total sur V . Une étoile E_0 de G est bien ordonnée si son sommet central est le plus petit (selon \prec) parmi les sommets de $V(E_0)$. Une forêt est **bien ordonnée** si tous ses arbres sont bien ordonnés. Pour économiser sur la longueur de la terminologie, renommons **constellations** de (G, \prec) les forêts clique-connectantes stellaires bien ordonnées de (G, \prec) . Le premier intérêt des constellations est que

Proposition 1.25 *Pour tout graphe $G = (V, E)$ et tout ordre total \prec sur V , les partitions en cliques de G sont en bijection avec les constellations de (G, \prec) . \square*

Le deuxième intérêt des constellations de (G, \prec) est qu'elles définissent un hypergraphe héréditaire conforme. Autrement dit, elles sont les stables d'un graphe auxiliaire, comme nous l'expliquons maintenant. Deux arêtes $e, f \in E(G)$ sont **dépendantes** dans (G, \prec) s'il n'existe pas de constellation qui contient ces deux arêtes à la fois. e et f sont indépendantes sinon. Observons tout d'abord que e et f sont indépendantes si elles sont disjointes. D'autre part, si $e = uv$ et $f = uw$, alors

$$(1.34) \quad (e, f) \text{ sont indépendantes} \iff (u \prec v \text{ et } u \prec w \text{ et } vw \in E(G))$$

Le **graphe des arêtes dépendantes** de $(G = (V, E), \prec)$ est le graphe $D(G, \prec) = (E, E')$ qui a E comme ensemble de sommets et tel que $ef \in E'$ si et seulement si elles sont dépendantes dans (G, \prec) . La Proposition suivante est montrée de manière légèrement différente dans [47].

Proposition 1.26 *Pour tout graphe G et tout ordre total \prec sur $V(G)$, $F \subseteq E(G)$ est une constellation de (G, \prec) si et seulement si F est un stable de $D(G, \prec)$.*

Preuve La nécessité d'être un stable de $D(G, \prec)$ est évidente (car les constellations forment un hypergraphe héréditaire). Soit F un stable de $D(G, \prec)$. Vérifions que F définit une forêt clique-connectante stellaire bien-ordonnée de $D(G, \prec)$ en collectant ces propriétés au fur et à mesure.

F est une Forêt : sinon, F contient un circuit $C \subseteq F$. Prenons un sommet x de C maximum (pour \prec). Les deux arêtes incidentes à x sur C sont dépendantes, contradiction.

F est stellaire : sinon F contient un chemin v_4, v_3, v_2, v_1 sur trois arêtes. Par symétrie, supposons $v_3 \prec v_2$, mais alors les arêtes v_2v_3 et v_2v_1 sont dépendantes, contradiction.

F est bien ordonnée : sinon, il existe $v_1v_2, v_2v_3 \in F$ avec $v_3 \prec v_2$, contradiction.

F est clique-connectante : sinon, F contient deux arêtes uv et uw telles que $u \prec v \prec w$, mais $vw \notin E(G)$, contradiction.

□

Corollaire 1.27 *Pour tout graphe G et tout ordre total \prec sur $V(G)$,*

$$(1.35) \quad \bar{\chi}(G) + \alpha(D(G, \prec)) = |V(G)|$$

Cette formule n'est pas sans rappeler les formules de Gallai sur les liens entre stables, couvertures par des sommets, couplages et couvertures par des arêtes. Mais comme le souligne Schrijver [129], les résultats polyédraux sur les couplages ont un équivalent en termes de couverture par des arêtes et le polytope des stables est équivalent (par symétrie autour du point $\mathbf{1}_V/2$) au polytope des couverture par des sommets. Dans le cas du lien entre les partitions en cliques et les constellations, un équivalent polyédral n'est pas envisageable puisqu'il semble ne pas exister de définition naturelle du polyèdre des colorations.

Pour les forêts cliques connectantes en général et les constellations en particulier, il semble naturel de les plonger dans \mathbb{R}^E . Quelques remarques élémentaires sur les constellations, leur polytope et leur applications sont disponibles dans les Sections 3.4.7 et 4.4.2. ainsi que dans [47].

1.3.3.5 Sous-modularité et systèmes TDAU

De nombreux systèmes associés aux fonctions sous-modulaires sont box-TDI. La technique de preuve typique repose sur le Théorème 1.23 : pour toute fonction objectif, tel que le dual est réalisable, il existe une solution duale optimale dont les variables actives ont une structure particulièrement simple. Cette technique est utilisée dans la Section 4.2.2. Les résultats et méthodes liés à la sous-modularité et discutés ici sont des fondamentaux de la théorie coopérative des jeux (Section 1.4). Le point de départ commun de ces deux théories est l'étude des 2 polyèdres suivants

associés à une fonction d'ensemble f ²⁹ :

$$(1.36) \quad P_f := \{x \in \mathbb{R}^V \mid x \geq \mathbf{0}, x(U) \leq f(U) \text{ pour tout } U \subseteq V\}$$

$$(1.37) \quad EP_f := \{x \in \mathbb{R}^V \mid x(U) \leq f(U) \text{ pour tout } U \subseteq V\}$$

Nous supposons que le lecteur connaît l'algorithme glouton pour trouver un indépendant de poids maximum dans un (poly)matroïde [129]. Nous discutons ici seulement l'intégralité des systèmes (les questions algorithmiques sont traitées dans [129]).

Théorème 1.28 [129] *Pour tout matroïde \mathcal{M} , les inégalités de rang*

$$(1.38) \quad x(U) \leq r(U) \text{ pour tout } U \subseteq V$$

forment un système box-TDI (et décrivent donc l'enveloppe convexe des indépendants de \mathcal{M} lorsque l'on ajoute les inégalités $x(U) \geq 0$).

Idée de preuve Pour un objectif $w \in \mathbb{R}^V$ fixé, considérer $\max\{w^T x\}$ sujet à (1.38). Notons tout d'abord que les variables duales sont en bijection avec les sous-ensembles de V . Choisir une solution duale optimale y . **Décroiser** y consiste à prendre $S, T \subseteq V$ tels que $y_S > 0$ et $y_T > 0$ et tels que ni $S \subseteq T$ ni $T \subseteq S$, puis à diminuer y_S et y_T et à augmenter $y_{S \cap T}$ et $y_{S \cup T}$ d'une même valeur $\alpha := \min\{y_S, y_T\}$. Le décroissement maintient toujours la réalisabilité. Grâce à la sous-modularité, le décroissement préserve aussi l'optimalité. En décroissant y tant que c'est possible, on obtient une solution duale optimale dont les variables non-nulles forment une chaîne de sous-ensembles de V . La matrice d'incidence d'une chaîne étant TTU, le système (1.38) est TDAU. \square

Cela peut être généralisé pour deux (poly-)matroïdes :

Théorème 1.29 (d'intersection des polymatroïdes) [129] *Soit f_1 et f_2 deux fonctions sous-modulaires sur V . Le système suivant est box-TDI :*

$$(1.39)(i) \quad x(U) \leq f_1(U) \text{ pour tout } U \subseteq V$$

$$(ii) \quad x(U) \leq f_2(U) \text{ pour tout } U \subseteq V$$

Idée de preuve Pour un objectif $w \in \mathbb{R}^V$ fixé, choisir une solution duale optimale y . Séparer les composantes de y associées à f_1 et f_2 . Décroiser indépendamment ces deux composantes pour obtenir l'union de deux chaînes. La matrice d'incidence de l'union de deux chaînes étant TTU, le système est TDAU. \square

²⁹ P_f (et EP_f) viennent de (*extended*) *polymatroid* associé à f , quand f est sous-modulaire.

D'autres systèmes box-TDI liés à des fonctions sous-modulaires concernent les cas où l'on ne demande les inégalités que pour certains sous-ensembles de V . En particulier, lorsque f est sous-modulaire mais que $f(\emptyset) < 0$, EP_f est vide. Cependant, considérons le polyèdre EP'_f défini par :

$$(1.40) \quad x(U) \leq f(U) \text{ pour tout } U \in \mathcal{P}(V) \setminus \emptyset$$

Théorème 1.30 [129] *Pour toute fonction sous-modulaire f , le système (1.40) est non-vide et box-TDI.*

Idée de preuve Une solution duale optimale peut être décroisée en une famille laminaire. La matrice d'incidence d'une famille laminaire étant TTU, le système (1.40) est TDAU. \square

Par ailleurs, la fonction

$$\widehat{f}(U) := \min \left\{ \sum_{P \in \mathcal{P}} f(P) \mid \mathcal{P} \text{ est une partition de } U \text{ en ensembles non-vides} \right\}$$

est sous-modulaire et $EP'_f = EP'_{\widehat{f}}$. De plus, \widehat{f} est l'unique fonction maximale parmi l'ensemble des fonctions f' sous-modulaires telles que $f'(\emptyset) = 0$ et $f' \leq f$. La fonction \widehat{f} est la **troncation de Dilworth** de f . Les théorèmes d'intersection des polymatroïdes et la troncation de Dilworth ont une généralisation commune :

Théorème 1.31 [129] *Soit f_1 et f_2 deux fonctions sous-modulaires sur V . Le système suivant est box-TDI :*

$$(1.41) \quad x(U) \leq f_i(U) \text{ pour tout } U \in \mathcal{P}(V) \setminus \emptyset \text{ et } i = 1, 2.$$

Idée de preuve Une solution duale optimale pouvant être décroisée en l'union de deux familles laminaires, le système (1.41) est TDAU. \square

1.4 Jeux coopératifs : coeur, fonctions équilibrées et troncations d'une fonction d'ensemble

Dans cette section, nous présentons la théorie des jeux coopératifs en étudiant le coeur EB_f d'une fonction f

$$(1.42) \quad B_f := \{x \in \mathbb{R}_+^V \mid x(V) = f(V), x(U) \leq f(U) \text{ pour tout } U \subseteq V\}$$

$$(1.43) \quad EB_f := \{x \in \mathbb{R}^V \mid x(V) = f(V), x(U) \leq f(U) \text{ pour tout } U \subseteq V\}$$

Le résultat de base de cette théorie est le Théorème de Bondareva-Shapley qui caractérise l'existence d'une solution dans le cœur d'une fonction f grâce au Lemme de Farkas. Ce théorème est cependant nettement insuffisant pour traiter la question de la formation des coalitions (Section 1.4.1.4) ainsi que de l'existence d'une solution entière dans le cœur de f (Section 1.4.3).

Nous discutons les jeux coopératifs associés à certains problèmes d'optimisation combinatoire (Section 1.4.2) avant d'étudier le cas du voyageur de commerce, pour lequel de nombreuses questions de théorie des jeux restent ouvertes et suscitent des questions algorithmiques originales (Section 1.4.2.4).

Nous étudions finalement l'existence d'une solution entière dans le cœur d'une fonction totalement-équilibrée (Section 1.4.3). Nous observons qu'une solution entière existe dans le cas où f est sous-cardinale (Section 1.4.3.2) ainsi que dans le cas où f est taille-définie (Section 1.4.3.3) mais donnons un contre-exemple pour le cas général (Section 1.4.3.4).

1.4.1 Jeux coopératifs : une affaire de cœur

1.4.1.1 Introduction aux jeux coopératifs

La théorie de jeux coopératifs étudie la manière dont un ensemble V de joueurs est susceptible de se répartir un ensemble de ressources (pour la production desquelles ils ont coopéré) ou des coûts (lorsqu'ils payent pour un service qui leur est rendu en commun) [6]. Pour cela, on suppose que la répartition doit jouir de propriétés morales ou concurrentielles. On essaye alors de traduire ces principes en termes mathématiques.

Exemple 1.32 *Des équipes de recherche opérationnelle de diverses universités françaises s'associent pour inviter un chercheur Canadien pour donner une série de conférences. Celui-ci vient successivement dans les différentes villes lors d'un unique voyage (ce qui a pour but originel de faire une économie sur le coût global de transport mais a aussi pour bénéfice collatéral de lutter activement contre la production de gaz à effet de serre...). Le coût de transport doit alors être partagé entre les différentes équipes. On pourrait diviser le prix de manière uniforme entre toutes les équipes. Cette solution n'est pas retenue en général, car elle n'est ni équitable (au sens moral), ni stable (au sens concurrentiel). Par exemple, si le chercheur vient visiter 4 universités Parisiennes et un centre de recherche à Nice, il reviendrait moins cher aux universités Parisiennes de partager le coût d'un voyage intra-muros que de partager le coût d'un voyage passant par Nice.*

La théorie des jeux coopératifs prend son sens lorsque le fait de coopérer permet à un groupe d'obtenir une meilleure situation que si chacun agissait indépendamment des autres. On suppose en général que certains groupes de joueurs peuvent

former une coalition et jouer indépendamment des autres joueurs, voire explicitement contre les autres joueurs, comme dans l'exemple suivant.

Exemple 1.33 *Un groupe de pirates a trouvé un trésor. Une coalition secrète émerge et s'entend pour tuer ceux qui n'en font pas partie pendant la nuit. Le but étant de se partager le butin en plus petit nombre et donc avec une plus grosse part pour chacun. Le phénomène peut bien sûr survenir récursivement. Parfois, jusqu'à ce qu'il ne reste plus qu'un seul pirate.*

Une grande partie des concepts de la théorie des jeux coopératifs repose toutefois sur l'hypothèse que les coalitions ne peuvent pas jouer directement contre les autres, mais seulement indépendamment des autres [6]. En fait une grande partie des résultats se basent sur un cadre théorique élémentaire que nous présentons maintenant. Un **jeu coopératif** est une fonction d'ensemble $f : \mathcal{P}(V) \rightarrow \mathbb{R}_+$ plus une valeur dans l'ensemble {coût, revenu}. V est l'ensemble des **joueurs** et f est la **fonction caractéristique** du jeu.

On suppose en général que $f(U)$ est le coût que peut payer le sous-groupe U (ou le gain qu'il peut s'octroyer) s'il décide de former une coalition (*i.e.* d'agir indépendamment des autres joueurs). À partir de là, la question de base que se pose la théorie des jeux coopératifs est de savoir comment les coûts (ou les gains) vont être répartis. Une question moins courante, mais qui tient le devant de la scène dans le Chapitre 4, est de savoir quelles coalitions vont se former. Pour simplifier, l'enseignement de base de la théorie des jeux coopératifs est que ces deux types de questions sont duaux l'un de l'autre (au sens de la dualité en programmation linéaire). Nous discutons ces deux types de questions ainsi que le phénomène de dualité qui les relie dans les Sections 1.4.1.3 et 1.4.1.4.

Nous discutons la théorie pour les jeux de coûts seulement. Le Théorème de Bondareva-Shapley (ci-dessous) se transpose directement aux jeux de revenus [6], tandis que l'étude des jeux combinatoires et les questions d'intégralité nécessiteraient une inspection spécifique.

1.4.1.2 Petite philosophie à l'usage des (non)-spécialistes : programmation linéaire (en nombres entiers) et troncations des fonctions d'ensembles

Nous présentons ici un angle de vue alternatif concernant la Section 1.4 : l'existence de solutions optimales (entières) pour l'optimisation de $\mathbf{1}^T x$ sur P_f se traduit en termes de "troncations" de la fonction f .

Plusieurs questions liées au cœur d'un jeu coopératif reviennent à résoudre des programmes linéaires (éventuellement en nombres entiers) associés à une fonction d'ensemble f . À chacun de ces programmes correspond naturellement une tronca-

tion de f (*i.e.* une autre fonction d'ensemble associée à f) dont les valeurs sont en relation avec les solutions optimales des programmes linéaires considérés. Dans le cas où f est sous-modulaire, les valeurs optimales des différents programmes linéaires coïncident et les troncations fusionnent en une unique fonction : la troncation de Dilworth de f . En général, les troncations diffèrent, mais certaines peuvent quand même être égales sans que f soit sous-modulaire. L'égalité des troncations se traduit par des propriétés d'intégralité des programmes linéaires considérés, par l'existence de "solutions stables" en théorie des jeux ainsi parfois que par des cas polynomiaux pour des problèmes NP-dur. De même que ce paradigme aurait pu servir de point de départ à la théorie des graphes parfaits (relativement à la coloration minimum), il suscite aussi des pistes de cas intéressants pour le TSP.

L'idée de définir des troncations associées à une fonction f provient de la théorie de la troncation de Dilworth d'une fonction sous-modulaire (Théorème 1.30). En général, l'ensemble des fonctions sous-modulaires (à valeur 0 sur \emptyset) plus petites qu'une fonction f donnée n'admet pas d'élément maximum :

Exemple 1.34 Soit $V = \{1, 2, 3\}$ et f définie par :

$$(1.44) \quad f(S) := \begin{cases} 0 & \text{si } S = \emptyset \\ 2 & \text{si } |S| = 1 \text{ ou } 2 \\ 3 & \text{si } S = V \end{cases}$$

considérons la fonction f' , telle que $f'(S) = 2$ pour tout $S \neq \emptyset$ et f'' telle que $f''(S) = |S|$ pour tout $S \subseteq V$. f' et f'' sont sous-modulaires, inférieures à f et à valeur 0 sur \emptyset mais il n'existe pas de telle fonction qui soit supérieure à la fois à f' et à f'' □

La notion de troncation sous-modulaire n'a donc pas de sens pour une fonction arbitraire.

1.4.1.3 Théorie classique, cœur d'un jeu et fonctions équilibrées

Dans cette section, nous définissons le cœur d'un jeu coopératif et énonçons le théorème de Bondareva-Shapley, qui caractérise la non-vacuité du cœur en terme d'équilibre de la fonction caractéristique du jeu. Les problématiques et définitions de cette section deviennent beaucoup plus claires si l'on fait l'hypothèse que la fonction f est non-décroissante et que $f(\emptyset) = 0$.

L'hypothèse implicite que fait habituellement la théorie des jeux coopératifs est qu'un décideur (qui peut être une personne extérieure, ou l'assemblée des joueurs) doit décider comment répartir les coûts entre les joueurs de manière équitable. L'autre hypothèse, explicite celle-là, est que l'on cherche une *imputation* c'est-à-dire, un vecteur $x \in \mathbb{R}^V$ tel que $x(V) = f(V)$. Plusieurs paradigmes peuvent conduire à définir l'équité d'une imputation. Nous présentons ici seulement le pa-

radigme du cœur et référons à [6] pour une discussion des autres paradigmes classiques (valeur de Shapley, noyau...).

La définition du cœur, vient du constat suivant. Si on demande à un sous-ensemble U de joueurs de payer, ensemble, un coût supérieur à $f(U)$ (i.e. si $x(U) > f(U)$), ils pourraient former une coalition et tous en tirer un bénéfice, par exemple en redistribuant les coûts de la manière suivante : $x'_v = x_v - (x(U) - f(U))/|U|$.

Le cœur d'un jeu coopératif est l'ensemble des répartitions des imputations qui évitent qu'une telle coalition ait intérêt à se former : le **cœur** de f est l'ensemble EB_f des $x \in \mathbb{R}^V$ tels que

$$(1.45)(i) \quad x(U) \leq f(U) \quad \text{pour tout } \emptyset \neq U \subseteq V$$

$$(ii) \quad x(V) = f(V)$$

Une motivation alternative pour la définition du cœur est de nature éthique : bien que les joueurs ne soient pas toujours aptes à former une coalition et à contester la répartition des coûts, il n'y a pas de raison qu'un ensemble de joueurs paye pour les autres joueurs. Un ensemble de clients ne devrait donc pas payer plus que ce qu'il coûterait pour les servir eux seulement (sans servir les autres clients). Cette interprétation s'applique par exemple aux clients d'une entreprise en situation de monopole.

Le polyèdre $B_f = EB_f \cap \mathbb{R}_+^V$ est parfois aussi appelé **cœur**³⁰. Une **imputation stable** est un élément du cœur. En général, un jeu coopératif peut avoir un cœur vide, ce qui amène à des situations problématiques car instables.

Exemple 1.35 *Trois amis (Adrien, Bernard et Clément) en vacances sur une île Grecque décident de louer des scooters pour se déplacer. Ils veulent minimiser les coûts, mais sont obligés de louer au moins deux scooters (car ceux-ci n'ont que deux places). Soit f la fonction d'ensemble qui compte le nombre de scooters (et donc le prix) que doit payer un sous-ensemble d'amis :*

- $|V| = 3$
- $f(\emptyset) = 0$
- $f(U) = 1$ si $|U| = 1$ ou 2 .
- $f(V) = 2$

Si A et B s'entendent pour louer un scooter à deux, ils peuvent jouer indépendamment de C. On ne peut donc pas forcer A et B à payer un coût supérieur à 1. Supposons que A et B décident de partager en $x_A = t$ et $x_B = 1 - t$. C doit donc payer 1 tout seul. C peut alors proposer à A de changer de coalition pour s'associer avec lui. Ensemble, ils ne devront payer que 1 au lieu de $1 + t$. Ils peuvent donc

³⁰Les notations B_f et EB_f ne sont pas classiques en théorie des jeux. Elles viennent de (*extended*) *base polyhedron* du polymatroïde associé à une fonction f quand f est sous-modulaire.

tous deux diminuer le prix qu'il payaient lors de la première imputation. Mais B se retrouve avec 1 à payer et peut à son tour proposer à l'un des deux autres de s'associer avec lui et d'être gagnant par rapport à la situation présente...

Si le cœur d'un jeu coopératif est vide en général, une caractérisation de la non-vacuité est suscitée par l'exemple suivant :

Exemple 1.36 *Le loueur de scooter de l'Exemple 1.35 décide d'offrir un rabais pour la location de deux scooters, car il sent que les trois touristes sont sur le point de partir visiter l'île en bus. Le jeu s'exprime alors par une nouvelle fonction g :*

- $|V| = 3$
- $g(\emptyset) = 0$
- $g(U) = 1$ si $|U| = 1$ ou 2.
- $g(V) = \alpha$

Cette fois, (1.45) a une solution si et seulement si $\alpha \leq 3/2$. La suffisance de cette condition peut être vue en vérifiant que $\mathbf{1}_V/2$ est une solution réalisable. La nécessité vient du fait que $g(V) \leq 3/2$ est conséquence des autres inégalités (en sommant les inégalités sur les U ayant $|U| = 2$ puis en divisant par deux).

L'Exemple 1.36 reflète la situation générale concernant la vacuité du cœur comme nous le détaillons maintenant.

Une **partition fractionnaire** d'un ensemble V est une fonction $\lambda : \mathcal{P}(V) \setminus \emptyset \rightarrow [0, 1]$ telle que

$$(1.46) \quad \sum_{S \subseteq V} \lambda_S \mathbf{1}_S = \mathbf{1}_V$$

Une **couverture fractionnaire** d'un ensemble V est une fonction $\lambda : \mathcal{P}(V) \setminus \emptyset \rightarrow \mathbb{R}_+$ telle que

$$(1.47) \quad \sum_{S \subseteq V} \lambda_S \mathbf{1}_S \geq \mathbf{1}_V$$

Une fonction d'ensemble f sur V est **équilibrée** si

$$(1.48) \quad \sum_{S \subseteq V} \lambda_S f(S) \geq f(V)$$

pour toute couverture fractionnaire λ de V .

Le lemme de Farkas donne une condition nécessaire et suffisante pour que B_f et EB_f soient non-vides :

Théorème 1.37 [25] *Le polytope B_f associé à un jeu $f : \mathcal{P}(V) \rightarrow \mathbb{R}_+$ est non-vide si et seulement si la fonction f est équilibrée.*

Théorème 1.38 (Bondareva) ³¹[25] *Le cœur EB_f d'un jeu $f : \mathcal{P}(V) \rightarrow \mathbb{R}$ est non-vide si et seulement si*

$$\sum_{S \subseteq V} \lambda_S f(S) \geq f(V)$$

pour toute partition fractionnaire λ de V .

Dans le cas (habituel) où f est non-décroissante et à valeur dans \mathbb{R}_+ , B_f est non-vide si et seulement si EB_f est non-vide. De plus, considérer des partitions fractionnaires est alors équivalent à considérer des couvertures fractionnaires (en particulier pour définir l'équilibre de f).

Le **sous-jeu** induit par $U \subseteq V$ est la restriction de f à $\mathcal{P}(U)$. Une fonction est **totale-équilibrée** si la fonction induite par tout sous-jeu est équilibrée. Une fonction totale-équilibrée est donc non-décroissante. D'après le Théorème 1.37, $B_f \neq \emptyset$ pour tout sous-jeu de f si et seulement si f est totale-équilibrée.

Si le Théorème de Bondareva-Shapley généralise l'Exemple 1.36, une autre généralisation consiste à demander quel est le maximum que l'on peut faire payer à un ensemble de joueurs sans prendre le risque de la formation de coalitions.

Pour une fonction d'ensemble $f : \mathcal{P}(V) \rightarrow \mathbb{R}_+$, soit \bar{f} la fonction définie pour tout $U \subseteq V$ par

$$(1.49) \quad \bar{f}(U) := \min \left\{ \sum_{T \subseteq V} \lambda_T f(T) \mid \lambda \text{ est une couverture fractionnaire de } U \right\}$$

En particulier, nous autorisons à utiliser des valeurs $f(T)$ pour des ensembles $T \not\subseteq U$ pour construire $\bar{f}(U)$. Si $f(\emptyset) \geq 0$ et f est non-décroissante, cela n'est pas nécessaire et on a :

$$(1.50) \quad \bar{f}(U) = \min \left\{ \sum_{T \subseteq U} \lambda_T f(T) \mid \lambda \text{ est une partition fractionnaire de } U \right\}$$

Par définition, $\bar{f}(\emptyset) = 0$. Clairement, $\bar{f} \leq f$ et \bar{f} est totale-équilibrée. En fait \bar{f} est l'unique fonction maximale avec ces propriétés :

³¹Le "Théorème de Bondareva-Shapley" est équivalent mais n'est habituellement pas formulé en ces termes. Shapley [130] appelle "famille équilibrée" (*balanced set*) le support d'une partition fractionnaire. L'ensemble des partitions fractionnaires formant un polytope, on peut reformuler les conditions du Théorème 1.38 en termes des sommets de ce polytope, c'est-à-dire avec un nombre fini de conditions. Shapley observe que pour les sommets de ce polytope, les coefficients λ sont définis de manière unique par l'ensemble des variables actives (cette observation est à l'origine du mot "famille équilibrée"). De plus, Shapley [130] décrit tous les sommets du polytope des partitions fractionnaires pour $|V| = 4$. Nous utilisons implicitement les familles équilibrées dans la Section 4.2. Si les familles équilibrées sont souvent utilisées pour énoncer le Théorème 1.38, cela ne semble digne d'intérêt que dans la mesure où l'on est capable de décrire les sommets de leur polytope, ce qui semble ouvert pour $|V| \geq 5$.

Proposition 1.39 *Pour toute fonction $f : \mathcal{P}(V) \rightarrow \mathbb{R}_+$, et toute fonction f' totalement-équilibrée telle que $f' \leq f$, on a $f' \leq \bar{f}$.*

Preuve Soit $f : V \rightarrow \mathbb{R}_+$ et f' totalement-équilibrée telle que $f' \leq f$. Pour tout $U \subseteq V$, il existe une couverture fractionnaire λ de U telle que $\bar{f}(U) = \sum \lambda_T f(T)$. Comme f' est totalement-équilibrée et $f' \leq f$, on a aussi $f'(U) \leq \sum \lambda_T f(T)$, et donc $f' \leq \bar{f}$. \square

En vertu de la Proposition 1.39, appelons *troncation totalement-équilibrée* de f , la fonction \bar{f} .

1.4.1.4 Théorie duale : troncation sous-additive d'une fonction d'ensemble

Par le principe de dualité, le Théorème de Bondareva-Shapley répond partiellement à une autre question que la non-vacuité du cœur : celle de la formation des coalitions. Plus précisément, un jeu a un cœur non-vidé si et seulement si la coalition de tous les joueurs est optimale parmi l'ensemble des partitions fractionnaires des joueurs en coalitions. Autrement dit, un jeu est équilibré si et seulement si aucune coalition n'a intérêt à se former. Mais si le jeu n'est pas équilibré le Théorème de Bondareva-Shapley ne donne qu'une information partielle sur la formation des coalitions car, en pratique, une coalition se forme ou ne se forme pas. Autrement dit, c'est d'un problème de partitions entières et non de partitions fractionnaires dont il est question. Or, à notre connaissance, il n'existe pas de concepts théoriques simples pour traiter de la manière dont les coalitions devraient se former dans un contexte où chaque joueur cherche à minimiser son coût. Le concept que nous étudions ici n'est pas motivé à la base par un contexte de théorie de jeux, mais plutôt par un contexte de recherche opérationnelle (voir Chapitre 4). En théorie des jeux, il se traduit par la problématique suivante :

“Comment un coordinateur extérieur doit-il s'y prendre pour former l'ensemble des coalitions de manière à minimiser le coût total ?” (c'est-à-dire la somme des coûts des coalitions).

Pour une fonction d'ensemble f soit \hat{f} la fonction définie pour tout $U \subseteq V$ par

$$(1.51) \quad \hat{f}(U) := \min \left\{ \sum_{P \in \mathcal{P}} f(P) \mid \mathcal{P} \text{ est une partition de } U \text{ en ensembles non-vides} \right\}$$

Notons que si f est entière, alors \hat{f} l'est aussi. De même si f est non-décroissante ou si f est non-négative, alors \hat{f} l'est aussi. Par ailleurs, indépendamment de f , on a que $\hat{f}(\emptyset) = 0$, que $\hat{f} \leq f$ et que \hat{f} est sous-additive. En fait \hat{f} est l'unique fonction maximale avec ces propriétés :

Proposition 1.40 *Pour toute fonction $f : \mathcal{P}(V) \rightarrow \mathbb{R}$, et toute fonction f' telle que $f' \leq f$, $f'(\emptyset) = 0$ et f' est sous-additive, on a $f' \leq \hat{f}$.*

Preuve Soit $f : V \rightarrow \mathbb{R}$ et f' sous-additive telle que $f' \leq f$ et $f'(\emptyset) = 0$. Pour tout $U \subseteq V$, il existe une partition $\{C_1, C_2, \dots, C_k\}$ de U telle que $\widehat{f}(U) = f(C_1) + f(C_2) + \dots + f(C_k)$. Comme f' est sous-additive, et $f' \leq f$, on a aussi $f'(U) \leq f(C_1) + f(C_2) + \dots + f(C_k)$ et donc $f' \leq \widehat{f}$. \square

En vertu de la Proposition 1.40, appelons *troncation sous-additive*³² de f , la fonction \widehat{f} .

Si la théorie classique des jeux coopératifs se concentre sur la recherche d'une imputation, il est raisonnable d'étudier d'autres types de vecteurs de partage des coûts $x : V \rightarrow \mathbb{R}_+$, particulièrement dans le cas où f n'est pas sous-additive. Il est alors intéressant d'étudier quand le cœur de \widehat{f} est non-vide (c'est ce que nous faisons dans la Section 4.2). Il est donc intéressant d'étudier les fonctions f dont la troncation totalement-équilibrée et la troncation sous-additive sont égales (*i.e.* $\widehat{f} = \overline{f}$). L'égalité de ces deux troncations est aussi une propriété importante en combinatoire polyédrale comme nous le discutons dans la Section 1.4.2.3.

1.4.1.5 Sous-modularité, systèmes TDI et équilibre total

Dans cette section, nous observons que, parmi les fonctions non-négatives et non-décroissantes, les fonctions sous-modulaires forment une sous-classe des fonctions totalement-équilibrées. De plus, si on se restreint aux fonctions sous-additives, alors la classes des fonctions dont le cœur définit une système TDI est une classe intermédiaire.

La sous-modularité³³ est une propriété fondamentale en théorie des jeux [6] car elle implique des propriétés bien plus fortes que la non-vacuité du cœur. En particulier, on peut calculer une imputation de manière gloutonne si f est sous-modulaire :

Proposition 1.41 (Shapley) *Soit f une fonction d'ensemble sous-modulaire. Si $f(\emptyset) = 0$, alors $EB_f \neq \emptyset$. Si de plus f est non-décroissante, alors $B_f \neq \emptyset$ et f est totalement-équilibrée.*

Preuve Choisissons un ordre total arbitraire $\{1, 2, \dots, n\}$ sur V . Choisissons $x : V \rightarrow \mathbb{R}$ de manière gloutonne : $x_i := f(\{1, 2, \dots, i\}) - f(\{1, 2, \dots, i-1\})$. Par construction, $x(V) = f(V)$ et par sous-modularité, $x \in EB_f$. Si f est non-décroissante, $x \geq \mathbf{0}$, donc $x \in B_f$. En appliquant l'algorithme glouton pour $U \subseteq V$, on obtient une solution dans le cœur du sous-jeu associé à U . Donc f est totalement-équilibrée. \square

³²Le choix du nom aurait pu se porter sur *troncation de Dilworth* (car (1.51) est la définition donnée à la troncation de Dilworth [129] Chapitre 48). Cependant, cette terminologie aurait été ambiguë car, dans le cas où f est sous-modulaire, toutes les troncations que nous définissons dans ce chapitre coïncident. Cette égalité des troncations est un des intérêts majeurs de la troncation de Dilworth.

³³Aussi appelée concavité ou convexité en théorie des jeux.

Soit $|V| = 2$. La fonction

$$(1.52) \quad f(U) := \begin{cases} 0 & \text{si } |U| = 0 \\ 1 & \text{si } |U| = 1 \\ 3 & \text{si } |U| = 2 \end{cases}$$

montre qu'il n'est pas nécessaire que f soit totalement-équilibrée pour que

$$(1.53) \quad x(U) \leq f(U) \text{ pour tout } U \subseteq V$$

soit un système TDI. L'implication est toutefois valide si l'on suppose f sous-additive. Autrement dit,

Proposition 1.42 *Soit f une fonction d'ensemble non-négative et non-décroissante. Si (1.53) est un système TDI alors $\widehat{f} = \overline{f}$. En particulier, si f est sous-additive et si (1.53) est un système TDI alors f est totalement-équilibrée.*

Preuve L'équilibre total signifie que $\max\{\max c^T x \mid x \text{ est sujet à (1.53)}\}$ a une solution duale entière pour tout $c \in \{0, 1\}^V$. Par définition, cela est impliqué par la propriété TDI. \square

De la Proposition 1.42 on déduit que de nombreux jeux coopératifs associés à des problèmes d'optimisation combinatoire, dont les inégalités de rang décrivent un système TDI sont totalement-équilibrés. Il est plus surprenant que la réciproque soit parfois vraie et que l'équilibre total caractérise en fait ces structures combinatoires. C'est le cas notamment des graphes parfaits, comme nous l'observons dans la Section 1.4.2.

1.4.2 Jeu coopératif associé à un problème d'optimisation combinatoire

Si la donnée d'un jeu coopératif consiste en une liste de 2^V nombres dans le cas général, de nombreux jeux admettent un codage compact. C'est le cas des jeux définis par des quantités caractéristiques de structures combinatoires (comme le nombre chromatique d'un sous-graphe induit). Si la sous-modularité d'un jeu semble toujours traduire des cas polynomiaux, l'importance de l'équilibre total est moins claire. Nous reviendrons sur ce point dans la Section 1.4.3. En général, on associe un jeu de coût à un problème de minimisation et un jeu de revenu à un problème de maximisation. Nous dérogeons à cette règle dans les Sections 1.4.2.1 et 1.4.2.2 pour donner une caractérisation des hypergraphes héréditaires. Nous traitons ensuite du jeu de la coloration minimum et du TSP.

1.4.2.1 Rang d'un hypergraphe

Soit $\mathcal{H} = (V, \mathcal{E})$ un hypergraphe et soit r la fonction de rang définie par

$$(1.54) \quad r(U) = \max\{|E| \mid E \in \mathcal{E}, E \subseteq U\}$$

Nous avons vu (Proposition 1.2) que le rang d'un hypergraphe est toujours non-décroissant, entier, sous-cardinal que si $f(U) > f(U - v)$ pour tout $v \in U$ alors $f(U) = |U|$. Nous observons maintenant que dans le cas des fonctions définies par le rang d'un hypergraphe, la sous-additivité et l'équilibre total sont équivalentes et caractérisent l'hérédité :

Proposition 1.43 *Soit $f : V \rightarrow \mathbb{N}$ sous-cardinale et non-décroissante. Les conditions suivantes sont équivalentes :*

- i) f est le rang d'un hypergraphe héréditaire*
- ii) f est totalement-équilibrée*
- iii) f est sous-additive et si $f(U) > f(U - v)$ pour tout $v \in U$ alors $f(U) = |U|$*

Preuve *i) \implies ii)* Supposons que f soit le rang d'un hypergraphe héréditaire $H = (V, \mathcal{E})$. Soit $U \subseteq V$ et $K \subseteq U$ tel que $|K| = f(U)$ et $\lambda : \mathcal{P}(V) \rightarrow [0, 1]$, une partition fractionnaire de U . On a :

$$(1.55) \quad \sum_{S \subseteq U} \lambda_S f(S) \geq \sum_{S \subseteq U} \lambda_S f(S \cap K) = \sum_{S \subseteq U} \lambda_S |S \cap K| = |K| = f(U)$$

Et donc f est totalement-équilibrée.

ii) \implies iii) L'équilibre total implique toujours la sous-additivité. Supposons qu'il existe U tel que $f(U) > f(U - v)$ pour tout $v \in U$ mais $f(U) \leq |U| - 1$. Considérons le recouvrement fractionnaire λ de U tel que $\lambda_{U-v} = 1/(|U| - 1)$ pour tout v et $\lambda_T = 0$ sinon. On a

$$(1.56) \quad \sum \lambda_S f(S) = \sum_{v \in U} \frac{1}{|U| - 1} f(U - v) \leq \frac{|U| (f(U) - 1)}{|U| - 1} < f(U)$$

et f n'est pas totalement-équilibrée.

iii) \implies i) Supposons que $f(U) > f(U - v)$ pour tout $v \in U$ implique $f(U) = |U|$. D'après la Proposition 1.2, f est le rang d'un hypergraphe $H = (V, \mathcal{E})$. Supposons que H ne soit pas héréditaire. Il existe donc $K \in \mathcal{F}$ et $S \subsetneq K, S \notin \mathcal{F}$. Puisque $S \notin \mathcal{F}$, on a $f(S) < |S|$. On a donc

$$(1.57) \quad f(S) + f(K \setminus S) < |S| + |K \setminus S| = f(K)$$

et f n'est donc pas sous-additive. □

1.4.2.2 Nombre de stabilité

Soit $G = (V, E)$ un graphe et α la fonction d'ensemble sur V définie par $\alpha(U) := \alpha(G[U])$. Comme les stables de G définissent une famille héréditaire, α est une fonction totalement-équilibrée pour tout graphe (cf. Proposition 1.43). Par ailleurs :

Proposition 1.44 *α est une fonction sous-modulaire si et seulement si G est un graphe sans- P_3 (P_3 est le graphe qui a 3 sommets et deux arêtes).*

Preuve Les stables de P_3 ne définissent pas les indépendants d'un matroïde car les stables maximaux n'ont pas tous la même cardinalité. Réciproquement, un graphe sans- P_3 est une union disjointe de cliques, dont l'ensemble des stables forme les indépendants d'un matroïde de partition (voir aussi Chapitre 2). \square

1.4.2.3 Nombre chromatique

Soit $G = (V, E)$ un graphe et χ la fonction d'ensemble sur V définie par $\chi(U) := \chi(G[U])$. Le cœur de la fonction χ est non-vidé si et seulement si le nombre chromatique de G est égal à son nombre chromatique fractionnaire.

Proposition 1.45 *Soit $G = (V, E)$ un graphe, alors χ est équilibrée si et seulement si $\chi(G) = \chi^*(G)$:*

Preuve Soit $\lambda : \mathcal{P}(V) \rightarrow [0, 1]$ une partition fractionnaire de V qui minimise

$$(1.58) \quad \sum_{U \subseteq V} \lambda_U \chi(U)$$

Montrons qu'il existe une telle partition fractionnaire avec $\lambda_U = 0$ si U n'est pas un stable. En effet, il suffit, pour chaque U tel que $\lambda_U > 0$ de considérer une $\chi(U)$ -coloration $A_1, A_2, \dots, A_{\chi(U)}$ de U , d'ajouter λ_U à chacun des λ_{A_i} et mettre λ_U à 0. L'optimum de (1.58) n'est donc autre que χ^* . \square

La Proposition est énoncée de manière alternative dans [66] et dans un cadre plus général. Notons une curiosité au passage : la reconnaissance de la non-vacuité du cœur de χ ne semble même pas dans NP³⁴. En effet, même si l'on dispose d'une coloration \mathcal{C} de G et d'une clique fractionnaire x de G de même valeur, il semble NP-difficile de certifier que x est effectivement une clique fractionnaire de G ³⁵. En fait, nous conjecturons que la reconnaissance des graphes pour lesquels $\chi(G) = \chi^*(G)$ est \sum_P^2 -complète.

Cependant, l'équilibre total caractérise une classe de graphe de la plus haute importance. Comme un graphe G satisfait $\chi^*(G[U]) = \chi(G[U])$ pour tout $U \subseteq V(G)$

³⁴Bien que Deng, Ibaraki et Nagamochi [66] "prouvent" que ce problème est NP-complet (leur preuve montrant seulement que le problème est NP-dur).

³⁵Le problème de séparation pour le polytope des cliques fractionnaires étant NP-dur.

si et seulement si G est parfait ([129], Théorème 65.10), on a le résultats suivant, observé aussi dans [66, 67].

Corollaire 1.46 χ est une fonction totalement-équilibrée si et seulement si G est un graphe parfait. \square

On montre, de manière similaire à la Proposition 1.44, que χ est sous-modulaire si et seulement si G est sans- P_3 .

De même que la pertinence de la dualité anti-bloquante implique de se restreindre aux graphes ([129], Chapitre 82), l'équilibre total du nombre chromatique d'un hypergraphe implique que celui-ci est en fait un graphe. Pour $U \subseteq V$, notons $\chi(H[U])$ le nombre chromatique de l'hypergraphe induit par U .

Corollaire 1.47 Pour un hypergraphe (sans boucles) H , la fonction χ est totalement-équilibrée si et seulement si H^{\min} est un graphe parfait.

Preuve Notons que colorier un hypergraphe H est équivalent à colorier l'hypergraphe H^{\min} . L'équivalence ayant été montrée ci-dessus dans le cas des graphes, montrons la nécessité que H^{\min} soit un graphe. Si ce n'est pas le cas, H^{\min} a une arête e telle que $|e| \geq 3$. On a $\chi(H[e]) = 2$ mais $\chi^*(H[e]) = |e|/(|e| - 1) < 2$ qui est atteint par la partition fractionnaire de e ayant les $|e|$ sous-ensembles de e de taille $e - 1$ pour support. \square

1.4.2.4 Le jeu associé au voyageur de commerce

Dans un graphe $G = (V, E)$, un *tour de Steiner* sur $U \subseteq V$ est une marche fermée couvrant les sommets de U [50]. On peut aussi voir un tour de Steiner sur U comme une fonction $x : E \rightarrow \mathbb{N}$ qui satisfait

$$\begin{aligned} (1.59)(i) \quad & x_e \geq 0 && \text{pour tout } e \in E \\ & (ii) \quad x(\delta(T)) \geq 2 && \text{pour tout } T \subseteq U \\ & (iii) \quad x(\delta(T)) \equiv 0 \pmod{2} && \text{pour tout } T \subseteq U \end{aligned}$$

Un cycle Hamiltonien est donc un tour de Steiner sur V , mais la réciproque n'est pas vraie. Si $l : E \rightarrow \mathbb{R}_+$ est une longueur associée à chaque arête de G , le problème de trouver un plus court tour de Steiner est aussi appelé *relaxation graphique* du TSP [50]. On peut ramener la relaxation graphique du TSP au TSP lui-même en considérant le graphe complet sur V muni de la métrique d définie pour tout $x, y \in V$ par

$$d(x, y) = \text{longueur d'un plus court chemin dans } (G, d) \text{ de } x \text{ à } y$$

Le problème que nous considérons ici a des applications dans la modélisation de tournées de véhicules qui doivent partir et revenir d'un unique dépôt. Le dépôt est un sommet de G , noté r . Pour $l : E \rightarrow \mathbb{R}_+$ et $r \in V$, considérons la fonction d'ensemble sur $V - r$ définie par

$$(1.60) \text{GTSP}((G, l, r), U) := \text{longueur min d'un tour de Steiner sur } U + r$$

Notons que, pour tout graphe pondéré enraciné (G, l, r) , la fonction GTSP est non-décroissante et à valeur 0 sur l'ensemble vide. Notons aussi que $\text{GTSP}((G, l, r), U)$ est donné par le plus court circuit Hamiltonien dans le graphe complet sur $(U + r)$ muni de la métrique d . On définit donc GTSP pour un espace métrique enraciné $((V, d), r)$ de manière similaire (sans avoir besoin de graphe). Le jeu du TSP associé à un espace métrique enraciné est simplement défini par la fonction GTSP . D'après le Théorème de Bondareva-Shapley, le cœur de ce jeu est non-vide si et seulement s'il existe un tour de Steiner couvrant $V + r$, optimal parmi l'ensemble des couvertures fractionnaires de V par des tours de Steiner enracinés sur r . Le graphe de Peterson (enraciné sur un de ses sommets) et les graphes de la Figure 1.2 sont des exemples de graphes enracinés pour lesquels GTSP n'est pas équilibrée dans le cas où $l \equiv 1$. Dans ce cas, la métrique d est simplement donnée par la longueur d'un plus court chemin (en nombre d'arêtes) entre deux points. Un graphe pour lequel la fonction GTSP est totalement-équilibrée pour la longueur $l = 1$ et indépendamment du choix de la racine n'a donc aucun de ces graphes comme sous-graphe isométrique³⁶.

Dans le cas pondéré, les propriétés de la fonction d'ensemble GTSP ne dépendent pas seulement du graphe G , mais aussi des longueurs l . Toutefois, la métrique d hérite de certaines propriétés de G . Une des conséquences en est que dans certaines classes de graphes, trouver un plus court tour de Steiner est polynomial quelles que soient les longueurs l [50, 77]. Notons en particulier que pour ce qui concerne les questions relatives aux tours de Steiner, affecter la longueur 0 à une arête e revient à contracter e et qu'affecter la longueur $+\infty$ revient à effacer e . Autrement dit, les propriétés de graphes valides indépendamment de la longueur l sont en général préservées par mineurs. Il n'est donc pas surprenant qu'il existe des caractérisations intéressantes de certaines classes de graphes à partir des propriétés de la fonction GTSP , dès que l'on s'intéresse à des propriétés valides pour toutes longueurs l . Les graphes pour lesquels la fonction GTSP est sous-modulaire pour toute affectation d et toute racine r sont appelés *naturellement sous-modulaires* [94]. Nous préférons le terme, moins ambigu de *GTSP sous-modulaires*. Plusieurs caractérisations de ces graphes sont données dans [94]. En particulier, ce sont les graphes (connexes) dont les composantes 2-sommets connexes sont soit des K_4 soit des graphes outer-planars. Herrer et Penn [94] ne semblent pas avoir remarqué que cette classe est fermée par mineurs. De leur travaux, on déduit cependant que $K_{2,3}$

³⁶Pour calculer les valeurs de GTSP , il faut utiliser la métrique du graphe de départ (et non la métrique d'un sous-graphe induit, qui peut ne pas être isométrique).

est le seul mineur exclu à la classe des graphes GTSP sous-modulaires. Notons que les arbres sont donc une sous-classe des graphes GTSP sous-modulaires. Okamoto [121] propose un survey des questions liées à l'équilibre de la fonction *GTSP*. Inspiré par le théorème de Herrer et Penn [94], Okamoto [121] pose la question de la caractérisation des graphes ***GTSP (totalement-)équilibrés***³⁷, définis par le fait que la fonction *GTSP* soit (totalement-)équilibrée indépendamment de la racine r et des longueurs l .

Cette classe de graphe est liée aux graphes ***TSP-parfaits***³⁸ définis par Fonlupt et Naddef [77] comme les graphes pour lesquels l'enveloppe convexe des tours de Steiner sur V est définie par la relaxation suivante de (1.59) :

$$(1.61)(i) \quad x_e \geq 0 \quad \text{pour tout } e \in E$$

$$(ii) \quad x(\delta(U)) \geq 2 \quad \text{pour tout } U \subseteq V$$

Nous observons quelques liens étroits entre les graphes TSP-parfaits et les GTSP (totalement-)équilibrés.

Proposition 1.48 *Tout graphe TSP-parfait est GTSP totalement-équilibré.*

Preuve Montrons que les inégalités qui définissent le cœur de la fonction GTSP sont plus forte que les inégalités du système (1.61). Soit $G = (V, E)$ et $r, s \in V$. Pour un graphe $G = (V, E)$ et $r \in V$, soit \mathcal{T} l'ensemble des marches fermées contenant le sommet r . Soit $\Lambda : \mathcal{T} \rightarrow \mathbb{R}_+$ une couverture fractionnaire de V par des tours enracinés sur r . Pour $T \in \mathcal{T}$ et $e \in E$, soit $T(e)$ le nombre de fois que T traverse e . Considérons le vecteur $x : E \rightarrow \mathbb{R}_+$ défini par

$$x_e := \sum_{T \in \mathcal{T}} \lambda_T T(e)$$

Considérons un sommet $s \in V$ avec $s \neq r$. Pour tout $T \in \mathcal{T}$ couvrant r et s et toute (r, s) -coupe $U \subseteq V$, on a $x(\delta(U)) \geq 2$. D'autre part, comme \mathcal{T} est une couverture fractionnaire de V , la somme des $\lambda \in \Lambda$ associés à des tours qui contiennent s est supérieure ou égale à 1. Donc toute (r, s) -coupe U satisfait les contraintes (1.61)(ii). Cela termine la preuve car toute coupe de G est une (r, s) -coupe pour un certain s . \square

Corollaire 1.49 *Les graphes M_1 et M_2 de la Figure 1.2 font partie de la liste des mineurs exclus minimaux pour la classe des graphes GTSP (totalement-)équilibrés.*

Preuve Les graphes M_1 et M_2 ne sont pas GTSP équilibrés, comme expliqué dans la Figure 1.2. Puisque M_1 et M_2 sont des mineurs minimaux exclus pour la TSP

³⁷Qu'il appelle "naturellement (totalement-)équilibrés".

³⁸Qu'il serait plus juste d'appeler GTSP-parfaits.

perfection [77], la Proposition 1.48 implique qu'ils le sont aussi pour la classe des graphes GTSP (totalement-)équilibrés. \square

Les graphes GTSP totalement-équilibrés ne sont pas toujours TSP-parfaits comme le montre le troisième et dernier mineur exclus $M_3 = \overline{C_6}$ à la TSP-perfection. En effet, K_6 est GTSP totalement-équilibré [110] et a M_3 comme mineur. Au vu de ces résultats :

Conjecture 1.50 Soit G un graphe. Les conditions suivantes sont équivalentes :

- G est GTSP totalement-équilibré,
- G est GTSP équilibré,
- G est sans-mineur $\{M_1, M_2\}$.

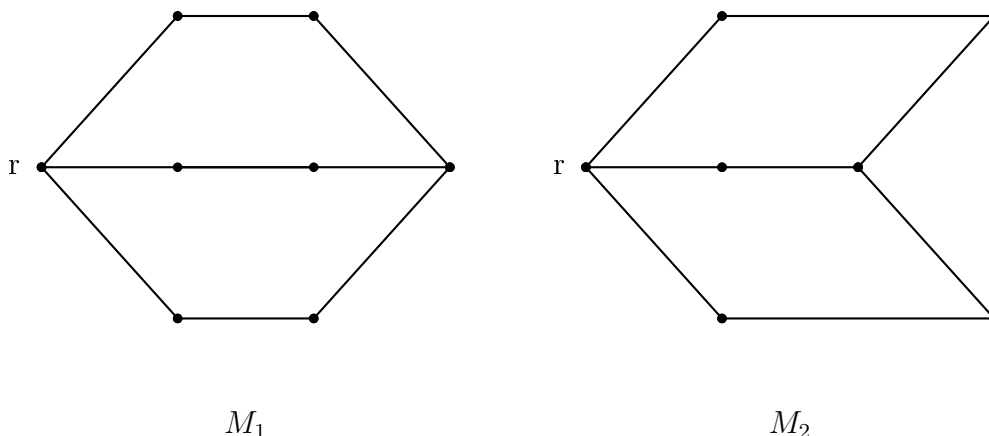


Figure 1.2 – Deux mineurs exclus à la classe des graphes GTSP totalement-équilibrés. Pour ces deux graphes (pondérés avec les longueurs $\mathbf{1}$), il existe une couverture fractionnaire en tours fermés contenant r , de coût inférieur au plus court tour de Steiner. Par exemple, M_1 peut être couvert par trois circuits de longueur 6, pondérés avec un poids $1/2$, donnant une couverture fractionnaire de coût 9, alors que $GTSP((M_1, \mathbf{1}, r), V(M_1)) = 10$.

Pour ce que nous en savons cependant, il est ouvert de savoir si l'équilibre total de la fonction $GTSP_r$ se traduit par une bonne caractérisation pour le problème du voyageur de commerce. Plus précisément, le problème suivant semble ouvert :

[TBTSP] TSP totalement-équilibré

Données : Un espace métrique enraciné $M = (\{1, 2, \dots, n\}, d, r)$.

Promesse : $GTSP(K_n, d, r)$ est une fonction totalement-équilibrée

Résultat : Un tour sur V de longueur minimum.

Commentaires : NPP. L'appartenance à co-NPP semble ouverte.

Une des difficultés est que si $M = (V, d, r)$ est un espace métrique enraciné dans lequel $GTSP$ est totalement-équilibrée, nous ne voyons pas comment certifier polynomialement qu'un vecteur $x : V \rightarrow \mathbb{R}$ est un élément du cœur de $GTSP$, bien que la donnée du vecteur x soit de taille polynomiale. En effet, comment vérifier polynomialement que $x(U) \leq GTSP((V, d, r), U)$ pour tout $U \subseteq V$?

De même, pour un espace métrique enraciné donné, la question de la reconnaissance de l'équilibre (total) de $GTSP$, semble ouverte [121]³⁹. Pire, les questions de reconnaissance et d'optimisation semblent aussi ouvertes si l'on s'intéresse à la sous-modularité de la fonction $GTSP$ pour un espace métrique donné.

En conclusion de cette section, rappelons que même si elles se révélaient non-pertinentes du point de vue de la complexité algorithmique, les propriétés d'équilibre total et la sous-modularité de la fonction $GTSP$ sont fondamentales du point de vue de la théorie coopérative des jeux (voir Exemple 1.32).

1.4.3 Imputations entières dans le cœur

Dans cette section, nous étudions l'existence d'une imputation entière dans le cœur d'une fonction entière totalement-équilibrée. Nous observons qu'une telle imputation existe si la fonction est sous-cardinale ou si elle est taille-définie, mais nous donnons un contre-exemple dans le cas général.

Si dans la théorie duale, la problématique se formule naturellement en variables entières, dans la théorie classique, les questions d'intégralité des solutions ne sont souvent même pas évoquées. La raison est que le partage des coûts ou des gains se fait habituellement de manière monétaire (et donc continue). Cependant, certaines problématiques font apparaître des questions de partage de biens insécables, comme les sièges en théorie des votes :

Exemple 1.51 *Soit $p : V \rightarrow [0, 1]$ tel que $p(V) = 1$ et D un nombre entier. On montre facilement qu'il existe toujours $x : V \rightarrow \mathbb{N}$ tel que $x(V) = D$ et tel que $\lfloor D \cdot p_v \rfloor \leq x_v \leq \lceil D \cdot p_v \rceil$. Mais peut-on partager les D sièges entre les partis de telle manière qu'aucun sous-ensemble U de partis ne puisse s'estimer laissé et ait intérêt à former une coalition pour obtenir plus de sièges⁴⁰ ? Autrement dit, existe-t-il*

³⁹Okamoto prouve la NP-complétude de la reconnaissance de l'équilibre pour le jeu du TSP enraciné (*i.e.* pas pour la relaxation graphique et donc pas pour la fonction $GTSP$). Sa preuve utilise la violation des inégalités triangulaires.

⁴⁰Notons que cet exemple n'est pas vraiment réaliste car tous les sous-ensembles de partis ne sont pas susceptibles de former des coalitions. Les solutions entières dans le cadre des jeux coopératifs sont toutefois digne d'un grand intérêt (Annexe B). Pour pallier l'irréalisme d'un modèle autorisant la coopération de tous les sous-ensembles dans le cadre politique, il serait intéressant de restreindre la coopération à une famille laminaire de sous-ensembles de V . Certains jeux à coopération restreinte aux cliques d'un graphe sont discutés dans le Chapitre 4. D'autres jeux à coopération restreinte sont étudiés dans [19].

$x : V \rightarrow \mathbb{N}$ tel que $x(U) \geq \lfloor D \times p(U) \rfloor$ pour tout $U \subseteq V$.

Une première condition nécessaire pour que le cœur de f ait une imputation (entière) est que le jeu soit équilibré. Une deuxième est que $f(V) \in \mathbb{N}$. Cela n'est clairement pas suffisant comme le montre l'exemple suivant, (construit à partir de l'Exemple 1.51 en prenant $|V| = 4$ et $p_v = 1/4$ pour tout $v \in V$ et $D = 2$) :

Exemple 1.52 Soit $V = [1..4]$ et f définie par

$$(1.62) \quad f(S) := \begin{cases} 0 & \text{si } |S| = 0 \\ 1 & \text{si } |S| = 1 \text{ ou } 2 \\ 2 & \text{si } |S| = 3 \text{ ou } 4 \end{cases}$$

Alors f est équilibrée et entière mais n'a pas de solution entière dans son cœur (qui consiste en un seul point : $x = \mathbf{1}/2$).

Nous montrons dans la Section 1.4.3.1, qu'en un sens, une condition nécessaire pour l'existence d'une imputation entière est que f soit totalement équilibrée et entière.

1.4.3.1 Troncation totalement-équilibrée entière

De manière similaire aux coupes de Gomory-Chvátal, on peut, à partir des valeurs de f , inférer d'autres inégalités qui donnent des bornes supérieures au programme $\max\{\mathbf{1}^T x \mid x \in P_f \cap \mathbb{N}^V\}$:

Ingrédient 1.53 Si λ est un recouvrement fractionnaire de U , on a

$$(1.63) \quad x(U) \leq \left\lfloor \sum_{S \subseteq U} \lambda_S f(S) \right\rfloor$$

pour tout x entier dans P_f . □

Cependant, il ne suffit pas de prendre la partie entière de chaque valeur de la troncation totalement-équilibrée pour tirer tout son arôme de l'Ingrédient 1.53 (voir Exemple 1.52) : contrairement aux troncations sous-additive et totalement-équilibrée, il faut calculer la troncation totalement-équilibrée entière récursivement en appliquant l'Ingrédient 1.53. Pour $f : \mathcal{P}(V) \rightarrow \mathbb{R}_+$, on suppose, quitte à d'abord remplacer $f(U)$ par $\min_{T \supseteq U} f(T)$, que f est non-décroissante. On définit alors $\bar{f}^{[\cdot]}$ récursivement par :

$$(1.64) \quad \bar{f}^{[\cdot]}(U) := \min\{\lfloor f(U) \rfloor, \min\{\sum_{T \subsetneq U} [\lambda_T \bar{f}^{[\cdot]}(T)] \mid \lambda \text{ est une partition fractionnaire de } U\}\}$$

Par définition, $\bar{f}^{\lfloor \cdot \rfloor}$ est totalement-équilibrée et entière. C'est en fait l'unique fonction maximale avec cette propriété :

Proposition 1.54 *Soit $f : \mathcal{P}(V) \rightarrow \mathbb{R}_+$, et $f' : \mathcal{P}(V) \rightarrow \mathbb{N}$ totalement-équilibrée telle que $f' \leq f$, alors $f' \leq \bar{f}^{\lfloor \cdot \rfloor}$.*

Preuve Soit $f : V \rightarrow \mathbb{R}_+$ et f' totalement-équilibrée entière telle que $f' \leq f$. Supposons que $f'(U) > \bar{f}^{\lfloor \cdot \rfloor}(U)$ pour un certain $U \subseteq V$ et choisissons U minimal par inclusion. Si $\bar{f}^{\lfloor \cdot \rfloor}(U) = \lfloor f(U) \rfloor$ on a aussi $f'(U) \leq \bar{f}^{\lfloor \cdot \rfloor}(U)$, contradiction. Il existe donc une partition fractionnaire λ de U telle que $\bar{f}^{\lfloor \cdot \rfloor}(U) = \lfloor \sum_{T \subset U} \lambda_T \bar{f}^{\lfloor \cdot \rfloor}(T) \rfloor$. Par minimalité de U , on a aussi $f'(U) \leq \sum_{T \subset U} \lambda_T \bar{f}^{\lfloor \cdot \rfloor}(T)$, contradiction. \square

Le principal intérêt de la troncation totalement-équilibrée entière est donné par la Proposition 1.55 qui montre, en un sens, qu'il est nécessaire que f soit totalement-équilibrée et entière pour qu'elle ait une imputation entière dans son cœur. Plus précisément,

Proposition 1.55 *Pour tout $f : \mathcal{P}(V) \rightarrow \mathbb{R}_+$, on a*

$$P_f \cap \mathbb{N}^V = P_{\bar{f}^{\lfloor \cdot \rfloor}} \cap \mathbb{N}^V$$

Preuve L'inclusion $P_f \cap \mathbb{N}^V \supseteq P_{\bar{f}^{\lfloor \cdot \rfloor}} \cap \mathbb{N}^V$ vient de $\bar{f}^{\lfloor \cdot \rfloor} \leq f$. Pour l'autre inclusion, considérons $x \in P_f \cap \mathbb{N}^V$. La fonction $x : \mathcal{P}(V) \rightarrow \mathbb{N}$ qui à U associe $x(U)$ est entière. Comme elle est linéaire, elle est aussi totalement-équilibrée. D'après la Proposition 1.54, on a $x \leq \bar{f}^{\lfloor \cdot \rfloor}$ et donc $x \in P_{\bar{f}^{\lfloor \cdot \rfloor}} \cap \mathbb{N}^V$. \square

Dans la Section 1.4.3.4, nous montrons cependant que cette troncation n'est pas suffisante pour caractériser l'existence de points entiers dans le cœur d'une fonction f .

1.4.3.2 Fonction totalement-équilibrée sous-cardinale

Dans cette section, nous montrons que le cœur d'une fonction f sous-cardinale, entière et totalement-équilibrée contient toujours un point entier. D'après la Proposition 1.43, trouver un tel point est équivalent à trouver une arête de cardinalité maximum dans un hypergraphe héréditaire lorsque l'on dispose d'un oracle de rang. Or, cette tâche est triviale même dans un hypergraphe quelconque :

Proposition 1.56 *Soit $H = (V, \mathcal{E})$ un hypergraphe. Si l'on dispose d'un oracle de rang pour H , on peut trouver une arête de cardinalité maximum de \mathcal{E} , en temps $\mathcal{O}(|V|)$.*

Preuve Construisons cette arête de manière gloutonne : Posons d'abord $U := V$. Puis, pour chaque élément v de U , enlevons v de U ssi $r(U - v) = r(U)$. \square

Si le fait que H soit héréditaire n'a aucune importance pour le bon déroulement de l'algorithme, la Proposition 1.43 montre que l'hérédité est nécessaire pour que le vecteur caractéristique de l'arête ainsi trouvée soit dans le cœur de la fonction $r(H)$. En conséquence des Propositions 1.56 et 1.43, on a :

Corollaire 1.57 *Soit $f : \mathcal{P}(V) \rightarrow \mathbb{N}$ totalement-équilibrée et sous-cardinale. Alors le programme $\max\{c^T x \mid x \in P_f\}$ a la propriété de dualité entière pour tout $c \in \{0, 1\}^V$. En particulier le cœur de f contient un point entier.* \square

Bien que les inégalités de rang d'un hypergraphe héréditaire définissent un système ayant la propriété de dualité entière pour toute fonction objectif $c \in \{0, 1\}^V$, ce système ne définit pas, en général, un polyèdre entier :

Exemple 1.58 *Soit $G = C_5 + v$ le graphe à 6 sommets qui a un C_5 induit et un sommet isolé. Soit $c \in \mathbb{R}^V$ tel que $c_u = 1$ pour tout $u \in C_5$ et $c_v = 2$.*

$$\max\{c^T x \mid x(U) \leq \omega(G[U]) \text{ pour tout } U \subseteq V\} = 7/3$$

est atteint par $x^ = \mathbf{1}/3$.*

1.4.3.3 Cœur d'une fonction totalement-équilibrée taille-définie

Une fonction est **taille-définie**⁴¹ si $f(U)$ ne dépend que de $|U|$, c'est-à-dire s'il existe $\phi : [0..|V|] \rightarrow \mathbb{R}$ telle que $f(U) = \phi(|U|)$ pour tout $U \subseteq V$. Nous observons que le cœur d'une fonction taille-définie totalement-équilibrée entière contient un point entier, mais ne définit pas toujours un polyèdre entier.

Proposition 1.59 *Soit f une fonction taille-définie totalement-équilibrée entière définie par ϕ . Alors il existe un point entier dans le cœur de f .*

Preuve Posons $t := \lceil \phi(|V|)/|V| \rceil$ et choisissons $S \subseteq V$ tel que le vecteur x défini par

$$x_v := \begin{cases} t & \text{si } v \in S \\ t - 1 & \text{si } v \notin S \end{cases}$$

satisfasse $x(V) = f(V)$. Montrons que x est dans le cœur de f . Sinon, parmi les $Q \subset V$ tels que $x(Q) > \phi(|Q|)$, choisissons Q de cardinalité maximum. On a $\phi(|Q|) + 1 \leq x(Q) \leq |Q|t$ et donc $\phi(|Q|)/|Q| < t$. Soit $v \in V \setminus Q$. Considérons la partition fractionnaire de $Q + v$ par tous les sous-ensembles de taille $|Q|$. Par équilibre total de f , on a :

$$\phi(|Q + v|) \leq \frac{|Q| + 1}{|Q|} \phi(|Q|)$$

et donc

$$\phi(|Q + v|) - \phi(|Q|) \leq \frac{\phi(|Q|)}{|Q|} < t$$

⁴¹ *size-defined.*

Par intégralité de ϕ , on a donc $\phi(|Q+v|) \leq \phi(|Q|) + t - 1$. Finalement,

$$x(Q+v) \geq x(Q) + t - 1 > \phi(|Q|) + t - 1 \geq \phi(|Q+v|)$$

ce qui contredit la maximalité de Q . \square

Observons que le cœur d'une fonction taille-définie totalement-équilibrée entière n'est pas toujours un polyèdre entier :

Exemple 1.60 Soit f la fonction définie sur $V = \{1, 2, 3, 4\}$ par

$$(1.65) \quad f(U) := \begin{cases} 0 & \text{si } |U| = 0 \\ 2 & \text{si } |U| = 1 \text{ ou } 2 \\ 3 & \text{si } |U| = 3 \text{ ou } 4 \end{cases}$$

$\max\{(2, 1, 1, 1)^T x \mid x \in P_f\}$ est atteint avec $x = 1/2(3, 1, 1, 1)$. La valeur objectif est $9/2$. Une solution duale optimale est donnée par $y_V = y_{1,2} = y_{1,3} = y_{1,4} = 1/2$.

1.4.3.4 Cœur d'une fonction totalement-équilibrée générale

Dans cette section, nous donnons un exemple de fonction totalement-équilibrée entière qui n'a pas d'imputation entière dans son cœur.

Au vu des résultats concernant le cas sous-cardinal et taille-défini, nous avons conjecturé qu'il existe une solution entière dans le cœur d'une fonction totalement-équilibrée entière. Cette conjecture était l'une des motivations pour étudier les liens entre la théorie des jeux coopératifs et la combinatoire polyédrale. Cette conjecture est malheureusement fautive, comme le montre l'exemple, dû à Gijswijt [83] que nous décrivons dans le reste de cette section.

Soit $V := \{1, 2, 3, 4, 5, 6\}$ et A la matrice $V \times V$ suivante :

$$(1.66) \quad A := \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Pour $i \in [1..6]$, soit T_i le sous-ensemble de V support de la i -ème ligne de A et $T := \{T_1, T_2, \dots, T_6\}$. Pour $S \subseteq V$, soit

$$(1.67) \quad f(S) := \begin{cases} 0 & \text{si } |S| = 0 \\ 4 & \text{si } |S| = 1 \text{ ou } 2 \\ 4 & \text{si } |S| = 3 \text{ et } S \in T \\ 6 & \text{si } |S| = 3 \text{ et } S \notin T \\ 6 & \text{si } |S| = 4 \\ 7 & \text{si } |S| = 5 \\ 8 & \text{si } |S| = 6 \end{cases}$$

Proposition 1.61 *La fonction f est totalement-équilibrée, mais il n'existe pas de solution entière dans son cœur.*

Preuve Soit

$$P_f := \{x \in R^V \mid x \geq 0, x(U) \leq f(U) \text{ pour tout } U \subseteq V\}$$

Montrons tout d'abord que f est totalement-équilibrée. Grâce au Théorème de Bondareva-Shapley, il suffit de vérifier que pour tout $S \subseteq V$, il existe $x_S^* \in P_f$ tel que $x_S^* = f(S)$. Comme la fonction f est invariante par l'action du groupe des permutations circulaires sur les 6 éléments de V , on n'a pas besoin de donner un tel x^* pour chacun des 2^6 sous-ensembles de V . A permutation circulaire près, on trouve un seul ensemble de taille 5 et trois ensembles de taille 4. Toutes les configurations de $S \subseteq V$ sont donc couvertes dans (1.68).

$$(1.68) \quad x_S^* := \begin{cases} 4/3 \times (1, 1, 1, 1, 1) & \text{si } S = V \\ (1, 1, 2, 2, 1, 0) & \text{si } S = \{1, 2, 3, 4, 5\} \\ (0, 1, 2, 2, 1, 0) & \text{si } S = \{2, 3, 4, 5\} \\ (1, 0, 2, 2, 1, 0) & \text{si } S = \{1, 2, 4, 5\} \\ (1, 2, 0, 1, 2, 0) & \text{si } S = \{1, 2, 4, 5\} \\ (1, 1, 0, 2, 0, 0) & \text{si } S = T_1 \\ 2 \times \mathbf{1}_S & \text{si } |S| = 3 \quad \text{et } S \notin T \\ 2 \times \mathbf{1}_S & \text{si } |S| = 2 \\ 4 \times \mathbf{1}_S & \text{si } |S| = 1 \end{cases}$$

Montrons maintenant que f n'a pas de solution entière dans son cœur. Soit $B_f := \{x \in P \mid x(V) = f(V)\}$. Il suffit de vérifier que $B_f = \{x_V^*\}$. Montrons que pour tout $x \in B_f$ et tout $i \in [1..6]$, on a $f(T_i) = x(T_i)$. Par contradiction, supposons $f(T_i) > x(T_i)$ pour $i \in [1..6]$. En ajoutant les 5 autres inégalités $f(T_j) \geq x(T_j)$, on a :

$$(1.69) \quad x(V) = f(V) = 8 = 1/3(f(T_1) + \dots + f(T_6)) > 1/3(x(S_1) + \dots + x(S_6)) = x(V)$$

ce qui est impossible. On a donc $f(T_i) = x(T_i) = 4$ et donc $Ax = (4, 4, 4, 4, 4, 4)$. Comme A est non-singulière (*e.g.* $\det(A) = -9$), x^* est la seule solution du système $Ax = (4, 4, 4, 4, 4, 4)$. \square

1.5 Une classification des problèmes d'ordonnement chromatique

La théorie de la coloration peut être vue comme l'étude des partitions d'un ensemble fini d'objets sous contraintes d'incompatibilité. La théorie de l'ordonnement, quant à elle, s'intéresse principalement aux permutations d'un ensemble fini

d'objets. Ces deux disciplines débordent constamment l'une dans l'autre, créant l'*ordonnement chromatique*⁴². Après avoir introduit quelques concepts de l'ordonnement chromatique par batchs (Section 1.5.1), nous proposons une classification de type $\alpha|\beta|\gamma$ pour les problèmes d'ordonnement chromatique traités dans ce mémoire (Section 1.5.2).

L'intérêt de cette classification est double : premièrement, elle devrait favoriser la communication et participer à l'interfécondation ordonnancement-coloration. Deuxièmement, et c'était notre motivation originelle, cette classification participe à la mise en œuvre du projet de synthèse présenté dans l'Annexe A.

1.5.1 Introduction à l'ordonnement chromatique par batch

La théorie de l'ordonnement traite de l'organisation temporelle du travail. On dispose de *machines* sur lesquelles des *tâches* doivent être exécutées. En pratique les machines peuvent être des processeurs sur lesquels on effectue des calculs (qui sont alors les tâches). Le couple *machines-tâches* peut aussi signifier *camions-marchandises* dans les problèmes de tournées de véhicules, *traducteurs-textes* dans une entreprise de traduction, ou encore *enseignants-cours* dans un lycée ou une université. Ordonner consiste donc à affecter les tâches aux machines et à décider à quel moment chaque tâche sera effectuée.

Un des intérêts de la théorie de l'ordonnement est de pouvoir exprimer, dans un même formalisme mathématique, des problèmes issus de contextes concrets différents, mais relevant d'une même problématique. Nous référons à [32, 21, 112] pour une présentation "classique" des problèmes d'ordonnement.

Les ordonnements sont des objets mathématiques complexes, que l'on formalise en général dans le langage de la programmation linéaire en nombres entiers. Cette modélisation semble superflue dans le cadre de ce mémoire, car un ordonnancement admet aussi une représentation graphique simple et intuitive : son diagramme de Gant. Nous nous contentons donc d'illustrer les concepts sur quelques exemples. Commençons par un cas que nous connaissons bien : l'organisation du travail d'un chercheur très en retard dans son planning.

Exemple 1.62 Ordonnement classique⁴³

Marla Tex, professeure à l'Université Libre Mondiale, a passé les dernières semaines à essayer de savoir si $P=NP$. Elle se retrouve le dimanche 31 mars au soir avec la liste suivante de choses à faire d'ici le 10 avril au soir (veille de son départ pour une conférence à Tokyo).

⁴²Le terme *Chromatic Scheduling* est promu par Dominique de Werra depuis plus de 30 ans [54, 108, 63, 62, 59, 56, 57, 61, 58].

⁴³L'exemple 1.62 illustre le problème noté $1|r_j|\sum w_j U_j$ dans la notation $\alpha|\beta|\gamma$ (présentée dans la Section 1.5.2).

Identifiant	Tâche	Disponibilité r_j	Echéance d_j	Durée p_j	Valeur w_j
T_1	Paris	–	6	2	2
T_2	Waterloo	–	4	2	3
T_3	Europe	–	10	3	30
T_4	JCRA	3	4	1	1
T_5	PLS	–	9	2	2
T_6	Mari	5	7	1	2
T_7	Amant	–	10	1	2
T_8	Chalet	5	7	2	3

Tableau 1.1 – Les engagements de Marla : Une instance du problème $1|r_j|\sum w_jU_j$

- Ecrire un résumé pour une conférence à Paris.
- Ecrire un résumé pour une conférence à Waterloo.
- Boucler un dossier de demande de financement européen (Europe).
- Participer aux Journée de Combinatoire Rhône-Alpes (JCRA).
- Ecrire un article de vulgarisation pour le journal Pour la Science (PLS).
- Passer une journée de week-end avec son mari au bord d'un lac.
- Passer une journée avec son amant.
- Passer le week-end dans un chalet en montagne avec des amis.

A chaque activité j est associée un intervalle de temps $[r_j, d_j]$ durant lequel l'activité peut être réalisée ainsi qu'un temps d'opération p_j (l'activité j ne peut être commencée avant la date r_j , ne peut être terminée après la date d_j et dure p_j journées). Les données (r_j, d_j, p_j) sont résumées dans le Tableau 1.1 (dans lequel " $r_j = -$ " équivaut à $r_j = 0$). Marla s'aperçoit qu'elle ne pourra pas répondre à tous ses engagements avant son départ, mais ne se résout pas pour autant à réduire son temps de sommeil ou à bâcler ses tâches. Elle décide d'estimer la valeur w_j qu'a pour elle la réalisation de chacune de ses activités et cherche comment s'organiser pour réaliser un ensemble d'engagements de valeur totale maximum avant leur échéance.

Après une minute de réflexion, elle arrive à l'ordonnancement de la Figure 1.3, (dans lequel certaines tâches ne sont pas réalisées, car Marla estime que les tâches réalisées en retard ne valent pas mieux que des tâches non réalisées). La fin du traitement de la tâche j est noté C_j .

Dans la notation $\alpha|\beta|\gamma$ (section 1.5.2) la modélisation qu'elle a fait de son problème d'organisation se note

$$1|r_j|\sum w_jU_j$$

Marla n'est pas vraiment intéressée pour trouver une solution optimale de son problème d'organisation personnelle (dont la modélisation est relativement subjective,

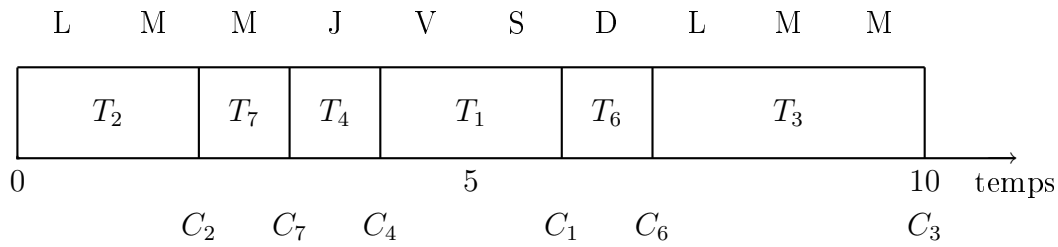


Figure 1.3 – Le premier planning de Marla Tex

en particulier à cause des valeurs w_j). Par curiosité, elle consulte quand même le site web de Brucker et Knust [34] pour connaître la complexité de sa formalisation. Elle se rend compte qu'il y a au moins un aspect du problème qu'elle n'a pas précisé : la préemption. La préemption, consiste à arrêter une tâche en cours pour en commencer une autre avant de revenir à la première. L'ordonnement de la Figure 1.3 est non-préemptif. La modélisation de Marla est NP-dure si l'on se restreint aux ordonnancements non-préemptifs mais pseudo-polynomial si la préemption est autorisée [34].

Lundi matin, lors de la préparation du résumé de sa conférence de Waterloo (voir Exemple 1.62), Marla traite du problème suivant :

Exemple 1.63 *Ordonnement en batchs parallèles avec un graphe d'intervalle comme contraintes de compatibilité, et nombre de tâches bornées dans un batch*⁴⁴. $B1 | col(G = co\text{-}interval), b \leq n, p\text{-}batch | C_{\max}$

Un ensemble de n pièces en fonte I_1, \dots, I_n doivent être cuites dans un four en cloche. Le four peut cuire jusqu'à b pièces à la fois. Chaque pièce possède un temps de cuisson minimal p_j . L'organisation du four consiste en une succession de phases de cuisson durant lesquelles on ne peut pas ajouter ni enlever des pièces sous la cloche. Lorsqu'on cuit un ensemble \mathcal{J} de pièces en même temps, on les cuit donc pendant le maximum des temps de cuisson : l'ensemble \mathcal{J} reste dans le four pendant $\max_{j \in \mathcal{J}} p_j$. En plus du temps de cuisson, les pièces à cuire doivent l'être à des températures spécifiques. A chaque pièce est associé un intervalle de températures admissibles. Pour que des pièces puissent être cuites ensemble, il est nécessaire que leurs intervalles de températures aient une intersection commune. Le but est de trouver un planning de cuisson qui utilise le four pendant le plus court temps total possible. De manière équivalente, on veut avoir fini de cuire la dernière pièce le plus tôt possible.

La Figure 1.4 présente une instance de ce problème. Chaque tâche y est représentée par un intervalle, dont la hauteur représente le temps minimal de cuisson et les extrémités latérales représentent les températures minimales et maximales de

⁴⁴L'étude de ce problème a fait l'objet d'une publication commune avec Gerd Finke, András Sebő et Maurice Queyranne [76].

cuisson.

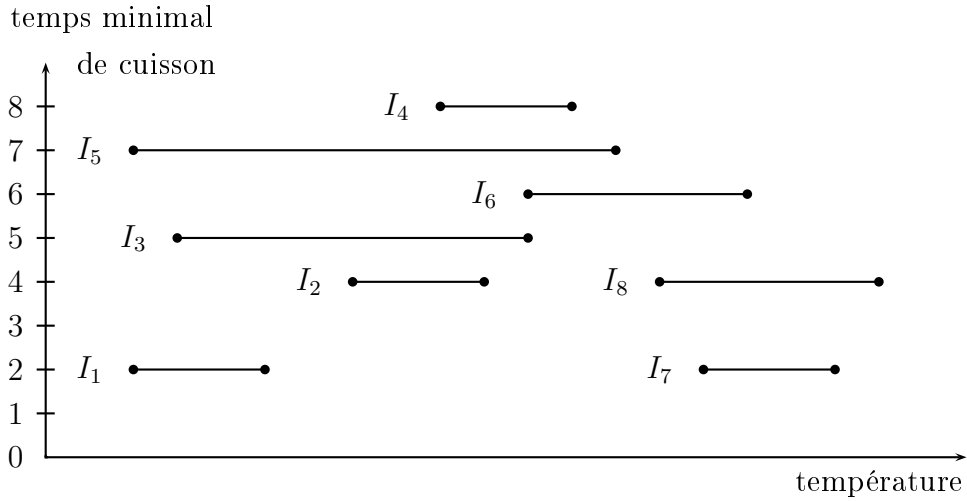


Figure 1.4 – Une instance du problème $1|G = \text{co-interval}, b = 3, p\text{-batch}|C_{\max}$

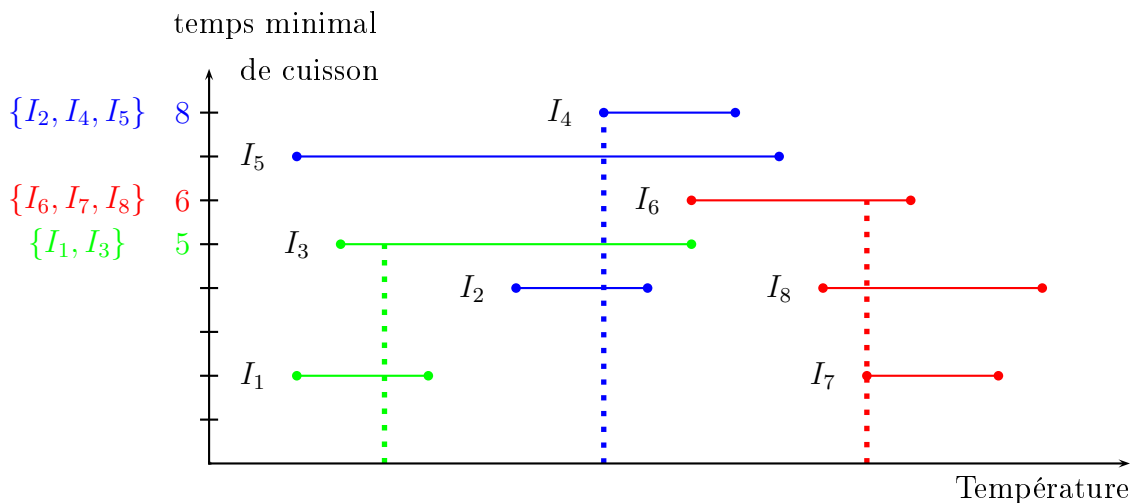


Figure 1.5 – Une partition des intervalles en batches de taille au plus 3.

Un planning de cuisson peut alors être représenté par un ordonnancement par batch, dans lequel plusieurs tâches sont opérées en même temps, comme dans les Figures 1.5 et 1.6. Dans l'extension de la notation $\alpha|\beta|\gamma$ que nous proposons (Section 1.5.2) ce problème se note

$$B1|col(G = \text{co-interval}), b \leq n, p_S = p\text{-batch}|C_{\max}$$

Du fait des contraintes d'incompatibilités, les résultats concernant ce problème ne sont pas facilement accessibles dans la littérature issue de l'ordonnancement⁴⁵.

⁴⁵En particulier, ils ne sont pas référencés dans [34] (ni ailleurs semble-t-il).

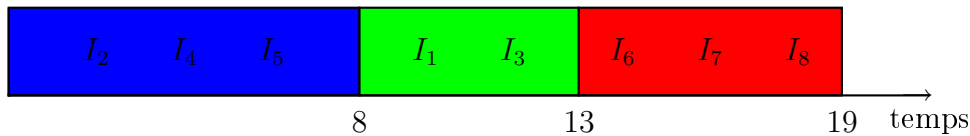


Figure 1.6 – Un ordonnancement correspondant à la partition de la Figure 1.5

Cependant (à force de transitivité bibliographique et de participation à des conférences), on trouve de nombreux résultats concernant ce problème dans la littérature. Une des raisons expliquant la difficulté de l'accès aux résultats est la largeur du spectre des applications. En effet, ce problème intervient dans des contextes de tournées de véhicules avec fenêtres de temps⁴⁶ [68], de planification de production [76, 107], de réseaux de communication [14]. Ce problème est polynomial si l'on suppose que le temps d'opération d'un batch est unitaire [125, 113, 22, 76, 107]. Il est polynomial si l'on supprime les contraintes de cardinalité [68, 76, 38, 14]. Il est polynomial si l'on supprime les contraintes de compatibilité entre les intervalles (voir [33] et/ou Section 4.4.3). La complexité du problème général reste toutefois ouverte.

Exemple 1.64 *Ordonnement par batches avec économie d'échelle.*

Lundi midi, alors que Marla discute d'ordonnement par batch avec ses collègues, elle réalise que ce mode d'organisation peut s'appliquer à l'organisation de son propre travail (Exemple 1.62) : elle peut faire des économies de temps en préparant deux résumés similaires pour ses conférences de Paris et de Waterloo. Bien que l'un des résumés doive être écrit en anglais et l'autre en français, il lui suffit d'en préparer un puis de le traduire. Elle estime à trois jours (au lieu de quatre, comme l'indique le Tableau 1.1), le temps de préparation de ses deux résumés. Cela ne colle pas avec son modèle en batches parallèles dans lequel le temps d'opération serait de deux jours⁴⁷. Elle réalise aussi qu'il y a un lac près du chalet où elle avait renoncé à aller avec ses amis ce week-end. Elle appelle son mari qui accepte d'annuler leur journée ensemble pour se joindre au groupe d'amis pour une journée⁴⁸. Marla constate le lien entre la notion d'incompatibilité pour les intervalles de l'Exemple 1.63 et l'incompatibilité de certaines activités dans son propre planning. Marla arrive finalement au planning suivant :

⁴⁶La compatibilité des intervalles correspond aux fenêtres de temps (il faut supposer que la durée des trajets est petite devant la taille des fenêtres de temps). Les temps d'opération (qu'il faut toutefois généraliser à des fonctions d'ensemble comme dans (1.60) page 66) correspondent à la longueur des trajets. La minimisation du makespan correspond à la minimisation de la longueur totale des trajets.

⁴⁷Par ailleurs, dans un ordonnancement par batches, les deux tâches devraient être traitées simultanément alors que Marla n'a nul besoin de faire ses deux résumés ensembles, ni même l'un à la suite de l'autre. Ce genre de variantes n'est cependant pas discuté dans ce mémoire.

⁴⁸Dans l'ivresse qu'elle éprouve à gagner du temps sur son planning, elle imagine un instant gagner encore un jour en invitant son amant en week-end, mais se rappelle que ni son mari ni ses amis ne sont au courant de cette relation et que c'est peut-être mieux ainsi.

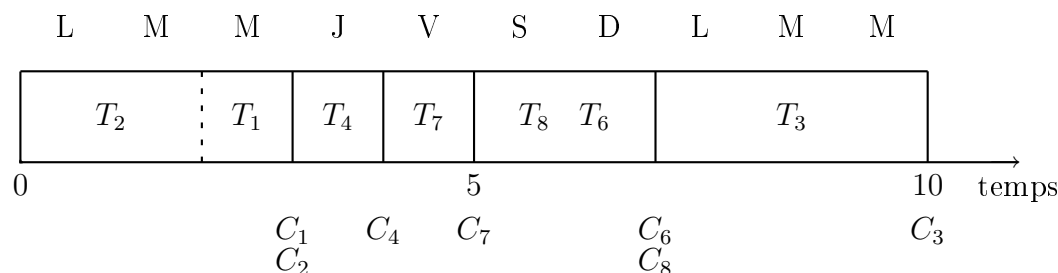


Figure 1.7 – Le planning de Marla Tex avec économies d'échelle

Marla est très intriguée par son nouveau planning car elle sent que le gain de temps associé au traitement groupé des tâches est un phénomène général. Elle sent cependant qu'il n'est pas aisé de quantifier l'économie d'échelle de manière générale, car celle-ci dépend du type de machine considéré (voir Section 2.4). Après avoir remanié son planning, elle appelle son amant pour l'informer du changement de programme. Celui-ci lui répond qu'il n'est pas disponible selon son bon vouloir et lui raccroche au nez. Découragée par la complexité de la logistique qu'implique l'économie d'échelle, Marla se remet à préparer ses conférences...

1.5.2 Classification $\alpha|\beta|\gamma$ pour l'ordonnancement chromatique

Dans cette section, nous proposons une classification de type $\alpha|\beta|\gamma$ pour les problèmes d'ordonnancement pas batch avec contraintes d'incompatibilité. Les deux principales originalités par rapport à la notation classique sont le champ β_{Incomp} qui décrit l'incompatibilité des tâches (voir Chapitres 3, 4 et 5) et le champ, $\beta_{ProcTime}$ qui, pour prendre en compte l'économie d'échelle, n'est plus défini pour chaque tâche mais pour chaque sous-ensemble de tâches (voir Chapitre 4).

1.5.2.1 Champ α : ressources machines

Dans ce mémoire, nous étudions seulement le cas de l'ordonnancement par batch sur une machine, le champ α prendra toujours la valeur $B1$ (B pour batch, 1 pour une machine)⁴⁹. Une machine par batch opère en traitant plusieurs tâches simultanément. Un ordonnancement par batch sur une machine se résume donc à une partition ordonnée $\mathcal{Q} = (K_1, K_2, \dots, K_k)$ de V .

En raison de la Remarque 1.65 ci-dessous, nous définissons maintenant la machine unique⁵⁰ et les machines parallèles. Dans les problèmes classiques, un ordonnance-

⁴⁹Cette notation n'est pas classique, Brucker et Knust [34] la notent 1 et spécifient dans le champ β que la machine opère par batches.

⁵⁰*Single machine.*

ment sur une machine est une fonction $\mathcal{O} : V \rightarrow \mathbb{N}$ telle que $\mathcal{O}(v) \cap \mathcal{O}(w) = \emptyset$. Ici, $\mathcal{O}(v)$ donne l'ensemble des instants pendant lesquels la tâche v est traitée. La contrainte d'intersection vide traduit le fait que la machine ne peut traiter qu'une seule tâche à la fois. Un ordonnancement sur m machines parallèles est alors une fonction de $\mathcal{O} : V \times [1..m] \rightarrow \mathbb{N}$ tel que $\mathcal{O}(v, j) \cap \mathcal{O}(w, j) = \emptyset$ pour tous $v, w \in V$ et tout $j \in [1..m]$ et tel que $\mathcal{O}(v, i) \cap \mathcal{O}(v, j) = \emptyset$ pour tous $v \in V$ et tout $i, j \in [1..m]$. Autrement dit, à chaque instant, *chaque machine exécute au plus une tâche et chaque tâche est exécutée par au plus une machine*.

Remarque 1.65 [22, 34] *Dans le cas de temps d'opération unitaires, les ordonnancements par batch sur une machine sont en bijections avec les ordonnancements sur machines parallèles. De plus, contraindre les batchs à contenir au plus b tâches équivaut à considérer b machines parallèles, tandis que des batchs non contraints en cardinalité équivalent à un nombre infini de machines parallèles.*

1.5.2.2 Champ β : caractéristiques des tâches

– β_{Incomp} indique les *contraintes d'incompatibilité*. Celles-ci décrivent les sous-ensembles de tâches qui ne peuvent pas être traitées simultanément. Dans ce mémoire, nous supposons que l'incompatibilité est donnée par un graphe G ayant l'ensemble des tâches pour sommets et dans lequel deux sommets forment une arête si ces deux tâches ne peuvent pas être placées dans un même batch. L'ensemble des tâches traitées à un instant donné doit donc induire un stable dans le graphe d'incompatibilité. Le cas par défaut (\emptyset) est que toutes les paires de tâches sont compatibles, autrement dit que le graphe d'incompatibilité ne contient aucune arête. La partition définie par un ordonnancement par batchs est donc une coloration de G ⁵¹. Nous adoptons la notation proposée dans la Section 2.3.3 : les sous-ensembles de tâches admissibles dans un même batch (ou dans une même couleur) sont notés $col(G)$.

De manière alternative au graphe d'incompatibilité, on pourrait considérer le graphe de compatibilité \overline{G} , complément de G . Un sous-ensemble de sommets de V est une clique dans \overline{G} si et seulement si c'est un stable de G . Un ordonnancement par batch consiste donc en une partition ordonnée de $V(\overline{G})$ en cliques⁵².

⁵¹Plus généralement, les contraintes sont décrites par un hypergraphe d'incompatibilité $H = (V, \mathcal{E})$. L'ensemble des tâches traitées à un instant donné ne doit alors contenir aucune arête de H . En plus de la difficulté conceptuelle, se pose le problème du codage de l'hypergraphe d'incompatibilité (qui peut avoir un nombre exponentiel d'arêtes). Pour simplifier la présentation, les discussions liées aux hypergraphes sont reportées aux sections "Pour aller plus loin" dans chaque chapitre de la deuxième partie de ce mémoire. Notons au passage qu'un hypergraphe d'incompatibilité permet de décrire une contrainte de type "cardinalité maximum" (l'hypergraphe est alors l'ensemble des circuit du matroïde uniforme \mathcal{U}_n^b).

⁵²Plus généralement, on peut aussi décrire la compatibilité par un hypergraphe H' dont les arêtes sont les sous-ensembles traitables simultanément. Dans le cas où H' est héréditaire, il est équivalent à un hypergraphe d'incompatibilité (et on a $H' = col(H)$, voir section 2.3).

La valeur prise par β_{Incomp} est une classe de graphes. Les classes utilisées dans ce mémoire sont présentées dans le Chapitre 2 et l'Annexe A.2 page 257.

- β_{Dispo} définit les **contraintes de disponibilité** des tâches. En général, pour chaque tâche, on décrit un sous-ensemble de \mathbb{N} pendant lequel cette tâche peut être opérée. Le cas par défaut (\emptyset) est que les tâches sont toutes disponibles à tous les instants. Le cas le plus général est qu'on associe une liste d'entiers à chaque tâche (List). Un cas intermédiaire intéressant est quand les tâches sont disponibles pendant un intervalle de temps (r_j, \bar{d}_j) , éventuellement non-contraint à droite (r_j) ou à gauche (\bar{d}_j). Ici r_j est la **date de disponibilité** et \bar{d}_j est la **date échu**. Un autre cas intéressant, qui ne s'applique que dans le cas de temps d'opérations unitaires, est que les tâches sont soit disponibles tout le temps soit déjà affectées : la liste est soit \mathbb{N} soit un singleton (PrExt). Le sous-cas de PrExt dans lequel au plus une tâche est préaffectée à chaque instant est $(1 - \text{PrExt})$. Le diagramme des réductions entre les contraintes de disponibilité est présenté dans la Figure 1.8. Les contraintes de disponibilité sont étudiées dans le Chapitre 5⁵³.

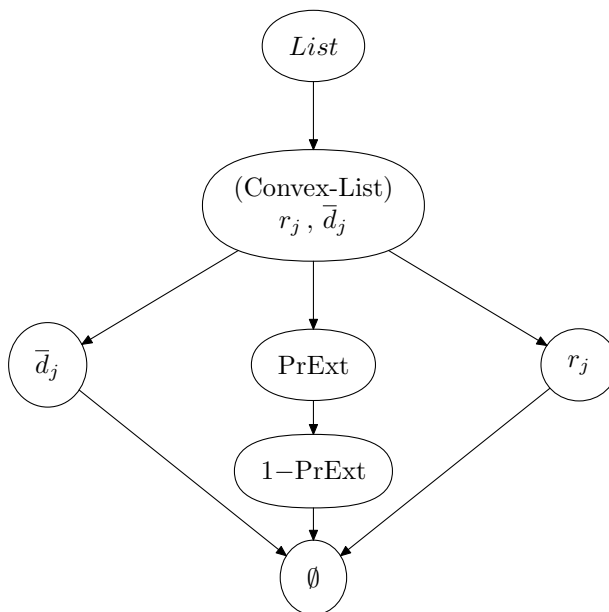


Figure 1.8 – Contraintes de disponibilité sur les tâches : coloration par liste.

⁵³La Figure 5.3 page 221 montre d'une part que les contraintes de disponibilité sont étudiées en ordonnancement chromatique avec une terminologie différente, et d'autre part que les contraintes de disponibilité sont intimement liées avec la fonction objectif (γ).

- β_{Card} définit les **contraintes de cardinalité** des batchs. Le cas par défaut (\emptyset) est qu'il n'y a pas de contraintes de taille. L'autre cas principal qui nous intéresse donne un majorant b à la taille d'un batch. On note $(b \leq n)$ si cette borne fait partie de l'entrée et $(b = \text{const})$ si c'est une constante. Parfois, l'entrée donne une borne différente pour chaque batch ($\{b_i\}_1^m$). Le diagramme des réductions entre les contraintes de cardinalité est présenté dans la Figure 1.9⁵⁴. Les contraintes de cardinalité (principalement $b \leq n$) sont étudiées dans le Chapitre 3.

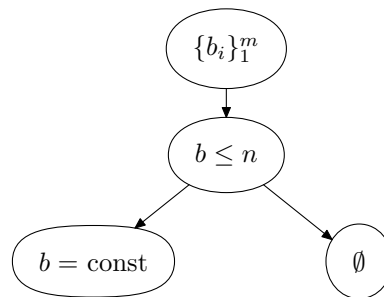


Figure 1.9 – Contraintes de cardinalité sur les batchs.

⁵⁴En ordonnancement chromatique, certains modèles considèrent aussi des bornes inférieures sur la taille d'un batch. Par exemple, le problème du couplage multidimensionnel [52] demande que tous les batchs soient de taille exactement b . La coloration équitable [23, 42] demande que les batchs aient une taille comprise entre $b - 1$ et b .

- β_{ProcTime} décrit le **temps d'opération**. Dans le cadre de ce mémoire, le temps d'opération est une fonction qui, à chaque sous-ensemble de tâches, $\mathcal{J} \subseteq V$ associe un temps d'opération, que nous supposons entier ou rationnel.

$$p : \mathcal{P}(V) \rightarrow \mathbb{N}$$

La valeur que prend le paramètre β_{ProcTime} est une classe de fonctions d'ensembles. Celles que nous utilisons sont décrites dans la Section 2.4, le Chapitre 4 et l'Annexe A.2 page 256. Il arrive, pour les classes les plus générales de fonctions d'ensemble, que celles-ci ne soient pas codables dans un espace polynomial en $|V|$ (par exemple, les fonctions sous-modulaires). Pour contourner ce problème, on suppose parfois que la fonction est donnée par un oracle de valeurs : étant donné $U \subseteq V$, l'oracle fournit la valeur $p(U)$ en temps constant.

Les problèmes d'ordonnancement chromatique comportant des temps d'exécution non-unitaires sont discutés dans le Chapitre 4.

- $\beta_{\text{Preemption}}$ Un ordonnancement est non-préemptif si chacune des tâches doit être exécutée pendant un intervalle (contigu) de temps. L'ordonnancement est **pré-emptif** si l'on peut interrompre l'opération d'une tâche pour la reprendre plus tard. La préemption peut être autorisée (*pmtn*), mais par défaut, elle ne l'est pas (\emptyset). Tous les problèmes étudiés dans ce mémoire le sont de manière non-préemptive, comme c'est habituellement le cas en ordonnancement par batches. Comme nous ne reviendrons pas sur ces résultats classiques importants, il nous faut mentionner que l'ordonnancement chromatique sur un nombre infini de machines parallèles est étudié dans ses versions préemptives (multi-coloration) et non-préemptive (coloration par intervalles⁵⁵). Dans le cas préemptif, le problème est soluble en temps polynomial sur les graphes parfaits [129]. Le cas non-préemptif est plus compliqué (voir [129], page 1151 et surtout [88]).

1.5.2.3 Champ γ : objectif

Etant donné un ordonnancement, on appelle C_j la date de fin d'opération de la tâche j . La qualité de l'ordonnancement est jugée par une fonction objectif qui est une fonction de $\mathbb{N}_+^V \rightarrow \mathbb{N}$ associant une valeur à la liste des C_j . Parfois la fonction objectif dépend de l'échéance d_j à laquelle la tâche était attendue. Notons la différence de nature entre les dates échues (\bar{d}_j), qui sont des contraintes "dures", et les dates d'échéance (d_j), qui ne sont pas des contraintes, mais servent à définir la fonction objectif.

Pour chaque tâche j , on définit

$$(1.70) \quad \text{Le retard} \quad T_j := \max\{0, C_j - d_j\}$$

$$(1.71) \quad \text{La pénalité unitaire} \quad U_j := \begin{cases} 0 & \text{si } C_j \leq d_j \\ 1 & \text{sinon} \end{cases}$$

⁵⁵*interval coloring.*

Remarquons que l'objectif U_j est la traduction mathématique des devises peu communes : “*Tard ne vaut pas mieux que jamais*” ou encore, “*Maintenant que tu es en retard, prends tout ton temps*”. En particulier, pour cet objectif là, on peut se restreindre à ordonner un sous-ensemble de tâches “à l’heure” et négliger les autres (voir Exemple 1.62).

En général, la fonction objectif est donnée par une formule simple du style $C_{max} := \max_{j=1}^n C_j$ ou $\sum C_j := \sum_{j=1}^n C_j$. Souvent, on pondère l'objectif par l'importance des tâches. Pour cela, on définit des poids $w : \{1, 2 \dots n\} \rightarrow \mathbb{N}$ et on utilise un objectif du style $\sum w_j C_j := \sum_{j=1}^n w_j \cdot C_j$, qui est la **version pondérée** de $\sum C_j$. D'autres objectifs sont définis de la même manière à partir des retards et des pénalités unitaires, comme $\sum U_j$, $\sum w_j U_j$, $\sum T_j$, $\sum w_j T_j$. Parfois, la fonction objectif est définie de manière plus abstraite, par exemple $\sum f(C_j) := \sum_{j=1}^n f(C_j)$, où l'on suppose juste que f est une fonction de \mathbb{N} dans \mathbb{N} . La version pondérée de $\sum f(C_j)$ est $\sum w_j \cdot f(C_j)$. Plus généralement, on peut associer à chaque tâche une fonction particulière f_j comme dans $\sum f_j(C_j) := \sum_{j=1}^n f_j(C_j)$. Notons que dans ce cas, la version pondérée n'est pas plus générale. L'optimisation de la fonction $L_{max} := \max_{j=1}^n (C_j - d_j)$ se réduit à celle des fonctions $\sum U_j$ et $\sum T_j$ car la valeur optimale de L_{max} est égale au minimum des $t \in \mathbb{Z}$ tels qu'il existe un ordonnancement avec aucune tâche en retard après avoir appliqué la transformation $d_j := d_j + t$.

La Figure 1.10 représente les fonctions que nous utilisons dans ce mémoire. Les Figures 1.10 et 5.3 page 221 montrent que ces fonctions objectifs sont étudiées en ordonnancement chromatique avec avec une terminologie différente de celle utilisée en ordonnancement classique.

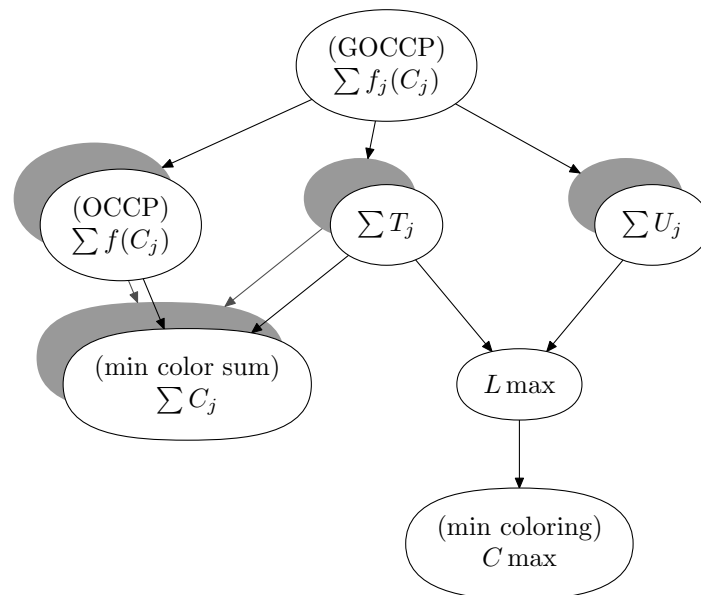


Figure 1.10 – Fonctions objectifs et réductions élémentaires pour l'ordonnancement chromatique. Les ombres correspondent aux versions pondérées (w_j). Entre parenthèses, le nom en ordonnancement chromatique du problème $B1 \mid p_S = 1 \mid$

Chapitre 2

Caractérisations de structures combinatoires

Une classe de graphes admet en général plusieurs types de caractérisations. Nous proposons une classification des types de caractérisations avant d'exposer quelques caractérisations des classes de graphes avec lesquelles nous travaillons dans la deuxième partie de ce mémoire. Ce kaléidoscope de caractérisations fait partie de la boîte à outils élémentaire pour aborder des problèmes d'ordonnement chromatique : chaque type de caractérisation offre un angle de vue différent sur une classe de graphes donnée. Savoir jongler avec les caractérisations permet de s'orienter rapidement vers les classes de graphes pertinentes face à une nouvelle problématique. Dans un deuxième temps, cela permet de trouver la méthode appropriée pour résoudre le problème en question et parfois d'exprimer de nouvelles caractérisations. Nous donnons ensuite quelques briques élémentaires qui pourraient servir de base à la construction de classifications plus larges comme des caractérisations de classes d'hypergraphes, voire de classes de fonctions d'ensemble. Dans cette direction, nous décrivons quelques propriétés des hypergraphes héréditaires. Cela permet de mettre en relief des similitudes liant les systèmes d'indépendance associés aux graphes et aux matroïdes, de donner des formules min-max communes pour certains problèmes d'optimisation et d'élargir la voie vers des généralisations de structures dans lesquelles ces formules sont valides.

Mots clés : Classes de graphes parfaits, caractérisations, inclusions. Graphes {parfaits, triangulés, de Meyniel, sans- P_4 , d'intervalles, LGBIP}.

Autres objets utilisés : Dualités pour les hypergraphes héréditaires et les clutters. Fonctions d'ensembles valeur-polymatroïdales.

Menu du jour

2.1	Plusieurs caractérisations dans un même but	88
2.1.1	Petite philosophie à l’usage des (non-)spécialistes	89
2.1.2	Graphe d’intersection d’une famille d’ensembles	92
2.1.3	Sous-structures exclues	93
2.1.4	Décomposition en graphes plus simples	97
2.1.5	Orientation des arêtes	98
2.1.6	Ordre, graphe de comparabilité et diagramme de Hasse	99
2.1.7	Propriété polyédrale et formule min-max	99
2.1.8	Optimalité d’un algorithme	100
2.1.9	Propriété d’une fonction d’ensemble associée	100
2.2	Classes de graphes	101
2.2.1	$\chi = \omega$ et perfection	101
2.2.2	Les classiques : triangulés, intervalles et LGBIP	103
2.2.3	Graphes sans- P_4	108
2.2.4	Améliorer la perfection	110
2.2.5	Classes de graphes pas forcément parfaits	111
2.3	Classes d’hypergraphes héréditaires	113
2.3.1	De l’intérêt de la généralisation aux hypergraphes	113
2.3.2	Couplage, couverture et coloration	114
2.3.3	Dualités et coloration	115
2.3.4	Mineurs de clutters et d’hypergraphes héréditaires	116
2.3.5	Matroïdes	117
2.3.6	Graphes, conformalité et dualité anti-bloquante	118
2.4	Classes de fonctions d’ensemble	119
2.4.1	Fonctions valeur-polymatroïdales	119
2.4.2	Une théorie générale à venir	122

2.1 Plusieurs caractérisations dans un même but

Dans cette section nous décrivons des “types de caractérisations” qui s’appliquent à des classes importantes de graphes. En général, une classe de graphe admet plusieurs caractérisations de différents types. Ces caractérisations (que l’on utilise ensuite comme définitions) offrent des angles de vues complémentaires sur les graphes contenus dans la classe \mathcal{G} considérée. Les “graphes d’intersection” et “orientation des arêtes” (Sections 2.1.2, 2.1.5 et 2.1.6) donnent des raisons structurelles pour lesquelles “les graphes du monde réel” G appartiennent spontanément à \mathcal{G} . Les “structures exclues” (Section 2.1.3) permettent de prouver facilement qu’un graphe n’est

pas dans la classe \mathcal{G} (certificat co-NP) et aide grandement à comparer les classes de graphes entre elles pour la relation d’inclusion. Les “propriétés polyédrales” ou la “validité d’un algorithme” (Sections 2.1.7 et 2.1.8) permettent de caractériser l’efficacité d’un paradigme d’optimisation. “La décomposition en graphes plus simples” (Section 2.1.4) donne souvent un certificat NP d’appartenance à la classe. Quand il est véritablement accompli, ce type de caractérisation semble jouir implicitement des qualités des autres caractérisations, qu’il aide à déduire.

2.1.1 Petite philosophie à l’usage des (non-)spécialistes

Les publications étudiant les problèmes d’optimisation combinatoire sous l’angle de vue des classes de graphes ne manquent pas (voir les références dans [29]). Ce type d’analyse peut apparaître au néophyte comme digne de peu d’intérêt (d’un point de vue recherche opérationnelle). Nous argumentons au contraire (Section 2.1.1.1), que l’étude des structures combinatoires en général et des classes de graphes en particulier est un axe de recherche majeur du point de vue de l’optimisation combinatoire appliquée. Nous mentionnons ensuite un paradigme élémentaire, mais dont la puissance semble sous-exploitée, qui permet de mettre à jour des liens intimes entre la combinatoire polyédrale et les opérations utilisées en théorie des graphes (Section 2.1.1.2).

2.1.1.1 L’importance de l’étude des classes de graphes

Commençons par un constat trivial : un programme informatique résout toujours un problème particulier. Plus précisément, un programme est une sorte de traducteur entre deux langages : celui des entrées et celui des sorties. Bien sûr, il existe des programmes *écrits dans le but de* résoudre des problèmes relativement généraux. Mais si nos fantasmes d’intelligence artificielle nous incitent à concevoir de tels programmes, la théorie de la complexité et l’expérience ont plutôt tendance à nous persuader que la simplicité des problèmes que l’on considère conditionne en grande partie l’efficacité des programmes que l’on sera capable de développer pour les résoudre¹.

Or l’apparente simplicité du concept de graphe est trompeuse : tel la tour de Babel, le concept de graphe illustre l’orgueil du mathématicien qui croit pouvoir saisir la richesse du monde avec un concept épuré. Mais la complexité algorithmique châtie ceux qui ignorent la richesse des mondes cachés derrière ce cristal sans âme qu’est le concept de graphe. En effet, les graphes généraux sont d’une généralité fatale à la mise en œuvre d’algorithmes théoriquement efficaces : la plupart des problèmes de bases, comme trouver un stable de cardinalité maximum ou une coloration

¹Dans un autre ordre d’idées, les traducteurs automatiques généraux pour traduire correctement les langues naturelles ne semblent pas près de voir le jour...

minimum ne sont, en un sens, même pas approximables sur les graphes généraux [8, 51]. Sur ce constat, on peut penser que c'est le problème de la coloration qui est difficile. La théorie des classes de graphes suggère toutefois un autre point de vue.

Le qualificatif “général”, associé au concept de graphe peut paraître surprenant. On le comprend par la pratique de la modélisation : lorsque l'on traduit un problème appliqué en termes de graphes, la nature du problème impose très souvent une structure particulière aux graphes qui interviennent dans les instances de cette modélisation. La mise-à-jour de propriétés que satisfont ces graphes est donc une condition sine qua none, préalable au développement d'un algorithme théoriquement efficace (pour peu que la modélisation considérée soit une généralisation du problème de coloration minimum).

Le premier type de caractérisations que nous présentons (Section 2.1.2) donne l'argument principal montrant que la modélisation en termes de graphes généraux n'est souvent pas la manière la plus naturelle de modéliser un problème. Ce type de caractérisations nous indique plutôt que les modélisations en termes de graphes sont souvent déjà le résultat d'une modélisation, tellement simple et tellement courante que l'on en a même pas forcément conscience (ce qui est bien dommage).

Le hic, c'est qu'il est difficile de mettre à jour des spécificités “utiles” que satisfont les graphes utilisés lors de la modélisation des problèmes appliqués. Rendre ce processus de mise-à-jour plus efficace est, à mon avis, l'une des raisons d'être majeures de la transdisciplinarité appliquée-fondamentale autour du thème des structures combinatoires. En effet, les problèmes appliqués peuvent contribuer à la pertinence de la connaissance académique par la mise-à-jour de structures récurrentes dans les problèmes étudiés. Ces structures peuvent alors orienter les recherches théoriques vers la mise au point de méthodes de résolution adaptées aux applications. De nombreux travaux étudient les problèmes sur des classes de graphes classiques. Malheureusement, peu de travaux étudient la pertinence applicative des structures combinatoires considérées. Pire encore, rares semblent être les nouvelles structures combinatoires définies pour capturer la particularité des applications considérées.

Les structures que nous décrivons dans ce chapitre sont principalement des cas d'écoles [129, 29, 88]. Il est toutefois intrigant que, pour de nombreuses applications, les structures mises en jeu puissent être décrites en combinant l'expressivité de ces cas d'école. La coloration bornée par exemple (Chapitre 3), peut être vue comme un problème de partition en indépendants communs d'un graphe (parfait dans de nombreuses applications) et d'un matroïde (uniforme). D'autres problèmes de partition font naturellement apparaître les indépendants communs de plusieurs graphes d'intervalles [76]. Malheureusement, trop peu de travaux théoriques étudient en profondeur les questions algorithmiques et structurelles propres à ces objets hybrides. Ceux-ci semblent pourtant avoir une grande *pertinence applicative*, que nous assimilons informellement au rapport

pertinence applicative := expressivité/complexité algorithmique

Ici, la complexité algorithmique est propre au modèle abstrait (avec lequel on *tente de décrire* l'application que l'on a en tête). Selon la théorie de la complexité algorithmique, la complexité d'un modèle est un invariant sur lequel nous n'avons que peu de prise. D'un autre côté, l'*expressivité*² d'un *modèle* concerne la nature et la précision avec lesquels les paramètres interviennent dans la spécification de ce modèle. On peut donc espérer améliorer la pertinence applicative en choisissant (ou en créant) des modèles qui expriment bien les caractéristiques de l'application visée, mais qui ne laissent pas la place aux paramètres superflus. La pertinence applicative des problèmes sur les graphes peut donc être améliorée en mettant à jour les propriétés des graphes qui interviennent réellement dans le modèle.

2.1.1.2 Opérations sur les graphes et combinatoire polyédrale

Considérons la formulation (polyédrale) du problème de la clique de poids maximum avec variables sur les sommets. Plus précisément, considérons une famille de coupes pour ce problème (par exemple, les inégalités de cliques et/ou les inégalités de trous impairs et/ou ...).

Remarque 2.1 *La classe des graphes pour lesquels une telle formulation polyédrale donne un polyèdre entier est fermée par passage au sous-graphe induit.*

Preuve L'intégralité du polyèdre signifie que la famille de coupes considérée donne une borne inférieure serrée pour toute fonction objectif $w : V \rightarrow \mathbb{R}$. Or effacer le sommet v correspond à se restreindre aux poids w tels que $w(v) = 0$. \square

J'ai longtemps considéré la remarque 2.1 comme triviale. Or, après avoir réfléchi aux graphes GTSP-équilibrés, j'ai eu la chance de lire le papier de Fonlupt et Naddef [77], qui mentionne le rôle des mineurs de graphes dans la relaxation graphique du TSP. Aujourd'hui, il me paraît surprenant que ni le mot ni même le concept de mineur ne soit mentionné dans plusieurs articles qui traitent de la relaxation graphique du TSP [94, 121], alors que la notion intervient de manière cruciale dans les énoncés qui y sont proposés. Mais je dois avouer qu'avant de lire [77], je n'en avais pas conscience non plus.

C'est dans ce contexte que la remarque 2.1 m'est apparue comme l'archétype d'un lien systématique entre les opérations sur les graphes et la combinatoire polyédrale³. Ce lien, ou plutôt, ce paradigme est mentionné dans la Section 2.1.3 et au cours de la Section 2.2, relativement aux opérations qui "écrasent" les graphes.

²C'est un néologisme.

³Bien que ce lien soit trivial au niveau logique, il semble ne pas l'être au niveau culturel.

Une des idées issues de l'utilisation de ce paradigme est présentée dans la Section 2.2.5.3 : la pertinence de la contraction $Y-\Delta$ dans le cadre de la relaxation graphique du TSP. L'agréable surprise et la grande conclusion émanant de ce paradigme est toutefois que de nombreuses classes de graphes, bien connues pour être stables par passage à l'induit, le sont aussi pour d'autres opérations susceptibles de traduire des formulations polyédrales avec variables sur les sommets ou les arêtes. Ce fait est d'autant plus réjouissant qu'il s'applique aux classes fondamentales pour les applications comme les intervalles, LGBIP, comparabilité et triangulés. L'approche polyédrale serait-elle efficace dans ces classes de graphes pour d'autres problèmes que ceux issus de la théorie des graphes parfaits ?

Une étude ultérieure serait nécessaire pour pouvoir interpréter le lien entre la combinatoire polyédrale et les opérations de (dé-)composition, évoquées dans la Section 2.1.4 (et avec plus de détails dans [129], Chapitre 65).

2.1.2 Graphe d'intersection d'une famille d'ensembles



est l'icône avec laquelle nous signalons une telle caractérisation. Soit \mathcal{F} une famille d'ensembles. Le **graphe d'intersection de \mathcal{F}** est le graphe $G_{\cap\mathcal{F}}$ qui a \mathcal{F} comme ensemble de sommets et tel que (F_1, F_2) est une arête si et seulement si $(F_1 \cap F_2) \neq \emptyset$. Le graphe d'intersection contient suffisamment d'information pour décrire les sous-ensembles de \mathcal{F} deux-à-deux disjoints⁴ : ce sont les stables du graphe d'intersection de \mathcal{F} .

Soit \mathcal{F} une famille d'ensembles. L'**hypergraphe d'intersection des arêtes** de \mathcal{F} , est l'hypergraphe $H_{\cap\mathcal{F}}$ qui a \mathcal{F} pour sommets et dont $\{F_1, F_2 \dots F_j\}$ est une arête si et seulement si $(\bigcap_{i=1}^j F_i) \neq \emptyset$. Clairement, $H_{\cap\mathcal{F}}$ est héréditaire pour toute famille \mathcal{F} . Si on peut toujours déduire le graphe d'intersection de \mathcal{F} à partir de son hypergraphe d'intersection (en considérant les hyperarêtes de taille 2), on ne peut pas, en général, déduire l'hypergraphe d'intersection de \mathcal{F} à partir de son graphe d'intersection :

Exemple 2.2 Soit $\mathcal{F} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$ et $\mathcal{F}' = \{\{0, 1\}, \{0, 2\}, \{0, 3\}\}$. Les graphes d'intersection de \mathcal{F} et \mathcal{F}' sont tous deux des triangles, bien que les hypergraphes d'intersections soient différents.


Dans certains cas cependant, le graphe d'intersection contient toute l'information contenue dans l'hypergraphe d'intersection. Une famille \mathcal{F} est **Helly** si pour toute sous-famille \mathcal{E} de \mathcal{F} , telle que $E_1 \cap E_2 \neq \emptyset$ pour tous $E_1, E_2 \in \mathcal{E}$, on a aussi $(\bigcap_{E \in \mathcal{E}} E) \neq \emptyset$. Si l'hypergraphe d'intersection de \mathcal{F} est Helly, alors $\{F_1, F_2 \dots F_t\}$ est une arête de $H_{\cap\mathcal{F}}$ si et seulement si c'est une clique de $G_{\cap\mathcal{F}}$.

⁴aussi appelés couplages.

Si \mathcal{F} n'est pas Helly, il convient d'être vigilant lorsque l'on considère les cliques de G . Une **Helly-clique** de \mathcal{F} est un sous-ensemble $\mathcal{E} \subseteq \mathcal{F}$ qui est contenu dans une arête de l'hypergraphe d'intersection de \mathcal{F} . Une Helly-clique de \mathcal{F} est donc toujours une clique du graphe d'intersection de \mathcal{F} . La famille \mathcal{F} de l'Exemple 2.2 montre que la réciproque est fautive. Notons que par définition, cette réciproque caractérise les familles Helly :

Remarque 2.3 *L'ensemble des Helly-cliques de \mathcal{F} est égal à l'ensemble des cliques de $G_{\cap \mathcal{F}}$ si et seulement si la famille \mathcal{F} est Helly.* \square

2.1.3 Sous-structures exclues

 est l'icône avec laquelle nous signalons une telle caractérisation. Certaines classes de graphes admettent une définition en termes de sous-graphes exclus, relativement à une famille d'opérations à préciser. Dans cette section, nous décrivons certaines de ces opérations sur les graphes qui interviennent fréquemment dans ces caractérisation par exclusions. Nous dessinons aussi quelques graphes particuliers qui interviennent dans plusieurs caractérisations par exclusion.

Notons que la caractérisation d'une classe de graphes \mathcal{G} par sous-graphes exclus ne décrit pas vraiment les graphes de la classe \mathcal{G} mais aide plutôt à maîtriser les graphes qu'elle ne contient pas. Cette interprétation philosophique peut en partie être formalisée dans le langage de la complexité : une caractérisation par graphes exclus nous guide souvent vers une caractérisation co-NP du problème d'appartenance à la classe \mathcal{G} . Une caractérisation par graphes exclus est particulièrement utile pour prouver un énoncé du type : "Il est nécessaire d'être dans \mathcal{G} pour avoir une propriété p donnée". En effet, prouver un tel énoncé est équivalent à vérifier les deux faits suivants :

- i) La propriété p est préservée par les opérations utilisées dans la caractérisation de \mathcal{G} considérée.
- ii) Aucune des obstructions de l'appartenance à \mathcal{G} ne satisfait la propriété p .

Dans les sous-sections suivantes, nous considérons un graphe $G = (V, E)$ et $\mathcal{F} = \{F_1, F_2, \dots\}$ une liste (éventuellement infinie) de graphes.

2.1.3.1 Suppression de sommets et sous-graphes induits

Pour $v \in V$, le sous-graphe $G - v$ est le graphe qui a $V - v$ comme sommets et dont deux sommets de $V - v$ sont reliés dans $G - v$ si et seulement s'il sont reliés dans G . Pour $U \subseteq V$, le graphe $G - U$ est obtenu en **supprimant**⁵ successivement tous les sommets de U et $G[U]$ est le **graphe induit** par U , obtenu en enlevant

⁵On dit aussi **effacer** ou *deleting*.

tous les sommets de $V \setminus U$. H est un sous-graphe induit de G s'il existe $U \subseteq V$ tel que $H = G[U]$.

G est *sans- \mathcal{F}* , si G ne contient aucun des graphes F_i comme sous-graphes induits.

2.1.3.2 Suppression et contractions d'arêtes

Nous définissons ici les opérations pour les graphes simples seulement. Pour $e \in E$, *supprimer*⁶ l'arête e donne le graphe $G \setminus e$ qui a V comme sommets et $E - e$ comme arêtes. *Contracter* l'arête $e = uv$ donne le graphe G/e obtenu en enlevant les sommets u et v puis en ajoutant un nouveau sommet adjacent à $N(u) \cup N(v)$. Le graphe G_1 est un *mineur* de G s'il peut être obtenu par une suite de suppressions et de contractions d'arêtes ainsi que de suppressions de sommets.

G est *sans mineurs- \mathcal{F}* , si G ne contient aucun des graphes F_i comme mineur. Les graphes définis par des propriétés topologiques, comme la plongeabilité sur une surface, sont souvent stables par passage au mineur. Par ailleurs, les graphes définis par des propriétés valables pour toutes pondérations des arêtes sont souvent valables par passage au mineur. Par exemple, si la pondération correspond à des longueurs, la contraction correspond à se restreindre à la longueur 0 et la suppression correspond à se restreindre à la longueur $+\infty$. La stabilité par passage au mineur survient en particulier dans les propriétés relatives à la relaxation graphique du TSP qui sont valables pour toutes les longueurs.

La *contraction fantôme*⁷ de l'arête $e = uv$, notée $G|e$ est équivalente à la contraction de e , sauf que le nouveau sommet est adjacent à $N(u) \cap N(v)$. Pour $e \in E(G)$, on a aussi

$$G|e := \overline{(G + e)}/e$$

Si une propriété est stable par passage au mineur, elle l'est aussi par contraction fantôme. Mais certaines classes de graphes sont stables par contraction et par contraction fantôme sans être stables par suppression d'arêtes. Les graphes d'intervalles sont un exemple.

⁶On dit aussi *effacer* ou *deleting*.

⁷C'est un néologisme.

2.1.3.3 Contraction de sommets et opération $\Delta - Y$

Soit $v \in V$, la **contraction voisinage-clique**⁸ de v donne le graphe G/v qui a $V - v$ comme sommets et dans lequel l'ensemble des sommets voisins de v induisent une clique. Autrement dit,

$$E(G/v) := E(G - v) \cup \{wx \mid w, x \in N_G(v)\}$$

Le cas spécial de la contraction voisinage-clique dans lequel $v + N(v)$ induit le graphe $K_{1,3}$ est la **contraction** $Y - \Delta$.

Il est trivial de voir que les propriétés de plongeabilité sur les surfaces sont stables par contraction $Y - \Delta$. La stabilité par contraction $Y - \Delta$ survient aussi dans les propriétés relatives à la relaxation graphique du TSP (Section 2.2.5.3).

La contraction $Y - \Delta$ est une opération qui “préserve l'information d'un graphe” : elle peut être inversée de la manière suivante. Si $G[\{x, y, z\}]$ est un triangle, l'opération $\Delta - Y$ consiste à enlever les arêtes de ce triangle et à ajouter un nouveau sommet voisin seulement de x, y et z . De nombreuses classes de graphes stables par contraction $Y - \Delta$ ne sont toutefois pas stables par l'opération $\Delta - Y$ comme les graphes planaires, planaires externes, GTSP-sous-modulaires et GTSP-équilibrés, discutés dans la Section 2.2.5.

2.1.3.4 Sous-graphes isométriques exclus

$U \subseteq V$ induit un sous-graphe isométrique de G si pour tous $u, v \in U$, la distance (*i.e.* le plus court chemin, en nombre d'arêtes) entre u et v dans $G[U]$ est la même que dans G .

G est **isométriquement sans- \mathcal{F}** , si G ne contient aucun des graphes F_i comme sous-graphe isométrique.

2.1.3.5 Quelques graphes spéciaux

Le **cycle** C_n est un graphe avec $[1..n]$ comme sommets tel que i, j forme une arête si et seulement si $|i - j| = 1$ ou $|i - j| = n - 1$. Un cycle est **pair** (respectivement **impair**) si n est pair (respectivement impair). Un **trou** est un cycle C_n induit avec $n \geq 4$. Un **chemin** P_n est un graphe avec $[1..n]$ comme sommets tel que i, j

⁸Cette opération ne semble pas avoir été étudiée auparavant. De même que le passage à l'induit, elle a une interprétation naturelle pour les formulations polyédrales avec variables sur les sommets du graphe. En particulier, cette opération intervient naturellement dans la formulation polyédrale d'un problème qui a une expressivité intermédiaire entre l'Hamiltonicité et la relaxation graphique du TSP. Cet axe de recherche n'est pas discuté plus au delà dans ce mémoire.

forme une arête si et seulement si $|i - j| = 1$. Malheureusement, contrairement aux cycles, le nombre d'arêtes et de sommets d'un chemin sont de parité différentes, ce qui prête souvent à confusion lorsque l'on parle de parité. Nous définissons la parité du chemin P_n comme la parité de son nombre n de sommets. Un **graphe complet** est un graphe avec n sommets toutes les arêtes possibles.

Si C est à la fois un cycle et un graphe partiel de G , les arêtes entre les sommets de C qui ne sont pas des arêtes de C sont des **cordes**. Une corde est **courte** si elle joint deux sommets à distance 2 d'un cycle. Deux cordes (i, j) et (k, l) d'un cycle C sont **croisées** si $\{k, l\} \cap \{i, j\} = \emptyset$ et k et l ne sont pas sur le même chemin de C joignant i à j . Les cordes (i, j) et (k, l) sont **non-croisées** sinon. Une **maison** est un cycle impair avec une corde courte.

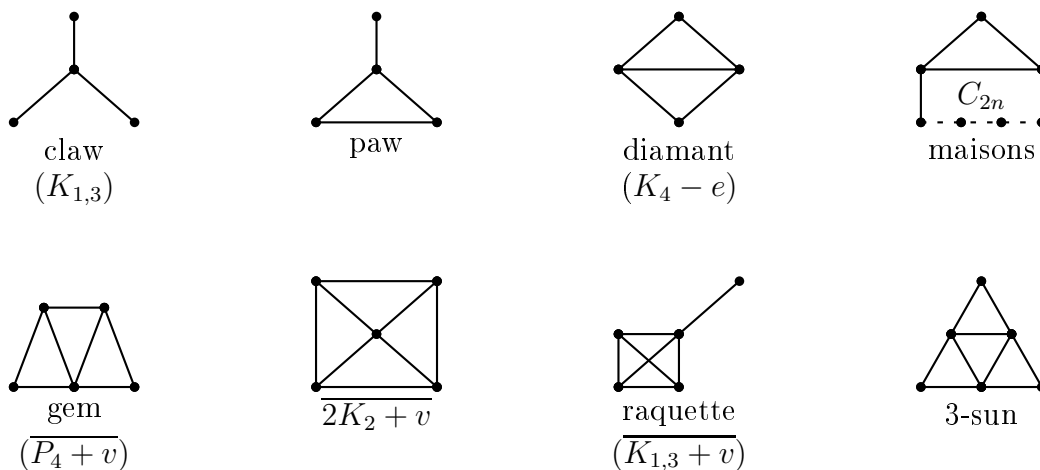



Figure 2.1 – Quelques graphes qui interviennent souvent dans les caractérisations par graphes induits exclus.

2.1.4 Décomposition en graphes plus simples

 est l'icône avec laquelle nous signalons une telle caractérisation. Si une caractérisation d'une classe \mathcal{C} par sous-graphes exclus est attrayante, elle parle toutefois plus de ce que *n'est pas* la famille \mathcal{C} que de ce qu'elle contient. Avec le recul, la popularité de la conjecture forte des graphes parfaits semble plus due à la simplicité de l'énoncé qu'à la pertinence scientifique. Bien que le paradigme de caractérisation par structures exclues soit très utile, comme nous l'avons vu dans la section précédente, une preuve de cette conjecture passe pour l'instant par un théorème plus profond qui décrit les graphes parfaits en termes de décomposition. En l'occurrence, tout porte à croire que le véritable théorème de décomposition des graphes parfaits n'a même pas encore été énoncé.

Le paradigme de décomposition est probablement le plus profond et le plus mystérieux des paradigmes de caractérisation de classe de graphes que nous présentons ici. Tout d'abord, les “bonnes décompositions” sont difficiles à définir et même à imaginer. Ensuite, prouver qu'elles s'appliquent correctement est souvent long et fastidieux. Finalement il reste en général ouvert de savoir si on peut transformer ces décompositions en opérations algébriques ; il n'est même pas toujours trivial de les raffiner en décompositions canoniques. Finalement, lorsqu'il est complètement abouti, un théorème de décomposition donne souvent aux autres caractérisations un goût de simple conséquence. De plus, les bonnes décompositions guident généralement vers des algorithmes d'optimisation et de reconnaissance efficaces, ou permettent au moins de prouver l'optimalité de tels algorithmes. La théorie des graphes parfaits est une mine d'or pour l'exploration de ce paradigme, qui s'applique à de nombreuses sous-classes des graphes parfaits et qui a été la motivation originelle de la définition de nombreuses opérations. Nous n'avons pas pu approfondir assez cette voie pour tirer pleinement parti de son potentiel. Aussi, nous décrivons seulement les décompositions qui nous servent directement. Nous référons à [129], Chapitre 65 (pour les graphes parfaits) et [29] (pour les sous-classes de graphes parfaits) pour un traitement plus complet de ce paradigme.

2.1.4.1 Vrai-faux jumeau et (co)-réplication

Dans un graphe $G = (V, E)$, **dupliquer** un sommet v consiste à ajouter un sommet v' à G , dont l'adjacence est la même que celle de v et tel que v et v' sont adjacents. De manière équivalente, deux sommets voisins ayant le même voisinage sont des **vrais jumeaux**. **Répliquer** un sommet v avec un facteur r signifie le dupliquer $r - 1$ fois si $r \geq 1$ et le retirer du graphe si $r = 0$. Deux sommets non-voisins ayant le même voisinage sont des **faux jumeaux**. v et w sont faux jumeaux dans G si et seulement s'ils sont vrais jumeaux dans \overline{G} . L'adjonction de faux jumeaux est donc la **co-duplication** (**co-réplication**).

2.1.4.2 Union disjointe, join complet, sommets universels et isolés

L'*union disjointe* $G_1 \cup G_2$ de deux graphes G_1 et G_2 tels que $V(G_1) \cap V(G_2) = \emptyset$ est le graphe qui a $V(G_1) \cup V(G_2)$ comme sommets et $E(G_1) \cup E(G_2)$ comme arêtes. Le *join complet* $G_1 \oplus G_2$ est l'opération complémentaire de l'union disjointe :

$$(2.1) \quad G_1 \oplus G_2 := \overline{\overline{G_1} \cup \overline{G_2}}$$

Autrement dit, l'adjacence induite sur G_1 et G_2 est préservée et tous les sommets de G_1 sont adjacents à tous les sommets de G_2 .

Un sommet v est *universel* s'il est voisin de tous les autres sommets. Autrement dit, v est universel dans G si G est obtenu par join complet entre $G[v]$ et $G[V - v]$, ou encore si v est isolé dans \overline{G} .

2.1.4.3 Sommet simplicial

Un sommet v est *simplicial* si son voisinage $N(v)$ induit une clique.

Un ordre (v_1, \dots, v_n) sur V est un *ordre de décomposition simplicial* ou *ordre simplicial* si pour tout i , v_i est simplicial dans $G[v_i, \dots, v_n]$.

2.1.4.4 Ensemble d'articulation

Un *ensemble d'articulation* dans un graphe G est un sous-ensemble U de sommets tels que $G[V \setminus U]$ ait strictement plus de composantes connexes que G . Si U est un ensemble d'articulation d'un graphe connexe G , soit $\{V_i\}_{i=1 \dots t}$ les composantes de $G[V \setminus U]$ (L'ensemble des V_i avec U forme donc une partition de V). En général, ce ne sont pas les propriétés des $G[V_i]$ qui jouent un rôle, mais les propriétés des $G[V_i \cup U]$.

U est une *clique d'articulation* (resp. *sommet d'articulation*) si U est un ensemble d'articulation et $G[U]$ induit une clique (resp. $|U| = 1$).

Un graphe peut évidemment avoir plusieurs cliques d'articulation. Gavril [81] a décrit la structure d'arbre de l'ensemble des cliques d'articulation d'un graphe G .

2.1.5 Orientation des arêtes



est l'icône avec laquelle nous signalons une telle caractérisation. Plusieurs classes de graphes peuvent être caractérisées en termes de digraphes. Il s'agit de considérer une classe \mathcal{D} de digraphes satisfaisant une ou plusieurs propriétés, comme

être sans circuit ou être transitif. A partir de \mathcal{D} , on définit la classe de graphes \mathcal{G} obtenus en effaçant l'orientation des digraphes de \mathcal{D} . De manière équivalente, on caractérise $G \in \mathcal{G}$ par l'existence d'une orientation \vec{G} de G telle que $\vec{G} \in \mathcal{D}$.

Ce genre d'approche donne une caractérisation des graphes de comparabilité (à partir des digraphes transitifs), mais aussi des graphes triangulés, des graphes parfaits (par les noyaux [129]) ainsi que de nombreuses sous-classes de graphes de comparabilité, ce que nous détaillons dans la prochaine section.

2.1.6 Ordre, graphe de comparabilité et diagramme de Hasse



est l'icône avec laquelle nous signalons une telle caractérisation. Plusieurs sous-classes importantes de graphes sont des sous-classes de graphes de comparabilité. Ces classes interviennent dans le cadre des problèmes de colorations mais aussi dans d'autres domaines de l'optimisation combinatoire (optimisation des fonctions sous-modulaires, ordonnancement avec contraintes de précédence). Comme il existe deux manières standards de représenter un poset : son diagramme de Hasse et son digraphe de comparabilité, il est important de pouvoir facilement faire la correspondance entre les deux. Cela permet d'une part de faciliter la diffusion des résultats entre disciplines, mais aussi de choisir la représentation appropriée en vue de résoudre un problème donné. Quelques résultats obtenus selon ce paradigme sont :

- La correspondance entre les graphes sans- P_4 et des digraphes séries-parallèles.
- L'équivalence des graphes trivialement parfaits, des familles laminaires et des ordres dont le diagramme de Hasse et une forêt orientée (entrante ou sortante).
- La correspondance entre les graphes sans- P_3 et les ordres dont le diagramme de Hasse est une union de chaîne.
- La caractérisation des ordres d'intervalles comme ceux pour lesquels une grande famille de problèmes de partition en stables et d'ordonnancement avec contraintes de précédence sont équivalents.

2.1.7 Propriété polyédrale et formule min-max



est l'icône avec laquelle nous signalons une telle caractérisation. Considérons un problème d'optimisation combinatoire et un PLNE générique représentant ce problème. On définit la classe des instances pour lesquelles le PLNE a une propriété d'intégralité donnée, par exemple, matrice totalement unimodulaire, polyèdre entier, mais aussi TDI, IDP, MIRUP ou encore rang de Chvátal = 1.

Ce genre d'approche est très connu pour donner une caractérisation des graphes parfaits (à partir de la condition $\chi(G) = \omega(G)$) ainsi que pour les graphes unimo-

dulaires, bipartis ou encore les clutters idéaux. Il est moins connu que les graphes de parité, de Meyniel et triangulés puissent aussi être caractérisés par de telles propriétés.

2.1.8 Optimalité d'un algorithme



est l'icône avec laquelle nous signalons une telle caractérisation. Considérons un algorithme A , qui ne résout pas nécessairement le problème pour lequel il a été conçu pour toutes les instances. On définit l'ensemble des instances pour lesquelles l'algorithme A donne un résultat correct. Ce genre d'approche est connu pour permettre de caractériser, au sein des hypergraphes héréditaires, ceux qui définissent les indépendants d'un matroïde. Mais les graphes sans- P_4 peuvent aussi être caractérisés par l'optimalité d'un algorithme glouton pour le problème de la coloration minimum. Des résultats partiels et récents, que nous n'avons pas eu le temps d'inclure dans ce mémoire montrent que l'optimalité des algorithmes classiques pour la coloration caractérisent quelques classes standards de graphes parfaits, pour peu que l'on demande l'optimalité de l'algorithme pour le problème de la coloration pondérée⁹.

2.1.9 Propriété d'une fonction d'ensemble associée

Considérons une fonction r définie sur l'ensemble des graphes, par exemple $r(G) = \alpha(G)$. On associe alors une fonction d'ensemble (sur V) à tout graphe $G = (V, E)$, par exemple pour $U \subseteq V$, on pose $r(U) := r(G[U])$. On peut bien sûr généraliser cette approche aux hypergraphes, ou en définissant une fonction d'ensemble sur l'ensemble des arêtes. Les indépendants des matroïdes sont bien connus pour être caractérisés par la sous-modularité de la fonction de rang. Le développement de la théorie des jeux coopératifs combinatoires a donné de nouvelles caractérisations de ce genre. Par exemple, un graphe est parfait si et seulement si la fonction d'ensemble $\chi(G[U])$ définie pour tout $U \subseteq V$ est totalement-équilibrée.

⁹De telles caractérisations pour le problème de la coloration minimum concernent par exemple les paires (algorithmes, classes de graphes) suivantes : (LexColor, Meyniel), (SimpleLex, Parité) ou encore (MCColor, Meyniel \cap faiblement triangulé). Ces trois caractérisations ne sont pas discutées plus au delà dans ce mémoire.

2.2 Classes de graphes

Dans cette section, nous mentionnons les types de caractérisations discutées précédemment qui s'appliquent pour quelques classes de graphes. Certaines caractérisations nouvelles sont discutées plus en détails. Les relations d'inclusions pour ces classes de graphes sont résumées dans la Figure A.2 page 257.

2.2.1 $\chi = \omega$ et perfection





Dans cette section, nous discutons les graphes pour lesquels $\chi = \omega$. Nous commençons par les graphes parfaits, avant de donner des raisons fortes, selon la théorie de la complexité algorithmique, pour lesquelles la condition $\chi = \omega$ ne définit pas une classe de graphes intéressante.

2.2.1.1 Graphes parfaits

La perfection est liée à l'étude des propriétés d'intégralité du système suivant associé à tout graphe G .

- (2.2) (i) $x_v \geq 0$ pour tout $v \in V$
(ii) $x(C) \leq 1$ pour toute clique maximale C de G

Théorème 2.4 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i)  *Le système (2.2) est TDI.*
- ii)  *Le système (2.2) définit un polyèdre entier.*
- iii)  $\chi(G[S]) = \omega(G[S])$ pour tout $S \subseteq V$.
- iv)  G n'a ni trou ni antitrou impair.

Un graphe est **parfait** s'il vérifie les conditions du Théorème 2.4. Une des forces de la théorie des graphes parfaits est donc que plusieurs propriétés d'intégralité du (2.2) définissent la même classe de graphes (cela n'est en général pas le cas en combinatoire polyédrale [129]).

2.2.1.2 Graphes pour lesquels $\chi = \omega$

Les graphes pour lesquels $\chi = \omega$ méritent l'attention de quelques minutes. On peut en déterminer le nombre chromatique χ ainsi que le nombre de clique ω en temps polynômial [129]. Cependant, trouver les objets optimaux correspondants est NP-dur. Une telle affirmation peut-être prouvée proprement à l'aide du cadre des problèmes de promesse.

Proposition 2.5 *Les problèmes suivants sont NP-durs :*

i) *Données : G . Résultat : Est-ce que $\chi(G) = \omega(G) = 3$?*

ii) *Données : G . Promesse : $\chi(G) = \omega(G) = 3$. Résultat : une 3-coloration de G .*

iii) *Données : (G, k) . Promesse : $\chi(G) = \omega(G)$. Résultat : une k -clique de G .*

Preuve i) Rappelons que décider si un graphe est 3-colorable est un problème NP-complet. Hors un graphe G est 3-colorable si et seulement si $\omega(G) \leq 3$ et si l'union disjointe de G avec un triangle vérifie $\chi(G) = \omega(G)$.

ii) Réduisons (au sens de Turing) 3-colorabilité à trouver une 3-coloration sous la promesse $\chi = \omega = 3$ ("non" et "oui" réfèrent à la 3-colorabilité de G). Etant donné un graphe G , si $\omega(G) \geq 4$ répondre "non". Si $\omega(G) \leq 3$, considérer G' , l'union disjointe de G avec un triangle. Appliquons un algorithme A à G' qui trouve une 3-coloration quand $\chi(G') = \omega(G') = 3$. Si la sortie de A est une 3-coloration répondre "oui". Si la sortie de A n'est pas une 3-coloration, répondre "non".

iii) Réduisons (au sens de Turing) l'existence d'une clique de taille au moins k à trouver une ω -clique sous la promesse $\chi(G) = \omega(G)$.

Etant donné un graphe G et un entier k , construisons le graphe G_k qui a k copies de $V(G)$ comme sommets et tel que (v, w) est une arête dans G_k si v et w n'appartiennent pas à la même copie et si v et w sont des copies de sommets reliés dans G . Appliquons un algorithme B à G_k qui trouve une k clique quand $\chi(G_k) = \omega(G_k)$. Répondre "oui" si et seulement si B renvoie une clique de taille k .

Montrons que la réduction est valide. Observons que G_k a une clique de taille k si et seulement si G en a une. Notons aussi que $\chi(G_k) \leq k$. On a donc pour tout graphe G et tout k , d'une part que $\omega(G_k) \leq \omega(G)$ et d'autre part que $\omega(G_k) \leq \chi(G_k) \leq k$. Si l'on a répondu "oui" on a $\omega(G_k) = \chi(G_k) = k \leq \omega(G)$.

Si l'on a répondu "non", c'est soit parce que $\omega(G_k) = \chi(G_k) < k$, soit que $\omega(G_k) < \chi(G_k) \leq k$. Dans les deux cas on a aussi $\omega(G) < k$. \square

2.2.2 Les classiques : triangulés, intervalles et LGBIP








2.2.2.1 Graphes triangulés

Un graphe est *triangulé*¹⁰ s'il n'a pas de trou.

Proposition 2.6 *La classe des graphes triangulés est stable par réplication, ajout de sommet universel, contraction d'arête, contraction fantôme, contraction voisinage-clique.*




Preuve Aucune de ces opérations ne crée de trous. □

Théorème 2.7 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i)  Tout sous-graphe induit de G a un sommet simplicial.
- ii)  Tout sous-graphe induit de G qui n'est pas une clique a une clique d'articulation.
- iii)  G est le graphe d'intersection de sous-arbres d'un arbre.
- iv)  G admet une orientation A telle que si $vw \in A$ et $vx \in A$ alors $wx \in E$.
- v)  $\bar{\chi}_b(H) = \sigma_b(H)$ pour tout graphe obtenu à partir de G par réplication.
- vi)  G est sans trou (C_k pour $k \geq 4$).
- vii)  G est sans- $\{\text{induit, contracté}^{11}\}$ - $\{C_4\}$.

2.2.2.2 Split graphes

Théorème 2.8 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i)  $V(G)$ peut-être partitionné en un stable et une clique.
- ii)  G est le graphe d'intersection d'une famille de sous-étoiles d'une étoile dont seules les sous-étoiles qui n'induisent pas une feuille peuvent apparaître plus d'une fois dans la famille.
- iii) G et \bar{G} sont triangulés.
- iv)  G est sans- $\{C_4, \bar{C}_4, C_5\}$.



Un graphe vérifiant les conditions du Théorème 2.8 est un *split* graphe.

¹⁰chordal, en anglais.

¹¹Pour la contraction d'arêtes ou pour la contraction voisinage-clique.

2.2.2.3 Graphes de comparabilité et de permutation

Théorème 2.9 Soit $G = (V, E)$ un graphe. Les conditions suivantes sont équivalentes


- i)  Il existe un poset (V, \prec) tel que $a, b \in E$ si et seulement si $a \prec b$ ou $b \prec a$.
- ii)  Il existe une orientation des arêtes de G telle que si $ab, bc \in A$ alors $ac \in A$.

Un graphe est de **comparabilité** s'il vérifie les conditions du Théorème 2.9. La caractérisation par sous-graphes induits exclus des graphes de comparabilité est assez longue et complexe [114]. Il est envisageable que la proposition suivante mène à des caractérisations plus simples.

Proposition 2.10 La classe des graphes de co-comparabilité est stable par contraction d'arêtes et par contraction voisinage-clique. \square




Un graphe est de **permutation** s'il vérifie les conditions du Théorème 2.11.

Théorème 2.11 Soit $G = ([1..n], E)$ un graphe. Les conditions suivantes sont équivalentes

- i)  Il existe une bijection f sur $[1..n]$ telle que $a, b \in E$ si et seulement si $(a - b)(f(a) - f(b)) > 0$.
- ii) G et \bar{G} sont de comparabilité.

2.2.2.4 Graphes bipartis

Théorème 2.12 Soit G un graphe. Les conditions suivantes sont équivalentes

- i)  La matrice d'incidence (arêtes-sommets) de G est totalement unimodulaire.
- ii)  $V(G)$ peut-être partitionné en deux stables ($\iff \chi(G) \leq 2$)
- iii)  G est sans cycle impair induit

Un graphe vérifiant les conditions du Théorème 2.12 est **biparti**.

2.2.2.5 Arbres et forêts

Une **forêt** est un graphe sans cycle. Un **arbre** est une forêt connexe.

Proposition 2.13 *La classe des forêts est stable par mineurs et par co-contraction d'arête.* \square

Remarque 2.14 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i) $\ominus G$ est sans cycles,
- ii) $\ominus G$ est sans-mineur- K_3 ,
- iii) $\ominus G$ est sans-{induit, contraction d'arête}- K_3 .



2.2.2.6 Line-graphes de bipartis

Proposition 2.15 *Les line-graphes de (multi-)graphes bipartis sont stables par contraction fantôme.*

Preuve Si $G = ((A, B), E)$ est une (multi-)graphe biparti, et $(a, b_1), (a, b_2) \in E$, la contraction fantôme dans $L(G)$ est obtenue en créant un nouveau sommet isolé b' dans G , en enlevant les arêtes (a, b_1) et (a, b_2) , en ajoutant l'arête (a, b') , puis en passant au line-graphe. \square

L'ensemble des stables du line-graphe d'un graphe biparti est un cas particulier bien connu de système d'indépendance défini par l'intersection de deux matroïdes : un tel système peut même être décrit par l'intersection de deux matroïdes de partition. Nous observons en fait que la possibilité d'exprimer l'ensemble des stables comme les indépendants commun de deux matroïdes caractérise les line-graphes de multigraphes bipartis :

Théorème 2.16 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i)  G est le line-graphe d'un multigraphe biparti.
- ii)  Il existe deux matroïdes \mathcal{M}_1 et \mathcal{M}_2 sur $V(G)$ dont l'ensemble des indépendants commun est égal à l'ensemble des stables de G .
- iii) $\ominus G$ est sans- $\{C_{2n+5}, K_{1,3}, \overline{P_4 + v}, \overline{2K_2 + v}\}$ [115]
- iv) $\ominus G$ est sans- $\{\text{induit, fantôme}\}$ - $\{C_{2n+5}, K_{1,3}, \overline{2K_2 + v}\}$

Preuve iii) \iff i) est montré dans [115]

i) \implies ii) Si $G = (V, E)$ est le line-graphe d'un multigraphe biparti $G' = ((A, B), V)$, les stables de G sont donnés par l'intersection des deux matroïdes de partition sur V correspondants à l'incidence de A et de B respectivement [129].

iii) \implies iv) Proposition 2.15.

iv) \implies iii) $\overline{P_4 + v}$ donne $K_{1,3}$ par contraction fantôme.

ii) \implies iii)¹² On vérifie que la famille des stables de chacun des graphes $\{C_{2n+5}, K_{1,3}, gem, \overline{2K_2 + v}\}$ ne peut être décrit par l'intersection de deux matroïdes de la manière suivante. Supposons qu'il existe deux tels matroïdes $\mathcal{M}_1 = (V, \mathcal{I}_1)$ et $\mathcal{M}_2 = (V, \mathcal{I}_2)$. Les sous-ensembles de V minimaux dépendants dans le système d'indépendance défini par $\mathcal{M}_1 \cap \mathcal{M}_2$ sont donnés par le clutter $(\mathcal{C}(\mathcal{M}_1) \cup \mathcal{C}(\mathcal{M}_2))^{\min}$ et ce clutter est l'ensemble des arêtes de G . Si les stables de G forment les indépendants communs de \mathcal{M}_1 et \mathcal{M}_2 , il existe un étiquetage des arêtes de G avec trois étiquettes possibles : $\{1\}, \{2\}$ ou $\{1, 2\}$ selon que l'arête est un circuit de \mathcal{M}_1 , de \mathcal{M}_2 ou des deux (les rôles de \mathcal{M}_1 et \mathcal{M}_2 sont symétriques). En général, on tire des conclusions de la structure des P_3 de G : pour chaque P_3 induit, les deux arêtes sont étiquetées chacune avec un numéro différent (et seulement avec un numéro). Pour C_{2n+5} , comme pour $K_{1,3}$ la contradiction est directe. Pour les deux graphes restants, il faut observer de plus que chaque $i \in \{1, 2\}$ appartient à l'étiquette des trois arêtes de chaque triangle, dès que deux arêtes de ce triangle contiennent i dans leur étiquette. \square

2.2.2.7 Graphes d'intervalles

Un *graphe d'intervalle(s)* est le graphe d'intersection d'intervalles de la droite réelle. De même que pour les graphes de comparabilité, la liste des sous-graphes induits exclus n'est pas triviale. D'autres caractérisations plus simples pourraient éventuellement être obtenues grâce à la proposition suivante.

Proposition 2.17 *Les graphes d'intervalles sont stables par contraction d'arêtes, contraction fantôme et contraction voisinage-clique.*

Preuve Pour deux intervalles I et J qui s'intersectent, la contraction consiste à remplacer I et J par leur union et la contraction fantôme par leur intersection. La contraction voisinage-clique de l'intervalle I est obtenue en remplaçant tous les intervalles J qui intersectent I par $I \cup J$ (puis en effaçant I). \square

Un *ordre* \preceq sur V est d'*intervalle(s)* si V est en bijection avec un ensemble d'intervalles de la droite réelle tels que si $[a_v, b_v]$ et $[a_w, b_w]$ représentent v et w respectivement, on a

$$(2.3) \quad v \prec w \iff b_v < a_w$$

Les graphes de comparabilité correspondants aux ordres d'intervalles ne sont donc pas les graphes d'intervalles mais leurs compléments.

Clairement, un ordonnancement par batch qui respecte les contraintes de précedence d'un poset donne une coloration du graphe de comparabilité de ce poset. La réciproque est fautive, comme le montre la Figure 2.2 : une partition en antichaîne d'un poset ne peut pas toujours être ordonnée en batchs qui respectent

¹²La preuve de cette implication repose sur des concepts généraux qui sont exposés dans la Section 2.3. La lecture préalable du Théorème 2.22 est recommandée pour des raisons pédagogiques.

les contraintes de précédence. Cela est cependant possible pour les ordres d'intervalles [113, 22]. Nous observons que cette propriété caractérise les ordres d'intervalles : le poset ayant $\overline{C_4} = 2K_2$ comme graphe de comparabilité est l'unique poset induit exclu pour lequel les colorations ne peuvent pas être ordonnées pour respecter les contraintes de précédence.

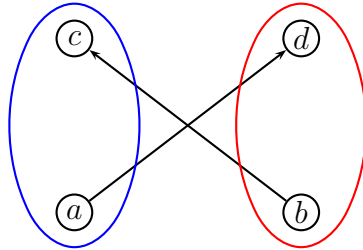







Figure 2.2 – Une partition en antichaînes du poset ayant $\overline{C_4}$ comme graphe de comparabilité. Cette partition ne peut pas être ordonnée de manière à respecter les contraintes de précédences.

Théorème 2.18 Soit G un graphe. Les conditions suivantes sont équivalentes

- i)  [85, 125] G est le graphe d'intersection d'une famille d'intervalles de la droite réelle.
- ii)  [85, 125] G est le complément du graphe de comparabilité d'un poset d'intervalles.
- iii) [85, 125] G est un graphe de co-comparabilité et triangulé.
- iv)  [85, 125] G est un graphe de co-comparabilité sans- C_4 .
- v)  [125] $G = (V, E)$ est le complément du graphe de comparabilité d'un poset (V, \prec) et pour tout $v, w \in V$ on a soit $A(v) \subseteq A(w)$ soit $A(w) \subseteq A(v)$.
- vi)  $G = (V, E)$ est le complément du graphe de comparabilité d'un poset (V, \prec) et pour toute partition en cliques $\{C_1, C_2, \dots, C_k\}$ de G , il existe une permutation π de $[1..k]$ telle que l'ordonnancement par batch $\{C_{\pi(1)}, C_{\pi(2)}, \dots, C_{\pi(k)}\}$ respecte les contraintes de précédence de \prec .

En un sens, les graphes d'intervalle sont donc exactement les graphes dans lesquels les problèmes d'ordonnancement avec contraintes de précédences sont équivalents aux problèmes d'ordonnancement avec contraintes d'incompatibilité. La validité de cette équivalence se concrétise principalement lorsqu'on l'on traite d'un problème d'ordonnancement avec minimisation du *makespan* C_{\max} ¹³. Cette équivalence s'ap-

¹³Sans contraintes supplémentaires qui pourraient briser l'invariance par permutation des batchs, comme des dates de disponibilité r_j par exemple.

plique directement aux résultats des Chapitres 3 et 4. Le Lemme 2.19 est un point d'articulation dans la preuve du Théorème 2.18.

Lemme 2.19 [85, 125] *Un graphe de comparabilité n'a pas d'antitrou \overline{C}_n pour $n \geq 5$.*

Preuve Le cas $n = 5$ étant évident, supposons que $n \geq 6$. Soit $v_1, v_2 \dots v_n$ les sommets de \overline{C}_n , ordonnés en suivant les arêtes de C_n . Supposons que \overline{C}_n soit un graphe de comparabilité. S.p.d.g, on peut orienter l'arête v_1, v_3 dans le sens $\overrightarrow{1, 3}$. On a alors $\overrightarrow{1, i}$ pour tout $i \in [3..n-1]$. On a ensuite $\overrightarrow{i, n-1}$ pour tout $i \in [1..n-3]$. Puis $\overrightarrow{n-3, i}$ pour $i \in \{n-1, n, 1\}$. L'arête v_1, v_{n-3} doit donc être dans les deux sens à la fois, contradiction. \square

Preuve[du Théorème 2.18] $i) \iff ii) \iff iii)$ est bien connu [85, 125, 88, 29].

$iii) \implies iv)$: Un graphe triangulé n'a pas de trou et donc pas de C_4 induit.

$iv) \implies iii)$: D'après le Lemme 2.19, un graphe de co-comparabilité sans- C_4 est un graphe triangulé.

$ii) \implies vi)$: Soit $V = \{I_1, I_2, \dots, I_n\}$ un ensemble d'intervalles de la droite réelle définissant l'ordre d'intervalle (V, \prec) . Soit $\{C_1, C_2, \dots, C_k\}$ une partition en cliques du graphe d'intervalle associé. Pour chaque clique C_i , il existe un point t_i tel que tous les intervalles de C_i contiennent le point t_i . Choisissons un tel point t_i pour chaque $i \in [1..k]$ et supposons (quitte à permuter les C_i) que $t_1 \leq t_2 \leq \dots \leq t_k$. Vérifions que l'ordonnancement par batches $\{C_1, C_2, \dots, C_k\}$ respecte les contraintes de précédence. Si ce n'est pas le cas, il existe deux intervalles $I \in C_i$ et $J \in C_j$ avec $1 \leq i < j \leq k$ tels que la fin de l'intervalle J précède strictement le début de I . Cela est impossible car $t_i \leq t_j$.

$vi) \implies iv)$: Si (V, \prec) n'est pas un ordre d'intervalles, le graphe G de comparabilité de (V, \prec) contient un \overline{C}_4 . Considérons la partition en cliques de \overline{G} donnée par la Figure 2.2 (les autres sommets étant placés dans des cliques de taille 1). Cette partition ne peut être ordonnée de manière à respecter les contraintes de précédence de (V, \prec) .

$ii) \implies v)$ Soit deux intervalles $I_1 = [a_1, b_1]$ et $I_2 = [a_2, b_2]$. On a, quitte à échanger les rôles de I_1 et I_2 , que $b_1 \leq b_2$. Par définition d'un ordre d'intervalle, on a $A(I_2) \leq A(I_1)$.

$v) \implies iv)$ Si (V, \prec) a le poset de la Figure 2.2 comme sous poset, on a $A(a) \not\leq A(b)$ et $A(b) \not\leq A(a)$. \square

2.2.3 Graphes sans- P_4

L'algorithme suivant donne une coloration réalisable pour tout hypergraphe et tout ordre total sur les sommets en entrée.

L'algorithme First-Fit :







Données : Un (hyper)graphe G et un ordre total sur $V(G)$.

Résultat : Une coloration des sommets de G .

Considérer les sommets dans l'ordre et donner à chacun la plus petite couleur possible (sans créer de conflit avec les sommets précédemment coloriés.)

L'algorithme First-Fit ne donne pas toujours une coloration optimale. Par exemple, si First-Fit prend le graphe $G = (\{1, 2, 3, 4\}, \{(1, 3), (2, 4), (3, 4)\})$ avec l'ordre $\{1, 2, 3, 4\}$ en entrée, il rend la 3-coloration $\{1, 2\}, \{3\}, \{4\}$ alors que $\chi(G) = 2$. D'un autre coté, si G est sans- P_4 , First-Fit est optimal.





Théorème 2.20 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i)  G est sans- P_4 .
- ii)  G peut être obtenu à partir d'un seul sommet en ajoutant itérativement des vrais et des faux jumeaux.
- iii)  Pour tout $U \subseteq V$, $G[U]$ est connexe si et seulement si $\overline{G[U]}$ ne l'est pas.
- iv)  L'algorithme First-Fit donne une coloration optimale pour tout ordre total sur V en entrée.
- v) Tout α -stable de G intersecte toute ω -clique de G .
- vi) Tout stable maximal de G intersecte toute clique maximale de G .
- vii) Tout sous-graphe induit connexe de G a un diamètre d'au plus 2.
- viii)  G est le graphe de comparabilité d'un poset séries-parallèles.
- ix)  Pour tout $d : V \rightarrow \mathbb{N}$, il existe une coloration $\phi : V \rightarrow \mathbb{N}$ telle que $\phi(v) \leq d(v)$ pour tout v si et seulement si pour toute clique K de G , on a $|K| \leq \max\{d(v) \mid v \in K\}$.

2.2.3.1 Graphes trivialement parfaits, P_4, C_4



Un graphe vérifiant les conditions du Théorème 2.21 est **trivialement parfait** [88, 87].

Théorème 2.21 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i)  G et n'a pas de P_4 ni de C_4 induit.
- ii)  Tout sous-graphe induit connexe de G a un sommet universel.
- iii)  G est le graphe d'intersection d'une famille laminaire.
- iv)  G est le graphe de comparabilité d'un poset dont le diagramme de Hasse est une forêt orientée (entrante ou sortante).

2.2.3.2 Graphe de partition, multiparti complet et P_3

Théorème 2.22 Soit G un graphe. Les conditions suivantes sont équivalentes


- i) G est une union disjointe de cliques
- ii) \overline{G} est multiparti complet
- iii) $\ominus G$ est sans- P_3
- iv)  G est le line-graphe d'une union disjointe d'étoiles
- v)  G est le graphe de comparabilité d'un poset dont le diagramme de Hasse est une union de chaînes.
- vi) La fonction $U \mapsto \alpha(G[U])$ est sous-modulaire (sur $\mathcal{P}(V)$).
- vii) La fonction $U \mapsto \overline{\chi}(G[U])$ est sous-modulaire.

2.2.4 Améliorer la perfection

2.2.4.1 Graphes de Meyniel


Un graphe vérifiant les conditions du Théorème 2.23 est de **Meyniel** .

Théorème 2.23 Soit G un graphe. Les conditions suivantes sont équivalentes :

- i) Tout cycle impair de G a au moins deux cordes.
- ii) $\ominus G$ est sans trou impair ni maison.
- iii)  G^\diamond est parfait pour toute partition partielle de $V(G)$ par des cliques ($\iff \overline{G}$ est PrExt-parfait).

2.2.4.2 Graphes de parité et de Gallai

Théorème 2.24 [37] Soit G un graphe. Les conditions suivantes sont équivalentes





- i) Pour toute paire de sommets (v, w) , tous les chemins induits joignant v et w ont la même parité.
- ii) Tout cycle impair de G a au moins deux cordes croisées.
- iii) $\ominus G$ est sans trou impair, ni maison, ni $\overline{P_4 + v}$.
- iv)  $G \times K_2$ est parfait [60, 99].

Un graphe vérifiant les conditions du Théorème 2.24 est dit de **parité** [36]

Un graphe est de **Gallai**¹⁴ si tout cycle impair de G a au moins deux cordes non-croisées.

2.2.4.3 Graphes triangulés sans-diamant

Théorème 2.25 [60] *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i)  G sans trou ni diamant.
- ii)  G est obtenu à partir de cliques par union disjointe et par sommets d'articulation.
- iii)  $G \times K_3$ est parfait.
- iv)  $G \times K_k$ est parfait pour tout $k \in \mathbb{N}$.

2.2.4.4 Graphes unimodulaires et équilibrés

Un graphe est **unimodulaire** si sa matrice d'incidence cliques-max-sommets est totalement unimodulaire. Un graphe est **équilibré** si sa matrice d'incidence cliques-max-sommets est équilibrée.

Proposition 2.26 *La classe des graphes unimodulaires est fermée par sous-graphe induit, par sommet d'articulation, et par réplication.*

Preuve Ces opérations peuvent être définies sur les matrices en général, et elles préservent la propriété TTU. \square

2.2.5 Classes de graphes pas forcément parfaits

D'autres classes de graphes jouent un rôle important dans les applications que les sous-classes de graphes parfaits. Des caractérisations intéressantes sont parfois connues.

2.2.5.1 Graphes d'arcs circulaires

Un **graphe d'arc(s) circulaire(s)** est le graphe d'intersection d'intervalles du cercle. Contrairement aux familles d'intervalles de la droite réelle, les familles d'in-




¹⁴*i-triangulated.*

tervalles du cercle ne sont pas toujours Helly (on peut avoir trois tels intervalles qui s'intersectent tous deux à deux mais dont l'intersection commune est vide).

2.2.5.2 Graphes planaires externes

Un *graphe* est *planaire-externe*¹⁵ si l'on peut le dessiner de manière planaire avec tous ses sommets sur la face extérieure.



Théorème 2.27 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i)  G est outer-planar.
- ii)  G sans mineur- $\{K_{2,3}, K_4\}$
- iii)  G est sans- $\{\text{mineur}, Y-\Delta\}$ - $\{K_4\}$



2.2.5.3 Graphes liés à la relaxation graphique du TSP

La lecture préalable des Sections 1.4.2.4, 2.1.3.2 et 2.1.3.3 est nécessaire à la lecture de la présente section.

Théorème 2.28 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i) G est GTSP-sous-modulaire [94].
- ii) Il existe au plus deux chemins non-triviaux¹⁶ sommets-disjoints entre toutes paires de sommets [94].
- iii)  G est obtenu à partir de copies de K_4 et de graphes planaires externes par sommets d'articulation [94].
- iv)  G sans mineur- $K_{2,3}$.

Théorème 2.29 [77] *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i) G est GTSP-parfait.
- ii)  G est obtenu par les opérations de compositions définies dans [77].
- iii)  G sans mineur- $\{M_1, M_2, \overline{C_6}\}$.

La contraction $Y-\Delta$ permettrait-elle une simplification, dans la description des graphes TSP-parfaits, et plus généralement, dans les classes de graphes liées à la relaxation graphique du TSP ? Un premier pas consisterait à éclaircir les conditions sous lesquelles cette opération s'applique. Par exemple,

¹⁵outer-planar.

¹⁶i.e. comportant au moins 2 arêtes.

Question 2.30 *Les graphes TSP-parfaits sont-ils les mêmes que les graphes sans- $\{mineur, Y-\Delta\}$ - $\overline{C_6}$*

La stabilité de l'opération Δ - Y est même envisageable dans les graphes TSP-parfaits. Une piste pour expliquer que ces deux opérations puissent s'appliquer est que les tours de Steiner dans les deux graphes (avant et après contraction) sont presque en bijections. Plus précisément, la bijection a lieu si l'on oublie la condition de couverture du sommet qui disparaît dans la contraction.

Dans l'étude des graphes GTSP-équilibrés, il reste peut-être un pas sérieux à franchir pour montrer que $\{M_1, M_2\}$ sont les seuls mineurs exclus. De plus, une caractérisation par décomposition, pour décrire les graphes GTSP-équilibrés reste encore à proposer.

Conjecture 2.31 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i) G est GTSP-équilibré.
- ii) G est GTSP-totalement-équilibré.
- iii) \mathcal{A} G sans mineur- $\{M_1, M_2\}$.
- iv) \ominus G sans $\{mineur, Y-\Delta\}$ - M_2 .

2.3 Classes d'hypergraphes héréditaires

Nous notons que, dans les hypergraphes, les problèmes de la couverture par des sommets, de la couverture par des arêtes et de la coloration sont conceptuellement équivalents (Section 2.3.2). Nous discutons ensuite dans quelle mesure les opérations de dualité définies dans les matroïdes et les clutters sont compatibles. Cela fait en particulier apparaître un lien élémentaire (mais fondamental) entre la coloration, le bloquant et le dual au sens des matroïdes (Section 2.3.3). Vient ensuite le tour des mineurs d'être interprétés de manière commune dans les matroïdes, les clutters conformes et plus généralement les hypergraphes héréditaires. Grâce à cette définition de mineurs, nous mentionnons comment caractériser les indépendants des matroïdes (Section 2.3.5) et les hypergraphes conformes (Section 2.3.6), à l'aide d'une unique obstruction parmi l'ensemble des hypergraphes héréditaires.

2.3.1 Petite philosophie à l'usage des (non-)spécialistes : l'intérêt de la généralisation aux hypergraphes

Dans la Section 2.3, nous présentons brièvement quelques généralisations communes des concepts qui s'appliquent à la fois dans les graphes (au sens hypergraphes conformes) et les matroïdes. Le but originel de cette description commune est de

montrer que les hypergraphes donnent un cadre conceptuel simple pour décrire les “problèmes classiques” étudiés dans [129], mais aussi certains “problèmes nouveaux”, comme la coloration bornée. L’espoir à plus long terme est d’étudier des généralisations communes de structures combinatoires pour lesquelles l’approche polyédrale permet de décrire les problèmes. Si l’on pense à la propriété TDI des inégalités de rang par exemple, on peut rêver d’une généralisation commune des graphes parfaits, des line-graphes et des matroïdes. Si l’on pense à la propriété IRP pour le problème de la coloration minimum, on peut aussi espérer une généralisation commune des graphes parfaits et des matroïdes. Soyons honnête, l’approche proposée ici ne donne pas d’espoir, à court terme, de comprendre ces éventuels objets. Mais à défaut de permettre de prendre le taureau par les cornes, cette approche permet de le prendre par la queue, en rappelant que de telles généralisations peuvent aussi être étudiées sous l’angle des hypergraphes qui leur font obstruction. Comme premier pas dans cette direction, nous observons que les matroïdes et les hypergraphes conformes peuvent être décrits de manière extrêmement simple en termes de mineurs exclus d’hypergraphes.

2.3.2 Couplage, couverture et coloration

Soit $\mathcal{H} = (V, \mathcal{E})$ un hypergraphe. Une *couverture par des sommets* de \mathcal{H} est un ensemble de sommets $T \subseteq V$ tel que toute arête contient au moins un sommet de T . A tout hypergraphe on associe la quantité $\tau(\mathcal{H})$, comme la cardinalité minimum d’une couverture par des arêtes. Un ensemble d’arêtes M qui ont deux-à-deux une intersection vide s’appelle un couplage (ou empilement d’arêtes). La cardinalité maximum d’un couplage est notée $\nu(\mathcal{H})$.

Pour tout hypergraphe \mathcal{H} et tout couplage M , une couverture par des sommets doit contenir au moins un sommet par arête de M . On a donc $\nu(\mathcal{H}) \leq \tau(\mathcal{H})$.

De manière transposée, un recouvrement est un ensemble d’arêtes qui recouvrent les sommets de \mathcal{H} . La cardinalité minimale d’un recouvrement est notée $\rho(\mathcal{H})$. On a $\rho(\mathcal{H}) = \tau(\mathcal{H}^T)$. Un ensemble de sommets S tel que $S \cap e \leq 1$ pour tout $e \in \mathcal{E}$ est appelé fortement stable. La cardinalité maximale d’un ensemble fortement stable est notée $\bar{\alpha}(\mathcal{H})$. On a $\bar{\alpha}(\mathcal{H}) = \nu(\mathcal{H}^T)$ et finalement $\bar{\alpha}(\mathcal{H}) \leq \rho(\mathcal{H})$.

Etant donné un hypergraphe, une couleur est un ensemble de sommets qui ne contient aucune arête. La cardinalité maximale d’une couleur est notée $\underline{\alpha}(\mathcal{H})$. Une coloration est une partition de V en couleurs. De manière plus classique, une coloration est une application ϕ de V dans \mathbb{N} telle que dans toute arête, il y a au moins deux sommets de couleurs différentes (autrement dit $|\phi(e)| \geq 2$ pour tout $e \in \mathcal{E}$). Le nombre minimum de couleurs nécessaires pour colorier \mathcal{H} est noté $\chi(\mathcal{H})$.

On peut transformer le problème de coloration en un problème de recouvrement en considérant l’hypergraphe colorant $col(\mathcal{H})$ de \mathcal{H} qui a le même ensemble de sommet

que \mathcal{H} et qui a l'ensemble des couleurs de \mathcal{H} comme arêtes. On a alors :

Proposition 2.32 *Pour tout hypergraphe \mathcal{H} , $\chi(\mathcal{H}) = \rho(\text{col}(\mathcal{H})) = \tau((\text{col}(\mathcal{H}))^T)$*

□

2.3.3 Dualités et coloration

Nous avons défini le transposé d'un hypergraphe (habituellement appelé "dual" et noté \mathcal{H}^*). Nous allons voir que d'autres opérations méritent aussi le qualificatif de "dual", lorsque l'on se restreint à certaines familles d'hypergraphes. Nous pensons donc que le terme "dual" devrait être évité autant que possible lorsque l'on définit une opération particulière de dualité. A la place, il nous semble plus pertinent de choisir systématiquement une terminologie particulière.

Le terme de dualité réfère en général au fait qu'une application est involutive. Pour un ensemble E , une application f de E dans lui-même est dite *involutive* si $f(f(x)) = x$ pour tout $x \in E$.

Pour de nombreuses opérations f , on a $f(f(f(x))) = f(x)$ pour tout x , mais pas $f(f(x)) = x$. Pour une application $f : E \rightarrow E$ telle que $f(f(f(x))) = f(x)$, on appelle $f(f(x))$ la *fermeture* de x . Un élément x est *fermé* s'il est égal à sa fermeture.

Pour un hypergraphe simple \mathcal{H} , on définit :

$$\mathcal{H}^{\min} = (V, \{F \in \mathcal{E} \mid F \text{ est minimal par inclusion dans } \mathcal{E}\})$$

$$\mathcal{H}^{\max} = (V, \{F \in \mathcal{E} \mid F \text{ est maximal par inclusion dans } \mathcal{E}\})$$

$$\mathcal{H}^\uparrow = (V, \{F \subseteq V \mid F \text{ contient une arête de } \mathcal{E}\})$$

$$\mathcal{H}^\downarrow = (V, \{F \subseteq V \mid F \text{ est contenu dans une arête de } \mathcal{E}\})$$

Un *clutter* est un hypergraphe tel qu'aucune arête n'en contient une autre. Pour tout hypergraphe \mathcal{H} , \mathcal{H}^{\min} et \mathcal{H}^{\max} sont donc des clutters.

Un hypergraphe est *héréditaire* si $\mathcal{H} = \mathcal{H}^\downarrow$.

Pour un hypergraphe $\mathcal{H} = (V, \mathcal{E})$, le *bloquant* $b(\mathcal{H}) = (V, \mathcal{T})$ est l'hypergraphe ayant pour arêtes l'ensemble des couvertures par des sommets (minimales par inclusion) de \mathcal{H} . Le bloquant d'un hypergraphe est donc toujours un clutter. En fait, les ensembles fermés pour la dualité bloquante sont exactement les clutters :

Proposition 2.33 [129, 49] *Pour tout hypergraphe \mathcal{H} , $b(b(\mathcal{H})) = \mathcal{H}^{\min}$. En particulier, $b(b(\mathcal{H})) = \mathcal{H}$ si et seulement si \mathcal{H} est un clutter.*

Il existe au moins une autre opération de dualité sur les hypergraphes qui a les clutters comme ensembles fermés : la dualité telle qu'on la conçoit pour les bases

d'un matroïde.

Le *dual au sens matroïdal* d'un clutter $\mathcal{C} = (V, \mathcal{E})$, est le clutter $dm(\mathcal{C}) = (V, \overline{\mathcal{E}})$ où $\overline{\mathcal{E}} := \{V \setminus E \mid E \in \mathcal{E}\}$. Le dual au sens matroïdal d'un clutter est donc toujours un clutter.

Proposition 2.34 *Pour tout clutter \mathcal{C} , $dm(dm(\mathcal{C})) = \mathcal{C}$.* □

Une troisième opération intéressante s'applique aux hypergraphe, mais cette fois, sans mener à un phénomène de dualité. Pour un hypergraphe $H = (V, \mathcal{E})$, le *colorant* $col(H) = (V, \mathcal{F})$ est le clutter des ensembles maximaux qui ne contiennent aucun élément de \mathcal{E} . Autrement dit, le colorant de H est le clutter des couleurs maximales (par inclusion) de H .

Le colorant est lié au bloquant et au dual au sens matroïdal de la manière suivante :

Théorème 2.35 *Pour tout hypergraphe $H = (V, \mathcal{E})$, on a $col(H) = dm(b(H))$. De plus, si l'on se restreint aux clutters, col est une opération inversible.*

Preuve Pour $X \subseteq V$, les conditions suivantes sont équivalentes :

- X intersecte toutes les arêtes de \mathcal{E}
- $V \setminus X$ ne contient aucune arête de \mathcal{E}

col est une opération inversible sur les clutters car elle résulte de la composition de deux opérations inversibles. □

Pour tout clutter H , il existe donc un unique clutter H' tel que $col(H') = H$. On note donc $H' = col^{-1}(H)$.

Le Théorème 2.35 permettrait de définir plusieurs notions des graphes et des matroïdes dans un langage unifié.

Par exemple, dans un matroïde \mathcal{M} , on a $col(\{\text{circuits}(\mathcal{M})\}) = \{\text{bases}(\mathcal{M})\}$ et $col(\{\text{bases}(\mathcal{M})\}) = \{\text{hyperplans}(\mathcal{M})\}$.

Dans un graphe G , on a $col(\{\text{arêtes}(G)\}) = \{\text{stables}(G)\}$ et $col(\{\text{cliques}(G)\})$ est l'ensemble des couleurs maximales possibles dans une clique-coloration de G .

2.3.4 Mineurs de clutters et d'hypergraphes héréditaires

Nous adoptons la définition de mineurs au sens de Schrijver [129] pour les Hypergraphes et de Cornuéjols [49] pour les clutters. Ces définitions sont compatibles. De plus, la définition de mineurs dans les matroïdes [129] est aussi compatibles avec celles-ci si l'on assimile un matroïde au clutter de ses bases.

Le passage à l'induit dans les graphes (effacer le sommet v) peut aussi être complété par une deuxième opération, qui consiste à mettre une boucle sur tous les voisins de v . Cela donne aussi une notion de mineur compatible avec les notions ci-dessus.


Elle admet aussi une interprétation pour le problème du stable de poids maximum : contracter un sommet équivaut à lui affecter un poids infiniment plus grand que le poids de tous les autres sommets. On n'utilise pas cette opération en général car elle donne, en un sens, un cas particulier du passage au sous-graphe induit.


Cependant, cela mène à l'utilisation de graphes avec boucles, qui semble être la bonne notion pour étudier les problèmes d'indépendants de poids maximum dans un cadre commun aux graphes et aux matroïdes.


2.3.5 Matroïdes


Les matroïdes peuvent être caractérisés simplement et de plusieurs manières parmi l'ensemble des hypergraphes héréditaires [129]. Nous proposons une caractérisation par mineur exclus qui n'est pas mentionnée habituellement. Cela consiste à noter que certaines des opérations connues sur les matroïdes (passage au mineur) sont compatibles avec la définition plus générale connue dans les hypergraphes. Une telle caractérisation pour les systèmes descriptibles par l'intersection de deux matroïdes n'est pas connue (bien qu'une caractérisation partielle soit donnée dans le Théorème 2.15).


Théorème 2.36 *Soit $H = (V, \mathcal{E})$ un hypergraphe simple héréditaire. Les conditions suivantes sont équivalentes :*

i)  H est un matroïde.

ii)  H est sans mineur- $\overline{P_3}$.

iii)  Le clutter $\text{col}^{-1}(H)$ est sans mineur- P_3 .

iv)  L'algorithme glouton donne un indépendant de poids maximum pour toute pondération $w : V \rightarrow \mathbb{N}$ [129].

v)  L'algorithme glouton donne un indépendant de cardinalité maximum de $H[U]$ pour tout $U \subseteq V$ et tout ordre sur U en entrée [129].

Idée de preuve Les clutters qui décrivent les bases (respectivement, les circuits) d'un matroïde sont stables par mineurs parmi l'ensemble des clutters.






v) \implies ii) Si l'algorithme glouton ne donne pas un indépendant de cardinalité maximum sur la restriction de H à U , soit X un ensemble rendu par l'algorithme glouton et $Y \in E(H)$ de cardinalité supérieure à X . \square

2.3.6 Graphes, conformalité et dualité anti-bloquante

Dans cette section, nous caractérisons les graphes parmi l'ensemble des hypergraphes. Pour ce faire il est plus agréable d'autoriser les boucles.

Un hypergraphe $\mathcal{H} = (V, \mathcal{E})$ est conforme si pour tout $U \subseteq V$ tel que chaque paire d'éléments de U soit contenu dans une arête de \mathcal{H} , l'ensemble U est aussi contenu dans une arête de \mathcal{H} . Notons que \mathcal{H} est conforme $\iff \mathcal{H}^{\max}$ est conforme $\iff \mathcal{H}^\downarrow$ est conforme.

Théorème 2.37 *Soit $H = (V, \mathcal{E})$ une hypergraphe simple héréditaire. Les conditions suivantes sont équivalentes :*

- i)  H est conforme.
- ii)  Le transposé H^T de \mathcal{H} est Helly.
- iii)  \mathcal{E}^{\max} est l'ensemble des cliques maximales d'un graphe $G = (V, E)$.
- iv)  H est sans mineur \mathcal{U}_3^2 .
- v)  $col^{-1}(H)$ est sans mineur \mathcal{U}_3^3 .
- vi) Pour tout $U \subseteq V$ tel que $r(\{v, w\}) = 2$ pour tout $v, w \in U$, on a $r(U) = |U|$.

Les indépendants d'un graphe $G = (V, E)$ définissent un hypergraphe héréditaire $\mathcal{H} = (V, \mathcal{I})$ dont le clutter des circuits $col^{-1}(\mathcal{H})$ est l'ensemble E , dont tous les éléments sont de taille 2. La classe des systèmes d'indépendance des graphes sans boucles n'est donc pas fermés par mineur (car contracter un sommet crée des boucles dans le graphe). Cependant, la classe des graphes avec possibilité de boucles est, elle, clairement fermée par mineur. Le seul mineur exclu est l'hypergraphe \mathcal{U}_3 qui a trois sommets et une seule arête de taille 3. De plus, comme nous l'avons explicité dans le Théorème 2.37, on peut caractériser les systèmes d'indépendance issus des graphes par le rang.

Pour un hypergraphe $\mathcal{H} = (V, \mathcal{E})$, l'antibloquant $a(\mathcal{H}) = (V, \mathcal{C})$ est l'hypergraphe ayant pour arêtes l'ensemble des fortement stables maximaux (par inclusion) de \mathcal{H} . On a clairement $a(\mathcal{H}) = a(\mathcal{H}^{\max}) = a(\mathcal{H}^\downarrow)$.

Proposition 2.38 *Pour un hypergraphe \mathcal{H} , $a(a(\mathcal{H})) = \mathcal{H}^{\max}$ si et seulement si \mathcal{H} est conforme.*

Un hypergraphe est donc “fermé” pour la “dualité” antibloquante si et seulement si c'est un clutter conforme.

2.4 Classes de fonctions d'ensemble

Les fonctions d'ensemble forment une famille d'objets combinatoires des plus généraux. Dans la Section 2.4.1, nous présentons quelques classes de fonctions classiques utilisées dans les problèmes de partition de coût minimum. Dans la Section 2.4.2 nous évoquons l'intérêt d'une classification systématique des fonctions d'ensembles.

2.4.1 Fonctions valeur-polymatroïdales

Plusieurs sous-classes de fonctions polymatroïdales jouissent de propriétés d'échanges plus fortes que la sous-modularité. La définition suivante permet de traduire de manière compacte l'une de ces propriétés¹⁷.

Une fonction f est *valeur-polymatroïdale* si $f(\emptyset) = 0$, f est non-décroissante, et pour tous $S, T \subseteq V$ tels que $f(S) \geq f(T)$ et tout $u \in V \setminus (T \cup S)$, on a

$$(2.4) \quad f(S + u) + f(T) \leq f(S) + f(T + u).$$

Proposition 2.39 *Une fonction valeur-polymatroïdale est polymatroïdale.*

Preuve Soit f une fonction valeur-polymatroïdale. Puisque f est non-décroissante, on a $f(S) \geq f(T)$ pour tout $T \subseteq S \subseteq V$. On a donc aussi $f(S + u) + f(T) \leq f(S) + f(T + u)$ pour tout $u \in V \setminus S$. D'après la condition (1.6) du Théorème 1.1 page 19, f est sous-modulaire. \square

La principale motivation derrière la définition des fonctions valeur-polymatroïdales est donnée dans la Proposition 2.40, que nous utilisons dans la Section 4.3.2 pour trouver une partition en cliques de coût minimum dans un graphe d'intervalle.

Proposition 2.40 *Pour tout hypergraphe héréditaire $H = (V, \mathcal{E})$ et toute fonction valeur-polymatroïdale f sur $V(G)$, il existe une partition $\{Q_1, Q_2, \dots, Q_k\}$ de V en éléments de \mathcal{E} qui minimise $\sum_i f(Q_i)$ telle que l'un des Q_i est un élément maximal (par inclusion) de \mathcal{E} .*

Preuve Soit \mathcal{Q} une partition de coût minimum de V . Choisissons $Q \in \mathcal{Q}$ tel que $f(Q) \geq f(T)$ pour tout $T \in \mathcal{Q}$. Si Q n'est pas un élément maximal de \mathcal{E} , il existe $t \in V \setminus Q$ tel que $Q + t \in \mathcal{E}$. Maintenant, t appartient à un $T \in \mathcal{Q} - Q$. Puisque f est non-décroissante, $f(Q) \geq f(T) \geq f(T - t)$. Puisque f est valeur-polymatroïdale, $f(Q + t) + f(T - t) \leq f(Q) + f(T)$. On répète alors ce procédé jusqu'à ce que Q devienne un élément maximal de \mathcal{E} . \square

En général, le rang d'un (poly)matroïde n'est pas valeur-polymatroïdal, et la Proposition 2.40 ne s'applique pas comme le montre la Figure 2.3.

¹⁷Les résultats de cette section ont fait l'objet d'un article commun avec Gijswijt et Queyranne [84].

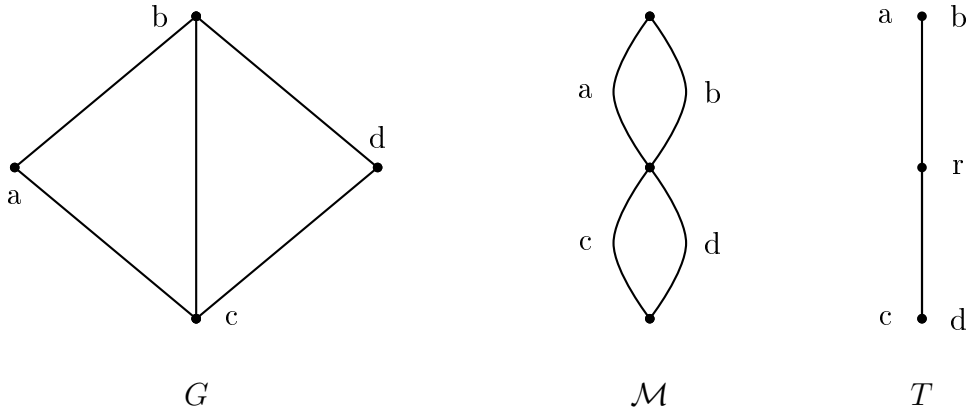


Figure 2.3 – La fonction f est donnée par le rang du matroïde \mathcal{M} . On obtient aussi $2f$ comme le plus court tour de Steiner sur l'arbre T enraciné sur r . Une partition de G en cliques de coût minimum est $\bar{\chi}(G, f) = 2 = f(\{a, b\}) + f(\{c, d\})$. Aucune partition en cliques de G de coût minimum ne contient une clique maximale.

2.4.1.1 Fonctions maximum et batchs parallèles

Pour $p : V \rightarrow \mathbb{R}_+$, définissons $f(U)$ pour tout $U \subseteq V$ par :

$$(2.5) \quad f(U) := \max_{u \in U} p(u)$$

Alors f est valeur-polymatroïdale. En effet, soient $S, T \subseteq V$ tels que $f(S) \geq f(T)$ et $u \in V \setminus (S \cup T)$. Alors, puisque $p(s) = f(S) \geq f(T) = p(t)$ pour $s \in S$ et $t \in T$, on a

$$f(S+u) + f(T) = \max\{p(s), p(u)\} + p(t) \leq p(s) + \max\{p(t), p(u)\} = f(S) + f(T+u).$$

Une fonction d'ensemble qui peut être définie selon (2.5) pour un certain p est une fonction ***p*-batch** ou ***max*-batch**.

2.4.1.2 Fonctions s-batch, batchs séries et fonctions sp-batch

Pour $p : V \rightarrow \mathbb{R}_+$ et $s \in \mathbb{R}_+$ définissons $f(U)$ pour tout $U \subseteq V$ par :

$$(2.6) \quad f(U) := s + \sum_{u \in U} p(u)$$

Une fonction d'ensemble qui peut être écrite selon (2.6) est une fonction ***s*-batch**, ou de batch en série. Ce type de fonction de coût arrive en ordonnancement par batch quand il y a un temps de réglage constant préalable au traitement d'un

ensemble de tâches¹⁸. Une fois le réglage effectué, la machine traite les tâches une par une, de manière séquentielle.

Les fonctions *sp-batch*¹⁹ généralisent à la fois les fonctions max-batches et s-batches : pour $p_1 : V \rightarrow \mathbb{R}_+$ et $p_2 : V \rightarrow \mathbb{R}_+$, posons pour tout $U \subseteq V$:

$$(2.7) \quad f(U) := \max_{u \in U} p_1(u) + \sum_{u \in U} p_2(u)$$

Les fonctions sp-batches sont valeur-polymatroïdales car :

Remarque 2.41 *Si f est une fonction valeur-polymatroïdale et g une fonction additive, alors $f + g$ est valeur-polymatroïdale.* \square

2.4.1.3 Fonctions cardinales tronquées et matroïde uniforme

Soit $b \in \mathbb{N}$. Pour tout $U \subseteq V$ posons :

$$(2.8) \quad f(U) := \min\{b, |U|\}$$

Une telle fonction de coût arrive lorsque l'on minimise le prix d'entrée à la piscine : Si le prix d'une entrée est 1 et qu'il existe un abonnement de coût b donnant droit à un nombre d'entrées illimitées, le prix minimum que l'on peut payer pour x entrées est $\min\{b, x\}$. Une telle fonction d'ensemble est *cardinale tronquée*. En posant, $n := |V|$, on peut aussi définir f comme le rang du matroïde uniforme \mathcal{U}_n^b . Nous présentons maintenant une classe plus générale.

2.4.1.4 Fonctions taille-définies concaves

Pour $\psi : \mathbb{N} \rightarrow \mathbb{R}_+$ telle que $\psi(0) = 0$. Pour tout $U \subseteq V$ posons :

$$(2.9) \quad f(U) := \psi(|U|)$$

Alors f est valeur-polymatroïdale $\iff f$ est polymatroïdale $\iff \psi$ admet une extension non-décroissante concave sur le segment $[0, |V|]$. Une telle fonction est *taille-définie concave* [84]. Non seulement les fonctions cardinales tronquées sont un cas spécial, mais toute fonction taille-définie concave entière s'écrit de manière unique comme combinaison conique de telles fonctions. En particulier

Remarque 2.42 *Les fonctions cardinales tronquées sont les rayons extrêmes du cône des fonctions taille-définies concaves.* \square

¹⁸set-up time.

¹⁹C'est un néologisme.

2.4.1.5 Probabilités indépendantes

Pour $p : V \rightarrow [0, 1]$, définissons $f(U)$ pour tout $U \subseteq V$ par :

$$(2.10) \quad f(U) := 1 - \prod_{u \in U} p(u) \text{ pour tout } U \subseteq V$$

Une telle fonction est **probabiliste**. Des références sur les applications de l'optimisation probabiliste sont disponibles dans [64].

Proposition 2.43 *Les fonctions probabilistes sont valeur-polymatroïdales*

Preuve Soit $S, T \subseteq V$ tels que $f(S) \geq f(T)$ et $u \in V \setminus (S \cup T)$. Posons $f(S) = 1 - \sigma$ et $f(T) = 1 - \tau$ (et donc $\sigma \leq \tau$). On a alors

$$\begin{aligned} f(S) + f(T + u) &= (1 - \sigma) + (1 - p(u)\tau) \\ &\geq (1 - p(u)\sigma) + (1 - \tau) = f(S + u) + f(T). \end{aligned}$$

□

2.4.2 Une théorie générale à venir

Les fonctions d'ensemble permettent non-seulement d'exprimer des fonctions de coût mais aussi des contraintes d'indépendance. Nous invitons le lecteur à noter que le problème de partition de coût minimum [Pconclu] (défini dans la conclusion de ce mémoire) est entièrement réductible à la donnée d'une fonction d'ensemble²⁰. Sans intérêt du point de vue de la complexité algorithmique, cette réduction montre toutefois l'énorme potentiel d'expressivité des fonctions d'ensemble. Cela indique surtout qu'une classification des fonctions d'ensemble, qui étende la classification connue des classes de graphes, serait aussi intéressante et générale que fastidieuse et technique. Une telle classification passerait donc par la mise en œuvre de paradigmes analogues à ceux présentés dans la Section 2.1. Cependant, rien n'indique que ces paradigmes puissent être étendus à autre chose que les graphes. D'autres paradigmes restent sûrement à trouver. La définition des fonctions valeur-polymatroïdales et surtout des fonctions sous-modulaires incitent à penser qu'un axe prometteur est d'analyser les fonctions d'ensemble selon les propriétés d'échange dont elles jouissent (comme les fonctions valeur-polymatroïdales ou sous-modulaires).

A court terme, ce qui me semble le plus réaliste, c'est de mettre à jour des caractérisations des classes de graphes (ou d'hypergraphes) intéressantes sur la base de propriétés de fonctions d'ensembles associées. A l'instar de la généralisation qui

²⁰Réaffecter le poids $+\infty$ aux ensembles qui ne sont pas admissibles pour former un batch.

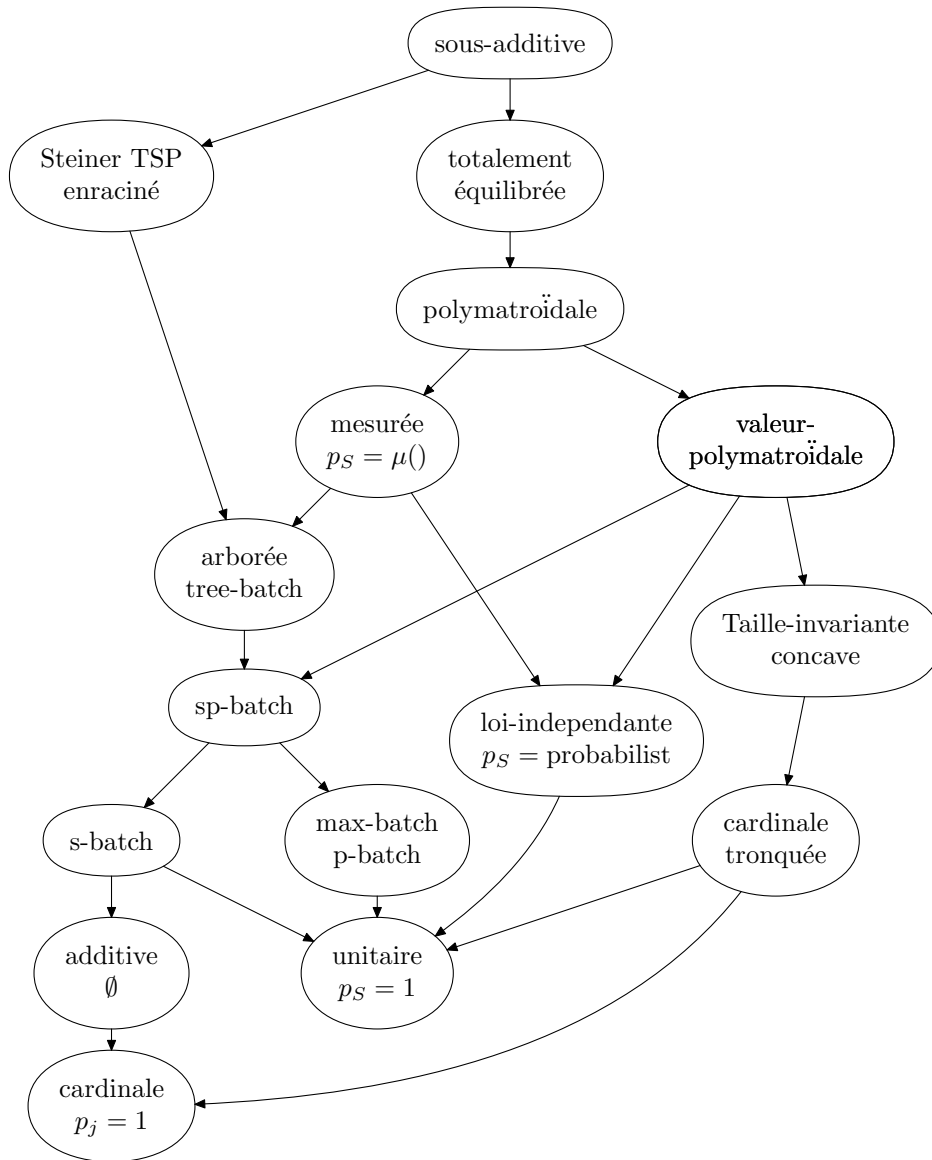


Figure 2.4 – Classes de fonction d'ensembles “discrètement convexes”

fait passer des matroïdes aux polymatroïdes puis aux fonctions sous-modulaires, j'espère que de telles caractérisations de classes de graphes mèneront à la découverte de classes de fonctions d'ensembles fondamentales, par la simple relaxation de la condition de sous-cardinalité.

A long terme, on peut espérer qu'une classification pertinente des fonctions d'ensemble nous guide dans l'élaboration de principes algorithmiques généraux (comme c'est déjà le cas pour ce qui relève de la sous-modularité). Les questions algorithmiques relatives aux fonctions totalement-équilibrées posées dans la Section 1.4 sont un exemple des espoirs balbutiants que fait germer cette "théorie générale" des structures combinatoires, lorsque l'on tente de fonder cette théorie dans le cadre formel des fonctions d'ensemble.

Deuxième partie

Théorèmes d'ordonnancement chromatique

Chapitre 3

La coloration bornée

Le nombre chromatique b -borné, $\chi_b(G)$, d'un graphe G est le nombre minimum de couleurs nécessaires pour colorier G avec la contrainte que chaque couleur soit utilisée au plus b fois (où b est un entier non-nul). Calculer χ_b est NP-dur, même dans les graphes bipartis ou les graphes d'intervalles. Nous étudions la frontière de polynomialité relative aux classes de graphes. Le résultat principal du chapitre est un algorithme polynomial pour les compléments de graphes triangulés, ce qui généralise plusieurs résultats récents. Cet algorithme s'insère dans une perspective plus large, basée sur l'étude d'une borne inférieure sur $\bar{\chi}_b(G) = \chi_b(\bar{G})$. Cette borne (appelée BT-mod) généralise les formules de Berge-Tutte pour le couplage non-biparti et l'inégalité de clique pour la coloration. Nous définissons et étudions les graphes bornés-parfaits, pour lesquels BT-mod est serrée pour tout sous-graphe induit et tout entier b . Nous caractérisons les graphes triangulés comme les graphes dont tous les répliqués sont bornés-parfaits et proposons plusieurs conjectures concernant les graphes bornés-parfaits.

Mots clés : coloration bornée, borne-perfection, graphe triangulé, décomposition simpliciale, formule de Berge-Tutte, réplique, graphe parfait, graphe sans- $K_{1,3}$, graphe sans- K_4 , complexité.

Autres objets utilisés : matroïde uniforme, forêt, couplage multidimensionnel (3-DM), approximation polynomiale.

Menu du jour

3.1	Quelques bases de coloration bornée	128
3.1.1	Philosophie : contraintes de cardinalité	129
3.1.2	Une caractérisation des graphes sans- $K_{1,3}$	131
3.1.3	Sous-problèmes NP-complets minimaux de [PCliqB]	132
3.2	La formule de Berge-Tutte modulo-b et les graphes bornés-parfaits	134
3.2.1	La formule de Berge-Tutte modulo- b	135
3.2.2	Les graphes bornés-parfaits	136
3.2.3	Borne-perfection, réplication et co-réplication	140
3.2.4	Une conjecture forte pour les bornés-parfaits	143
3.3	Algorithmes pour les graphes triangulés	146
3.3.1	Glouton pour les graphes d'intervalles	146
3.3.2	c -couplages pour les split graphes	151
3.3.3	Réseau simplicial pour les triangulés	154
3.4	Pour aller plus loin	164
3.4.1	Coloration avec bornes non-uniformes	164
3.4.2	Approximabilité et complexité	166
3.4.3	Programmes linéaires basés sur BT-mod	169
3.4.4	Applications et complexité de la (co)-réplication	171
3.4.5	[ColB] pour les arbres et borne-perfection faible	173
3.4.6	Extensions aux hypergraphes	176
3.4.7	Utilisation des forêts clique-connectantes	181

3.1 Quelques bases de coloration bornée

Dans une coloration d'un graphe, la *cardinalité* d'une couleur est le nombre de sommets coloriés avec cette couleur.

Dans de nombreuses applications de la coloration il convient de considérer des contraintes sur la cardinalité des couleurs¹. La *cardinalité maximum* d'une couleur est le nombre maximum de sommets que l'on peut colorier avec cette couleur. Nous considérons principalement le cas *uniforme* où toutes les couleurs ont une même cardinalité maximum b . Le problème central que nous étudions dans ce chapitre est donc :

[ColB] Coloration b -bornée (version décision)

¹Lire [22, 24, 76, 79, 107] pour connaître quelques applications. Lire l'Exemple 3.1 et la Remarque 3.2, puis réfléchir 2 minutes pour trouver une multitude d'applications.

Données : Un graphe G , deux entiers b et k .

Résultat : Est-ce que $\chi_b(G) \leq k$, *i.e.* est-ce qu'il existe une coloration de G en au plus k couleurs, chacune de cardinalité maximum b ?

Commentaires : NPC. Une coloration est un certificat NP. [Color] et Partition en triangles (sur le graphe complémentaire) sont des cas spéciaux.

En fait, il est souvent plus facile de décrire [ColB] sur le graphe complémentaire. Une ***b*-clique** est une clique de cardinalité au plus b . Une partition en b -cliques de G est une coloration b -bornée de \overline{G} . Nous étudions donc le problème :

[PCliqB] Partition en b -Cliques

Données : Un graphe G , deux entiers b et k .

Résultat : Est-ce que $\overline{\chi}_b(G) \leq k$, *i.e.* est-ce qu'il existe une partition de V en au plus k cliques, toutes de taille au plus b ?

Dans ce chapitre, le terme de ***batch*** est employé pour désigner une b -clique dans une partition en b -cliques de G .

3.1.1 Petite philosophie à l'usage des (non-)spécialistes : l'ubiquité des contraintes de cardinalité et des classes de graphes.

Dans cette section, nous évoquons l'importance des contraintes de cardinalité puis l'importance des classes de graphes dans l'analyse des problèmes de coloration.

Partons d'un exemple virtuel pour illustrer les applications de la coloration bornée.

Exemple 3.1 *Au cours d'un dîner de Gala, on souhaite répartir un ensemble de mathématiciens autour de tables. On souhaite leur permettre de se reposer intellectuellement lors du dîner. Pour ce faire, on évite de placer deux mathématiciens spécialistes d'un même domaine à une même table. Chacune des tables peut accueillir jusqu'à $b = 8$ personnes. On peut modéliser cela par le problème de coloration b -bornée d'un graphe dont les sommets correspondent aux mathématiciens et tel que deux mathématiciens sont reliés par une arête s'ils ont une spécialité en commun.*

On généralise aisément cet exemple aux "problèmes réels" de la recherche opérationnelle :

Remarque 3.2 *Les contraintes de cardinalité sur les couleurs sont nécessaires à la pertinence de la modélisation dans de nombreuses applications de la coloration, en particulier lorsqu'il s'agit de partitionner des objets du monde physique.*

Montrons maintenant pourquoi il est intéressant d'étudier ce problème sur certaines

classes de graphes.

Dans l'Exemple 3.1, si les mathématiciens n'ont qu'une seule spécialité, le graphe est une union disjointe de cliques (chaque clique correspond à une spécialité). Dans ce cas, on peut résoudre [ColB] par un algorithme glouton quel que soit b (en fait, cet algorithme glouton semble marcher même dans le cas non-uniforme, cf. Théorème 3.53). Supposons maintenant que chaque mathématicien ait (au plus) deux spécialités. Le graphe est alors un line-graphe (chaque spécialité est représentée par un sommet et chaque mathématicien est représenté par une arête²). [ColB] est alors NP-dur car [Color] l'est aussi. Cependant, de même que pour [Color], on peut donner une solution en temps polynomial pour [ColB] qui utilise au plus une couleur de plus que l'optimum (voir Section 3.1.2 et Corollaire 3.58 page 167). Voyons maintenant un exemple d'application pour [PCliqB].

Exemple 3.3 *Lors d'une conférence, on répartit des mathématiciens dans des chambres pouvant accueillir au plus $b = 4$ personnes. Afin de maximiser les interactions scientifiques, on tente de répartir les mathématiciens de manière à ce que, dans chaque chambre, chaque paire de mathématiciens ait une spécialité en commun. On peut modéliser cela par le problème de partition en b -cliques du graphe décrit dans l'Exemple 3.1.*

Encore une fois, si chaque mathématicien n'a qu'une seule spécialité, le graphe est une union disjointe de cliques. Notons $V = K_1, \dots, K_k$ l'ensemble de ces cliques. Dans le cas uniforme, un algorithme glouton donne encore la solution et on a $\bar{\chi}_b(G) = \sum_i \lceil |K_i|/b \rceil$ (voir Section 3.2). En revanche, dans le cas non-uniforme le problème est fortement NP-dur (voir Théorème 3.52 page 165).

Voici maintenant un exemple motivant une extension aux hypergraphes :

Exemple 3.4 *Au cours d'un déjeuner, des mathématiciens se répartissent autour de tables. Leur goût permanent du travail collectif les incite à choisir un unique sujet de discussion autour de chaque table. Un groupe de mathématiciens ne restera autour d'une table que s'ils partagent, tous ensemble, une même spécialité.*

Cette fois, le graphe décrit dans l'Exemple 3.1 ne contient pas assez d'information pour résoudre le problème de l'Exemple 3.4, car il est possible que 3 mathématiciens partagent une spécialité deux-à-deux sans en avoir une commune à tous les 3. Autrement dit, l'hypergraphe (mathématiciens-vs-spécialités) de l'Exemple 3.4 n'est pas forcément conforme (*i.e.* ne peut pas être représenté par les cliques d'un graphe). Le problème de la coloration bornée des hypergraphes est discuté dans la Section 3.4.6.1 dans le cas matroïdes.

²Si un mathématicien n'a qu'une seule spécialité, il peut être représenté par une arête le joignant à sa spécialité ainsi qu'à une "spécialité virtuelle" qui lui est propre.

3.1.2 Une caractérisation des graphes sans- $K_{1,3}$

Dans toute sous-classe de graphes sans- $K_{1,3}$, les problèmes [Color], [ColB] ainsi que la coloration équitable sont (polynomialement) équivalents. De plus, il y a un lien simple entre les trois “nombres chromatiques” correspondants [55, 79, 93]. Bien que ces résultats ne soient pas nouveaux, nous discutons ces réductions et ce lien car ils donnent une formule min-max pour $\chi_b(G)$ si G est parfait et sans- $K_{1,3}$. Cela sert d’introduction à la Section 3.2 dans laquelle nous donnons une formule min-max valide dans une classe plus générale de graphes.

Une *k -coloration équitable* de G est une k -coloration dans laquelle toutes les couleurs ont la même cardinalité avec une différence d’au plus 1. Soit $\chi_=(G)$ le plus petit k tel que G a une k -coloration équitable. Considérons le problème,

[ColEq] Coloration équitable

Données : Un graphe G et un entier k .

Résultat : Existe-t-il une k -coloration $\{C_1, C_2, \dots, C_k\}$ des sommets de G telle que $|C_i| - |C_j| \leq 1$ pour tout $i, j \in [1..k]$?

Le nombre chromatique borné d’un graphe est lié au nombre chromatique et au rapport $|V|/b$:

Ingrédient 3.5 [93] *Pour tout graphe G et tout entier b*

$$(3.1) \quad \chi_b(G) \geq \max\{\chi(G), \lceil |V|/b \rceil\}$$

Malheureusement, l’Ingrédient 3.5 n’a pas la forme requise pour donner une bonne caractérisation pour le problème de la coloration bornée même dans le cas où l’in-égalité (3.1) est serrée, puisqu’en général on ne peut pas certifier polynomialement que $\chi(G) \geq k$. Toutefois, bien qu’il ne donne pas de solution polynomiale à [ColB] pour les graphes sans- $K_{1,3}$, le Théorème 3.6 implique l’équivalence de [Color], de [ColB] et de [ColEq] pour toute sous-classe de graphes sans- $K_{1,3}$.

Théorème 3.6 [55, 79, 93] *Pour un graphe G , les conditions suivantes sont équivalentes :*

- i) G est sans $K_{1,3}$*
- ii) toute composante connexe d’un sous-graphe de G induit par deux stables dis-joints est une chaîne ou un cycle pair.*
- iii) pour tout sous-graphe induit H de G , on a l’égalité*

$$\chi_b(H) = \max\left\{\chi(H), \left\lceil \frac{|V(H)|}{b} \right\rceil\right\}$$

- iv) $\chi_=(H) = \chi(H)$ pour tout sous-graphe induit H de G .*

En fait, étant donnée une coloration optimale d'un graphe sans- $K_{1,3}$ et un entier b , on peut construire en temps polynomial une coloration b -bornée de G ayant $\max\{\chi(G), \lceil |V|/b \rceil\}$ couleurs [55, 79]. L'algorithme consiste en deux phases, "gaspillage de couleurs" et "échanges bichromatique". Le gaspillage consiste, étant donné un entier b et une k -coloration d'un graphe G , à recolorier arbitrairement certains sommets avec de nouvelles couleurs, de manière à obtenir une coloration de G utilisant $\max\{k, \lceil |V|/b \rceil\}$ couleurs.

L'algorithme Echange bichromatique :

Données : Un graphe G sans- $K_{1,3}$, un entier b et une k -coloration de G avec $k \geq \lceil |V|/b \rceil$.

Résultat : Une k -coloration b -bornée de G .

Tant qu'il y a une couleur C qui contient plus que b sommets :

Choisir une couleur D telle que $|D| < b$. Trouver une chaîne induite impaire dans $G[C \cup D]$ qui a ses extrémités dans C . Echanger les couleurs des sommets sur cette chaîne.

fin Tant que.

La validité et la polynomialité du procédé de gaspillage-échange ci-dessus donne :

Corollaire 3.7 [55, 79] *Si \mathcal{G} est une classe de graphes sans- $K_{1,3}$ alors [ColB] restreint à \mathcal{G} est polynomialement équivalent à [Color] restreint à \mathcal{G} . \square*

En particulier, [ColB] est polynomial pour les graphes parfaits sans- $K_{1,3}$ et donc pour les line-graphes de bipartis. Par ailleurs, le procédé d'échange bichromatique peut être adapté pour obtenir une k -coloration équitable d'un graphe G sans- $K_{1,3}$ à partir d'une k -coloration de G [55].

3.1.3 Sous-problèmes NP-complets minimaux de [PCliqB]

Dans cette section, nous décrivons les classes de graphes (minimales par inclusion) dans lesquelles [PCliqB] est NP-complet. Nous recensons trois telles classes. Le problème suivant, étudié par Dahlhaus et Karpinski [52] est lui aussi NP-complet dans ces trois classes de graphes.

[PerfectMDM] Couplage Multidimensionnel Parfait

Données : Un graphe $G = (V, E)$ et un entier b

Résultat : Existe-t-il une partition de V en cliques de cardinalité exactement b .

Commentaires : NPC. Polynomial sur les graphes triangulés [52].

Notons que [PerfectMDM] est un cas particulier de [PCliqB], mais aussi de [ColEq] (en passant au graphe complémentaire). Quand c'est possible, il est donc plus intéressant de prouver la NP-complétude de [PerfectMDM] sur \mathcal{G} que celle de [PCliqB]

sur \mathcal{G} . Notons que le complément d'un graphe obtenu par réplication d'un biparti complet est sans- $\{P_4, \text{paw}, 2K_2 + v\}$ et que l'union disjointe de split graphes complets est sans- $\{P_4, C_4, \text{paw}\}$.

Théorème 3.8 [*PerfectMDM*] est NP-complet même si l'on se restreint à l'une des classes suivantes

- i) bipartis complets répliqués
- ii) compléments d'unions disjointes de split graphes complets (amélioration de [22]).
- iii) co-bipartis (même si $k = 3$ est un paramètre fixé [22]).

Preuve i) On réduit [3-PARTITION] à [PerfectMDM] restreint aux graphes bipartis complets répliqués.

[3-Partition]

Données : Un ensemble A ayant $3m$ éléments. Une taille $s : A \rightarrow \mathbb{N}$. Une capacité $B \in \mathbb{N}$.

Résultat : Existe-t-il une partition \mathcal{A} de A en m ensembles: $\mathcal{A} = \{A_1, A_2, \dots, A_m\}$, telle que $|A_i| = 3$ et $\sum_{a \in A_i} s(a) = B$ pour tout i .

Commentaires : fortement NPC, même si $\frac{B}{4} < s(a) < \frac{B}{2}$ pour tout $a \in A$ [80].

Considérons le graphe biparti complet $G = ((A', C), E)$, où C est un ensemble à m éléments et A' est une copie de A . Répliquons $s(a)$ fois chaque sommet $a \in A'$ et $2B$ fois chaque sommet $c \in C$. Soit \widehat{G} le graphe ainsi obtenu (cette réduction est polynomiale car [3-PARTITION] est fortement NP-complet). Notons que \widehat{G} a exactement $3mB$ sommets. On vérifie que G a un couplage B -dimensionnel parfait si et seulement si \mathcal{I} est une instance "oui" de [3-PARTITION] :

- Etant donné une solution \mathcal{A} de [3-PARTITION], on construit une partition en $3m$ B -cliques de \widehat{G} en choisissant arbitrairement un sommet c_i de C pour chaque classe $A_i = \{a_{i1}, a_{i2}, a_{i3}\}$ de \mathcal{A} . Pour $j = 1, 2, 3$, on forme alors 3 batchs ayant respectivement les $s(a_{ij})$ sommets de a_{ij} et $B - s(a_{ij})$ sommets de c_i .
- Soit $\mathcal{K} = \{K_1, K_2 \dots K_{3m}\}$ une $3m$ -partition en cliques de tailles b de \widehat{G} . Après réplication, A' et C induisent chacun des unions disjointes de cliques. Par non-connexité, chaque K_i intersecte au plus une clique de A' et au plus une clique de C . Comme A' contient un stable de taille $3m$, chaque batch K_i contient complètement l'une des cliques de A' . Comme $s(a_{ij}) < B$ pour tous i, j , chaque K_i intersecte exactement une clique de C . L'ensemble C définit donc une partition sur A : $a_j, a_k \in A$ sont dans la même classe A_i si et seulement s'ils sont contenus dans deux K_i qui intersectent une même clique de C . On peut vérifier que $|A_i| = 3$. En effet, si $|A_i| \leq 2$, on a $2B + \sum_{a \in A_i} s(a) > B|A_i|$ (car $0 < s(a)$). Si $|A_i| \geq 4$, on a $2B + \sum_{a \in A_i} s(a) < B|A_i|$ (car $s(a) < B/2$). On a donc que $|A_i| = 3$, que $3B = 2B + \sum_{a \in A_i} s(a)$ et finalement que $\sum_{a \in A_i} s(a) = B$ pour chaque i .

ii) On réduit [3-PARTITION] à [PerfectMDM] restreints aux compléments d'union disjointes de split graphes complets. Pour chaque $a \in A$, construisons un split graphe complet G_a ayant pour sommets (S_a, K_a) avec $|S_a| = s(a) + 1$ et $|K_a| = m - 1$. Soit G l'union disjointe des G_a . Soit $D = B + 3m$. Observons que $Dm = |V(G)|$. On vérifie que \overline{G} a un couplage D -dimensionnel parfait si et seulement si \mathcal{I} est une instance "oui" de [3-PARTITION] :

- Etant donné une 3-Partition de A , on construit une partition de $V(\overline{G})$ en $3m$ stables de taille D : pour chaque classe $A_i = \{a_{i1}, a_{i2}, a_{i3}\}$ de la 3-Partition, on construit un stable contenant $\cup_j S_{a_{i,j}}$ et un sommet de chaque K_a pour $a \in A \setminus \{a_{i1}, a_{i2}, a_{i3}\}$.
- Soit $\mathcal{B} = \{B_1 \dots B_m\}$ une partition de $V(G)$ en m stables de taille D . Comme pour chaque a , G_a est un split graphe complet tel que $\omega(G_a) = m$, B_i contient soit un sommet de K_a soit tous les sommets de S_a . \mathcal{B} définit donc une partition sur A : $a_1, a_2 \in A$ sont dans la même classe si S_{a_1} et S_{a_2} sont contenus dans le même B_i . Soit t_i le nombre de stables S_a contenus dans B_i . On a

$$3m + t_i B / 4 < |B_i| = 3m + B < 3m + t_i B / 2$$

Donc $t_i = 3$ pour tout i et

$$\sum_{a | S_a \subseteq B_i} |S_a| = B$$

iii) Dans [22], la NP-complétude de [PCliqB] sur les graphes co-bipartis est redémontrée, mais la preuve donne directement la NP-complétude de [PerfectMDM] sur les co-bipartis car dans la réduction, les $k = 3$ stables de la partition ont tous la même taille. \square

Les classes de graphes "intéressantes" qui contiennent l'une des classes du Théorème 3.8 et dans lesquelles [PerfectMDM] et [PCliqB] sont par conséquent NP-complets sont présentées dans la Figure A.3 page 260. La NP-complétude de [ColB] sur les graphes trivialement parfaits venant du cas ii) et celle de [PCliqB] sur les unimodulaires vient du cas i) car les unimodulaires sont fermés par réplification (Proposition 2.26 page 111) et contiennent les bipartis.

3.2 La formule de Berge-Tutte modulo- b et les graphes bornés-parfaits

Dans cette section nous donnons une borne inférieure³ pour $\overline{\chi}_b(G)$ qui généralise à la fois l'inégalité $\overline{\chi}(G) \geq \alpha(G)$ et l'inégalité de Berge-Tutte pour le couplage maximum. Nous étudions les graphes pour lesquels cette borne est serrée. Nous nous concentrons sur la partition en b -cliques plutôt que sur la coloration bornée

³Cette borne inférieure a été découverte en collaboration avec G. Finke, M. Queyranne et A. Sebő [76].

car il est plus facile d'utiliser des cliques, avec lesquelles nous manipulons des composantes connexes, que des stables, qui impliquent de manipuler des composantes co-connexes.

3.2.1 La formule de Berge-Tutte modulo- b

Nous présentons ici une borne inférieure sur $\bar{\chi}_b(G)$ valide pour tout graphe G et tout entier b . Il y a trois ingrédients dans la recette de cette borne inférieure :

Ingrédient 3.9 $\bar{\chi}_b(G)$ est une fonction d'ensemble non-décroissante sur V car une partition en b -cliques de G induit une partition en b -cliques de $G[U]$ pour tout $U \subseteq V$. \square

Ingrédient 3.10 $\bar{\chi}_b(G)$ est égal à la somme des $\bar{\chi}_b$ sur l'ensemble des composantes connexes de G .

Preuve Si G n'est pas connexe, comme une clique est toujours contenue dans une composante connexe, le problème se décompose en sous-problèmes indépendants (un par composante connexe). \square

Ingrédient 3.11 Pour tout $U \subseteq V$, le nombre de b -cliques nécessaires pour partitionner $G[U]$ est au moins $\lceil |U|/b \rceil$. \square

Pour $U \subseteq V$, notons $C_1(U), C_2(U), \dots, C_t(U)$ les sous-ensembles de sommets des composantes connexes de $G[U]$. Définissons

$$(3.2) \quad \sigma_b(G) := \max_{U \subseteq V} \sum_{i=1}^t \left\lceil \frac{|C_i(U)|}{b} \right\rceil$$

Un ensemble U *définit* σ_b si c'est un argument du maximum dans (3.2).

Proposition 3.12 [76] Pour tout graphe G et tout entier positif b ,

$$(3.3) \quad \bar{\chi}_b(G) \geq \sigma_b(G) \quad (\text{Inégalité BT-mod})$$

De plus, étant donné G, b et k , décider si $\sigma_b(G) \geq k$ est dans NP.

Preuve Les nombres entre parenthèses indiquent le numéro de l'ingrédient à utiliser pour la relation considérée. Pour tout $U \subseteq V$, on a

$$\bar{\chi}_b(G) \stackrel{(3.9)}{\geq} \bar{\chi}_b(G[U]) \stackrel{(3.10)}{=} \sum_i \bar{\chi}_b(G[C_i(U)]) \stackrel{(3.11)}{\geq} \sum_i \left\lceil \frac{|C_i(U)|}{b} \right\rceil$$

Finalement, puisque l'inégalité ci-dessus est vraie pour tout U , elle le reste en passant au maximum sur $U \subseteq V$. L'appartenance à NP vient du fait que pour un

sous-ensemble U donné, on peut construire les composantes connexes de $G[U]$ en temps polynomial. Autrement dit, un ensemble U qui définit $\sigma_b(G)$ est un certificat NP. \square

Nous avons choisi d'appeler l'inégalité (3.3) la formule de Berge-Tutte modulo b (ou BT-mod) à cause de sa similarité avec la formule qui donne la cardinalité d'un couplage maximum dans un graphe. Montrons que BT-mod donne lieu à une formule min-max généralisant à la fois "Berge-Tutte-Gallai" concernant le nombre minimum d'arêtes nécessaires pour couvrir les sommets d'un graphe et l'égalité $\bar{\chi}(G) = \alpha(G)$ pour la partition en cliques des graphes parfaits.

Proposition 3.13 *Soit G un graphe et b un entier. On a $\bar{\chi}_b(G) = \sigma_b(G)$ dans les cas suivants :*

- i) $b = 2$ [76]
- ii) $\bar{\chi}(G) = \alpha(G)$ et $b \geq \omega(G)$

Preuve i) Une partition en 2-cliques de G est une partition de $V(G)$ qui contient des cliques de taille 1 et 2, autrement dit "des arêtes" et "des sommets". Si G a un sommet isolé v , on a $\bar{\chi}_2(G - v) = \bar{\chi}_2(G) - 1$ et $\sigma_2(G - v) = \sigma_2(G) - 1$. Il suffit donc de prouver la formule pour les graphes sans sommets isolés. Or dans un tel graphe, une partition en 2-cliques est équivalente à une couverture en arêtes et on a $\bar{\chi}_2(G) = \rho(G)$. Le fait que $\rho(G) = \sigma_2(G)$ dans un graphe sans sommets isolés (voir détail à la page 461 de [129]) termine la preuve.

ii) Si $b \geq \omega(G)$, la contrainte de cardinalité des cliques est une contrainte superflue, c'est-à-dire que toute clique est aussi une b -clique, donc $\bar{\chi}_b(G) = \bar{\chi}(G)$. De plus, pour tout graphe G , tout entier b et tout stable S de G , on a

$$(3.4) \quad \sigma_b(G) \geq \sum_{i=1}^t \left\lceil \frac{|C_i(S)|}{b} \right\rceil = |S|$$

On a donc $\sigma_b(G) \geq \alpha(G)$ et donc

$$\bar{\chi}_b(G) = \bar{\chi}(G) \geq \sigma_b(G) \geq \alpha(G)$$

Si $\alpha(G) = \bar{\chi}(G)$, un α -stable de G et une $\bar{\chi}$ -partition en cliques de G sont une preuve de l'égalité dans BT-mod. \square

3.2.2 Les graphes bornés-parfaits

Un graphe est dit **borné-parfait** si tous ses sous-graphes induits satisfont BT-mod avec égalité pour tout b . Une motivation pour demander l'égalité pour tout

sous-graphe vient de l'écueil suivant⁴ :

Proposition 3.14 *Les problèmes suivants sont NP-durs :*

- Données : (G, b) . Résultat : Est-ce que $\bar{\chi}_b(G) = \sigma_b(G)$?
- Données : (G, b, t) . Promesse : $\bar{\chi}_b(G) = \sigma_b(G)$. Résultat : $\bar{\chi}_b(G) \leq t$?
- Données : (G, b, t) . Promesse : $\bar{\chi}_b(G) = \sigma_b(G)$. Résultat : $\sigma_b(G) \geq t$?

Preuve Dans le cas particulier où $b \geq |V|$, les trois problèmes correspondent respectivement à “Reconnaître les graphes pour lesquels $\bar{\chi} = \alpha$ ”, “Partitionner en cliques les graphes pour lesquels $\bar{\chi} = \alpha$ ” et “Trouver un stable max dans les graphes pour lesquels $\bar{\chi} = \alpha$ ”. Chacun de ces problèmes est NP-dur comme détaillé dans la Proposition 2.5, page 102. \square

Nous souhaitons mettre en garde contre l'éventualité d'un cul de sac conceptuel : la définition des graphes bornés-parfaits est simplement une tentative pour contourner l'écueil de la Proposition 3.14. Cependant, à l'inverse des graphes parfaits, nous ne connaissons pas de formulation polyédrale pour le problème de la coloration bornée dans laquelle le passage au sous-graphe induit ait une interprétation naturelle. Il est donc sage d'envisager qu'une définition basée sur un autre renforcement de la condition $\bar{\chi}_b(G) = \sigma_b(G)$ mène à une classe de graphes plus intéressante. Après le scepticisme polyédral vient le scepticisme algorithmique : les deux problèmes de bases concernant les graphes borné-parfaits (optimisation et reconnaissance) restent ouverts à la clôture de ce mémoire :

Partition en b -cliques des graphes bornés-parfaits

Données : Un graphe G , deux entiers b et k .

Promesse : G est borné parfait

Résultat : Est-ce que $\bar{\chi}_b(G) \leq k$?

Commentaires : $\text{NPP} \cap \text{co-NPP}(\{(G, k, t) \mid \bar{\chi}_b(G) \leq k\})$ est une réalisation NP et $\{(G, k, t) \mid \sigma_b(G) \geq k\}$ est une réalisation co-NP). L'existence d'une réalisation $\text{NP} \cap \text{co-NP}$ est ouverte.

[BP] Borne perfection

Données : Un graphe G

Résultat : Est-ce que $\bar{\chi}_b(G[U]) = \sigma_b(G[U])$, pour tout $b \in \mathbb{N}$ et tout $U \subseteq V$?

Commentaires : Ouvert. Pas connu pour être dans NP ni dans co-NP.

Il existe cependant un argument de poids pour justifier la pertinence et la profondeur de la définition des graphes borné-parfaits. Cela est basé sur un phénomène analogue à celui qui avait motivé Claude Berge lors de la définition des graphes parfaits : cette classe contient plusieurs classes fondamentales dans lesquelles le

⁴Une fois que l'on a demandé l'égalité $\bar{\chi}_b = \sigma_b$ pour tout sous-graphe induit, la Conjecture 3.23 donne une prise sur la pertinence de demander ou non l'égalité pour tout b .

problème [PCliqB] est traitable de manière élégante (Théorème 3.15 et Conjecture 3.16). L'analogie avec les idées de Claude Berge se prolonge car la classe des graphes borné-parfaits semble jouir de propriétés surprenantes qui rappellent les fameuses conjectures de Berge pour les graphes parfaits (Conjectures 3.23 et 3.28).

Théorème 3.15 *Si G appartient à l'une des classes suivantes, alors il est borné-parfait. De plus, [PCliqB] est polynomial dans chacune de ces classes.*

- i) *Triangulés*
 - *Graphes d'intervalles [76]*
 - *Split graphes*
- ii) *Parfaits co-sans- $K_{1,3}$ [55, 93, 79]*
 - *Co-sans- P_3*
 - *Compléments de line-graphes de multi-graphes bipartis.*
- iii) *Parfaits sans- K_4 [76, 93]*
 - *Bipartis*

Preuve i) Voir Section 3.3.

ii) Si G est co-sans- $K_{1,3}$, d'après le Théorème 3.6, on a $\bar{\chi}_b(G) = \max\{\bar{\chi}(G), \lceil |V|/b \rceil\}$. Si G est parfait, on a $\bar{\chi}(G) = \alpha(G)$. Donc si G est parfait co-sans- $K_{1,3}$, on a

$$(3.5) \quad \bar{\chi}_b(G) = \max\{\alpha(G), \lceil |V|/b \rceil\}$$

Si (3.5) est serrée, BT-mod l'est aussi (un ensemble S qui réalise $\sigma_b(G)$ est donné soit par un α -stable soit par V lui-même). Pour la polynomialité, comme G est parfait, on peut trouver une partition en cliques optimum. On applique ensuite le Corollaire 3.7 pour trouver une $\bar{\chi}_b(G)$ partition en b -cliques.

iii) Si $b = 1$, c'est trivial. Sinon, si $b = 2$, on applique le Théorème 3.13 i). Si $b \geq 3$, on applique le Théorème 3.13 ii) □

Une généralisation commune de la borne perfection des graphes triangulés, bipartis et co-sans- P_3 serait :

Conjecture 3.16 *Les graphes de Gallai sont bornés-parfaits.*

Observons maintenant quelques cas de graphes qui ne sont pas bornés-parfaits. Un graphe est **minimalement non-borné-parfait** s'il n'est pas borné-parfait, mais si tous ses sous-graphes induits le sont.

Proposition 3.17 *Les trous impairs et leurs compléments sont minimaux non-bornés-parfaits.*

Preuve Pour voir que C_{2n+5} et $\overline{C_{2n+5}}$ ne sont pas bornés-parfaits, il suffit de considérer $b = 2n + 5$. La contrainte de taille est alors superflue et on a $\bar{\chi}_b(G) = \bar{\chi}(G) > \alpha(G) = \sigma_b(G)$.

Les trous impairs et leur compléments sont minimalement non-bornés-parfaits car en enlevant un sommet, on obtient soit une chaîne soit le complément d'une chaîne qui sont toutes deux parfaites et co-sans- $K_{1,3}$. □

La Figure 3.1 donne des exemples de graphes minimaux non-bornés-parfaits, ce qui peut être vérifié “à la main” et qui est discuté de manière plus générale et détaillée dans la Proposition 3.30. Pour ces graphes, le fait que $\bar{\chi}_3(G) > \sigma_3(G)$ peut aussi être prouvé par des arguments de dualité appliqués aux programmes linéaires présentés dans la Section 3.4.3.

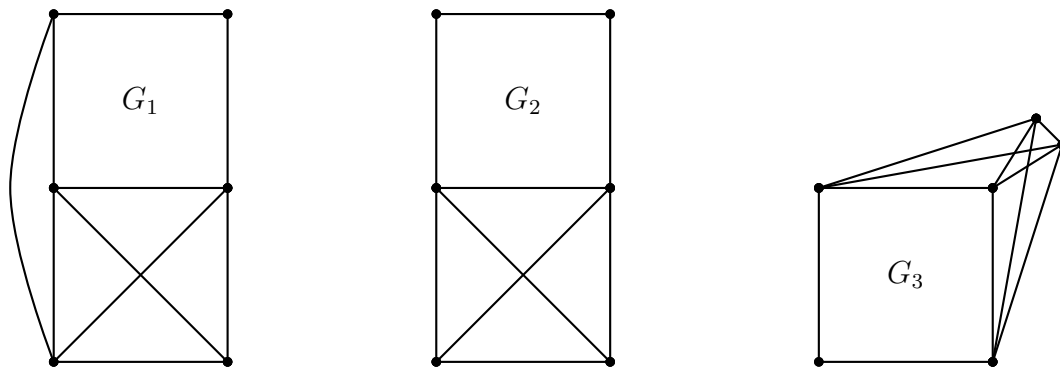


Figure 3.1 – Trois graphes pour lesquels $3 = \bar{\chi}_3(G_i) > \sigma_3(G_i) = 2$.

Grâce au Théorème 3.15, nous savons que les parfaits minimaux non-bornés-parfaits contiennent à la fois une clique de taille au moins 4, un trou (pair) et un $\text{co-}K_{1,3}$, (ce que confirme la Figure 3.1). Montrons de plus que ces trois structures doivent être imbriquées dans un graphe 2-sommet connexe.

Remarque 3.18 Si G est l’union disjointe des graphes $G_1, G_2 \dots G_t$, alors $\bar{\chi}_b(G) = \sum_i \bar{\chi}_b(G_i)$ et $\sigma_b(G) = \sum_i \sigma_b(G_i)$. \square

Remarque 3.19 Un graphe est borné parfait si et seulement si les graphes induits par chacune de ses composantes connexes le sont. \square

Proposition 3.20 Soit G un graphe ayant un sommet d’articulation v . Soient $C_1, C_2 \dots C_t$ les composantes connexes de $G - v$ et soit $G_i = G[C_i + v]$, pour $i = 1, 2 \dots t$. Alors G est borné-parfait si et seulement si tous les G_i sont borné-parfaits.

Preuve Si l’un des G_i n’est pas borné-parfait, alors G ne l’est pas non plus car G_i est un sous-graphe induit de G .

Supposons maintenant que G_i soit borné-parfait pour tout $i = 1 \dots t$ et fixons $b \in \mathbb{N}$. Notons que

$$\sum_i \bar{\chi}_b(G_i - v) = \bar{\chi}_b(G - v) \leq \bar{\chi}_b(G) \leq 1 + \sum_i \bar{\chi}_b(G_i - v)$$

et étudions les deux cas ainsi suggérés.

Cas 1) Si $\bar{\chi}_b(G) = \bar{\chi}_b(G - v)$, on a

$$\bar{\chi}_b(G) = \bar{\chi}_b(G - v) = \sum \bar{\chi}_b(G_i - v) = \sum_i \sigma_b(G_i - v) = \sigma_b(G - v) \leq \sigma_b(G)$$

et donc $\bar{\chi}_b(G) = \sigma_b(G)$ d'après la borne BT-mod.

Cas 2) Si $\bar{\chi}_b(G) = 1 + \bar{\chi}_b(G - v)$. Soit S_i un sous-ensemble minimal qui définit $\sigma_b(G_i)$. Montrons que l'union des S_i définit $\sigma_b(G)$. Puisque $\bar{\chi}_b(G) = 1 + \bar{\chi}_b(G - v)$, on a $\bar{\chi}_b(G_i) = 1 + \bar{\chi}_b(G_i - v)$ pour tout i et donc, par borne-perfection de G_i , que $\sigma_b(G_i) = 1 + \sigma_b(G_i - v)$. On a donc $v \in S_i$ pour tout i . Par minimalité, chacune des composantes connexes de chacun des S_i a une cardinalité congrue à 1 modulo b . Appelons D_i l'unique composante de S_i qui contient v . Les composantes connexes de $G[S_i]$ qui ne contiennent pas v sont aussi des composantes connexes de $G[\cup S_i]$ et l'union des D_i forme une unique composante dans $G[\cup S_i]$ de cardinalité congrue à 1 modulo b . On a donc $\sigma_b(G[\cup D_i]) = 1 + \sigma_b(G[\cup(D_i - v)])$ et donc $\sigma_b(G[\cup S_i]) = 1 + \sigma_b(G[\cup(S_i - v)])$. Finalement :

$$\begin{aligned} \sigma_b(G[\cup S_i]) &= 1 + \sigma_b(G[\cup(S_i - v)]) = 1 + \sum \sigma_b(G[S_i - v]) \\ &= 1 + \sum \bar{\chi}_b(G_i - v) = 1 + \bar{\chi}_b(G - v) = \bar{\chi}_b(G) \end{aligned}$$

□

Corollaire 3.21 *La borne-perfection est préservée par sommet d'articulation.*

Corollaire 3.22 *Les minimaux non-bornés-parfaits sont 2-sommets connexes.*

Il semble (en dehors des anti-trous impairs de taille ≥ 9), que les minimaux non-bornés-parfaits souffrent tous de non-égalité dans (BT-mod) déjà pour $b = 3$:

Conjecture 3.23 (Faible des graphes bornés-parfaits) *Un graphe est borné-parfait si et seulement s'il est parfait et $\bar{\chi}_3(G[U]) = \sigma_3(G[U])$ pour tout $U \subseteq V$.*

Nous avons fait une autre observation empirique qui, en conséquence de la Proposition 3.13 ii), est au moins aussi forte que la Conjecture 3.23.

Conjecture 3.24 *Si G est parfait et minimal non-borné-parfait, alors $\omega(G) = 4$.*

Dans la Section 3.2.4, nous discutons une conjecture encore plus forte qui décrit les graphes bornés-parfaits par sous-graphes induits exclus.

3.2.3 Borne-perfection, réplification et co-réplification

Dans la Section 3.2.2, par analogie avec les graphes parfaits, nous avons défini les graphes bornés-parfaits, qui satisfont BT-mod pour tout sous-graphe induit et tout b . Mais les graphes parfaits sont non-seulement stables par passage à l'induit mais aussi par réplification. Ce n'est pas le cas des bornés-parfaits comme le montrent

les graphes G_2 et G_3 de la Figure 3.1. En effet, G_2 est obtenu en dupliquant le “toit” de la “maison” \overline{P}_5 , et G_3 peut être obtenu soit en répliquant deux fois un sommet de C_4 , soit en co-dupliquant le sommet d’articulation d’une raquette (voir Figure 3.2 page 142). Dans cette section, nous caractérisons les graphes triangulés comme les graphes qui restent bornés-parfaits quand on les réplique et nous conjecturons une caractérisation similaire pour la co-réplication des graphes de Berge sans-raquette. De même que pour le passage à l’induit, nous ne connaissons pas d’interprétation polyédrale de la (co-)réplication. Cependant, de même que pour la coloration, la (co-)réplication admet une interprétation applicative intéressante dans le cadre de la coloration bornée. Cela est discuté dans la Section 3.4.4 en même temps que les questions algorithmiques.

3.2.3.1 Une nouvelle caractérisation des graphes triangulés

Les graphes qui restent bornés-parfaits par réplication sont exactement les graphes triangulés :

Théorème 3.25 *Soit G un graphe. Les conditions suivantes sont équivalentes*

- i) G est triangulé.*
- ii) Il existe $b \geq 3$ tel que $\overline{\chi}_b(H) = \sigma_b(H)$ pour tout graphe H obtenu par réplication à partir de G .*
- iii) $\overline{\chi}_b(H) = \sigma_b(H)$ pour tout graphe H obtenu par réplication à partir de G et tout entier b non-nul.*

Preuve Nous montrons dans la Section 3.3.3 que les graphes triangulés sont bornés parfaits.

i) \implies iii) vient du fait que la classe des graphes triangulés est fermée par réplication.

iii) \implies ii) est trivial.

ii) \implies i) Soit G un graphe non-triangulé et $b \geq 3$. G a un cycle C_k comme sous-graphe induit (avec $k \geq 4$). Nous construisons un graphe H répliqué de G , tel que $\overline{\chi}_b(G) > \sigma_b(G)$.

i) Si $k = 2q + 1$, on prend $H = C_k$

ii) Si $k = 2q$, on définit H à partir de C_k en répliquant un sommet avec un facteur b (voir G_3 dans la Figure 3.1 page 139 et G dans la Figure 3.19 page 171).

Il est facile en utilisant la symétrie de ces graphes de montrer que, dans les deux cas $\overline{\chi}_b(H) = q + 1 > q = \alpha(H) = \sigma_b(H)$. On peut aussi le voir en utilisant la formulation (3.11) page 170, comme illustré dans la Figure 3.19 page 171. \square

3.2.3.2 Une conjecture pour les graphes de Berge sans-raquette

Observons que parmi les trois principales classes de graphes bornés-parfaits découvertes jusqu'à présent, une est stable par réplification (les triangulés) et deux sont stables par co-réplification. Autrement dit :

Remarque 3.26 *Si G est parfait co-sans- $K_{1,3}$ ou parfait sans- K_4 alors G est borné-parfait pour tout graphe H obtenu par co-réplification de G .*

Preuve La perfection, le fait d'être co-sans- $K_{1,3}$ et le fait d'être sans- K_4 sont trois propriétés préservées par co-réplification. \square

La *raquette* est le graphe à 5 sommets, tel que l'un des sommet a degré 1 et les 4 autres induisent un K_4 (voir Figure 3.2).

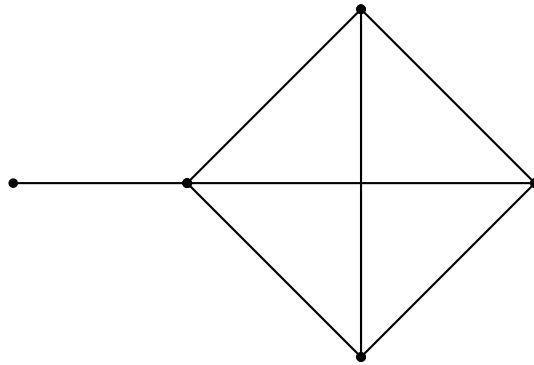


Figure 3.2 – La raquette.

La remarque 3.26 est liée à une observation empirique : les graphes de Berge sans-raquette sont bornés-parfaits. Puisque la classe des graphes (de Berge) sans-raquette est stable par co-réplification, il est équivalent de conjecturer leur borne-perfection ou de conjecturer la caractérisation suivante :

Conjecture 3.27 *Les conditions suivantes sont équivalentes*

- i) G est parfait et sans-raquette.*
- ii) G est parfait et $\bar{\chi}_3(H) = \sigma_3(H)$ pour tout graphe H obtenu par co-réplification à partir de G .*
- iii) $\bar{\chi}_b(H) = \sigma_b(H)$ pour tout graphe H obtenu par co-réplification à partir de G et tout entier b non-nul.*

Cette conjecture peut être déclinée en une conjecture forte : l'implication i) \implies iii) ainsi qu'une conjecture plus faible : l'implication i) \implies ii). Les autres implications sont évidentes : iii) \implies ii) est trivial et ii) \implies i) vient du fait que le graphe G_3 de la Figure 3.1 est obtenu en co-dupliquant le sommet d'articulation d'une raquette.

3.2.4 Une conjecture forte pour les bornés-parfaits

Nous conjecturons que les minimaux non-bornés-parfaits sont les trous impairs, leurs compléments et les graphes obtenus en remplaçant le C_4 d'un des graphes de la Figure 3.1 par un cycle pair. Cette conjecture implique les autres conjectures concernant les graphes bornés-parfaits présentées dans les sections précédentes. Nous discutons aussi des arguments qui ont mené à la formulation de cette conjecture et qui donnent des éléments de preuve.

Une *antenne* (voir Figure 3.3 pour une autre définition) est un graphe $G = (V, E)$ tel que

- i) Il existe $s, t \in V$ tels que $G - \{s, t\}$ est un cycle pair C_{2k}
- ii) Il existe $a, b \in V(C_{2k})$ tels que $G[a, b, s, t] = K_4$.
- iii) Le voisinage de s est inclus dans le voisinage de a .
- iv) Le voisinage de t est inclus dans le voisinage de a .

Notons que les rôles de s et t dans la définition des antennes sont symétriques.

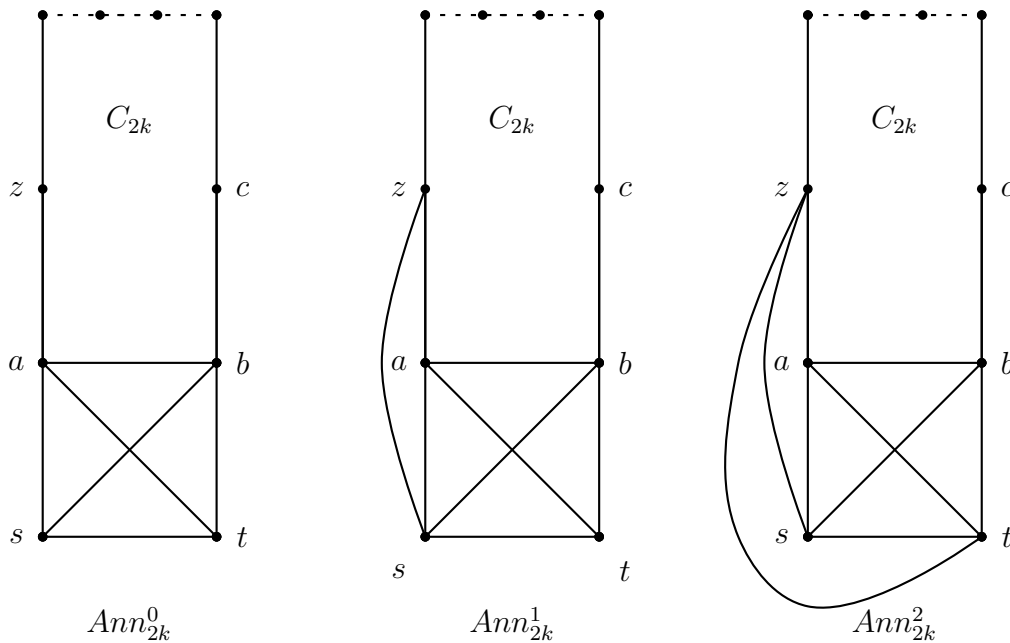


Figure 3.3 – A chaque cycle pair, on peut associer 3 antennes. Il existe donc trois classes d'antennes : Ann_{2k}^0 , Ann_{2k}^1 et Ann_{2k}^2 .

Conjecture 3.28 (Forte des graphes bornés-parfaits) *Les conditions suivantes sont équivalentes*

- i) G est parfait et sans antenne.
- ii) G est parfait et $\bar{\chi}_3(H) = \sigma_3(H)$ pour tout sous-graphe induit H de G .
- iii) $\bar{\chi}_b(H) = \sigma_b(H)$ pour tout sous-graphe induit H de G et tout entier b non-nul.

La première raison d'être de la Conjecture 3.28 est évidemment :

Proposition 3.29 *Les antennes sont des graphes minimalement non-bornés-parfaits.*

Preuve Montrons que pour tout $i \in \{0, 1, 2\}$ et tout $k \geq 2$, on a

$$k + 1 = \bar{\chi}_3(\text{Ann}_{2k}^i) > \sigma_3(\text{Ann}_{2k}^i) = k$$

$k + 1 \geq \bar{\chi}_3(\text{Ann}_{2k}^i)$ est donné par un couplage parfait contenant (ab) comme arête, tandis que $\sigma_3(\text{Ann}_{2k}^i) \geq k$ est donné par un stable de taille k de C_{2k} . L'inégalité $\bar{\chi}_3(\text{Ann}_{2k}^i) > \sigma_3(\text{Ann}_{2k}^i)$ peut être vue "à la main" mais est aussi prouvée dans la Section 3.4.3 comme conséquence de la dualité faible de la programmation linéaire. Ann_{2k}^i est minimal non-borné-parfait car pour $v \in \{a, s, t\}$, on a $\omega(\text{Ann}_{2k}^i - v) = 3$ et on a donc un graphe de Berge-sans- K_4 qui est borné-parfait d'après le Théorème 3.15 et pour $v \notin \{a, s, t\}$, le graphe $\text{Ann}_{2k}^i - v$ est triangulé et donc borné-parfait d'après le Théorème 3.15. \square

La deuxième raison d'être de la Conjecture 3.28 est la Proposition 3.30 qui montre que l'on ne peut pas construire d'autres graphes minimalement non-bornés-parfaits à partir d'une antenne en lui ajoutant des arêtes (mis-à-part en passant de Ann_{2k}^0 à Ann_{2k}^1 et de Ann_{2k}^1 à Ann_{2k}^2).

Proposition 3.30 *Soit $G = (V, E)$ un graphe et $s, t \in V$ tel que $G - \{s, t\}$ est un cycle pair C_{2k} . Alors G est minimalement non-borné-parfait si et seulement si c est une antenne.*

Preuve Soit $G = (V, E)$ un graphe minimalement non-borné-parfait avec $s, t \in V$ tel que $G - \{s, t\}$ est un cycle pair C_{2k} . Alors

Fait 1) $\omega(G) = 4$ et il existe a, b voisins sur C_{2k} tels que $G[a, b, s, t] = K_4$.

Fait 2) $\bar{\chi}_3(G) = k + 1$ et $\sigma_3(G) = k$

Soit $z, c \in V(C_{2k})$ tels que $za, bc \in E(C_{2k})$.

Fait 3) s n'est pas adjacent à la fois à z et c .

De même t n'est pas adjacent à la fois à z et c . De plus, quitte à échanger les rôles de z et c , on a

Fait 4) c n'est adjacent ni à s ni à t .

Nous donnons maintenant la fin de la preuve de la Proposition 3.30 avant d'expliquer les Faits ci-dessus. Si G n'est pas Ann_{2k} , appelons x le premier sommet de $N(s) \cup N(t)$ rencontré en parcourant le chemin $V(C_{2k}) - \{a\}$ de c vers z . Si $x = z$, alors G est une antenne et est donc minimalement non-borné-parfait d'après la Proposition 3.29. Si $x \neq z$, soit Q le sous-chemin induit entre b et x (inclus) contenant c . Si Q a un nombre pair de sommets, l'un de $G[Q + s]$ ou $G[Q + t]$ est un trou impair (car d'après le Fait 4, $|Q| \geq 3$), ce qui contredit la minimalité de G . Si Q a un nombre impair de sommets, le graphe $G[Q + s + t + a + b]$ est soit isomorphe à $\text{Ann}'_{|Q|}$ (si $sx \in E$ et $tx \in E$), soit isomorphe à $\text{Ann}_{|Q|}$ (si exactement l'une des arêtes sx et tx est présente dans E).

Preuve du Fait 1) Puisque G n'est pas un trou ni un antitrou impair, G est parfait. D'après le Théorème 3.15 iii), puisque G n'est pas borné-parfait, G contient un K_4 . Puisque $\omega(C_{2k}) = 2$, on a aussi $\omega(G) \leq 4$. \square

Preuve du Fait 2) Dans la chaîne suivante,

$$(3.6) \quad k + 1 \geq \bar{\chi}_3(G) > \sigma_3(G) \geq \alpha(G) \geq \alpha(C_{2k}) = k$$

la première inégalité vient en couvrant $V + s + t$ par un couplage parfait dont (s, t) forme une arête. La deuxième inégalité vient du fait que G est non-borné-parfait, et donc que $\bar{\chi}_b(G) > \sigma_b(G)$ pour un certain b , mais que, comme G est parfait, on a $k = \bar{\chi}_b(G) \geq \sigma_b(G) = \alpha(G) = k$ pour $b \geq 4 = \omega(G)$. \square

Preuve du Fait 3) Si s est adjacent à la fois à z et c , alors $G - \{t, a, b\}$ est un cycle impair dont toutes les éventuelles cordes partent de s . Appelons *segment*, un sous-chemin S de longueur au moins 1 du chemin $P := G - \{s, t, a, b\}$ tel que $G[S + s]$ est un circuit. Le chemin P est partitionné en segments. Par parité de P , il existe un segment S tel que $G[S + s]$ est un circuit impair. Soit S^* le premier tel segment rencontré en allant de z à c . Si $|S^*| \geq 5$, G n'est pas minimalement non-borné-parfait, contradiction. Si $|S^*| = 3$, alors $\bar{\chi}_3(G) \leq k$ car on peut partitionner G en deux triangles (S^* et $\{a, b, t\}$) et en arêtes, ce qui contredit le Fait 2. \square

Preuve du Fait 4) D'après le Fait 3, s et t sont tous deux adjacents à au plus un sommet de l'ensemble $\{z, c\}$. Montrons donc qu'ils ne peuvent être adjacents, l'un à c et l'autre à z . S.p.d.g supposons que $sz \in E$ et $tc \in E$. Alors $\bar{\chi}_3(G) = k$ car on peut partitionner $V(G)$ en deux triangles ($\{z, a, s\}$ et $\{b, c, t\}$) et en arêtes, ce qui contredit le Fait 2. \square

Cela termine la preuve de la Proposition 3.30. \square

La Conjecture 3.28 implique les autres conjectures formulées précédemment :

Corollaire 3.31 *La Conjecture 3.28 implique*

- i) la Conjecture 3.24 ($\omega(G) = 4$ si G est parfait et minimal non-borné-parfait).
- ii) la Conjecture 3.27 (borne-perfection des graphes de Berge-sans-raquette).
- iii) la Conjecture 3.16 (borne-perfection des graphes de Gallai).

Preuve i) Dans une antenne, on a $\omega(G) = 4$.

ii) Les antennes contiennent toutes une raquette.

iii)

On utilise le fait suivant :

Proposition 3.32 [35] *Soit $G = (X, E)$ un graphe de Gallai et $V \subseteq X$ tel que $G[V]$ soit un graphe biparti 2-connexé. Alors tous les $x \in X - V$ satisfont l'un des énoncés suivants :*

- i) $N(x) \cap V = V$

ii) $G[V + x]$ est un graphe biparti

Si un graphe G contient une antenne, alors G contient un C_{2k} . De plus dans une antenne, le sommet s (ou t) n'est pas adjacent à tous les sommets de C_{2k} mais induit un triangle avec eux. D'après la Proposition 3.32, un graphe de Gallai est donc sans antenne. \square

3.3 Algorithmes pour les graphes triangulés

3.3.1 Glouton pour les graphes d'intervalles

Pour les intervalles on peut construire une partition optimale en b -cliques de manière gloutonne, ce qui est un avantage à la fois pédagogique et algorithmique⁵.

La polynomialité de [PCliqB] pour les graphes d'intervalles peut être prouvée à partir d'un résultat de [125] : le problème d'ordonnancement sur machines parallèles avec durées unitaires et contraintes de précédence $P|p_j = 1, prec|C_{max}$ est polynomial si on restreint aux "ordres d'intervalles", autrement dit, aux contraintes de précédence dont le graphe de comparabilité est co-triangulé. La réduction de [ColB] pour les co-intervalles à ce problème d'ordonnancement a été observée dans [22, 113]. Cette réduction est expliquée dans la Section 2.2.2.7, où une caractérisation des graphes d'intervalles est donnée en termes d'équivalence des contraintes de précédence et de contraintes d'incompatibilité.

Nous discutons un algorithme légèrement différent de celui de [125]⁶ que nous avons découvert avant de réaliser que ce problème était déjà résolu. L'algorithme suivant – que nous appelons GAC (Algorithme Glouton avec Compatibilité) – construit une partition en b -cliques d'un graphe d'intervalle, où le graphe est donné par une représentation en intervalles. Dans cet algorithme, les batchs sont successivement créés dans l'ordre B_1, \dots, B_ℓ , (ℓ est un entier positif). A chaque étape de l'algorithme, un batch B_i est appelé *insaturé* si le nombre de sommets qu'il contient est plus petit que b (i.e. $|B_i| < b$). Sinon, B_i est *saturé*.

Considérons les intervalles I_1, \dots, I_n dans un ordre non-décroissant de leur points de terminaison b_i (voir Figure 3.4). Les termes 'premier', 'dernier', 'avant', 'après' réfèrent à cet ordre. L'algorithme suivant finit de s'occuper d'un batch avant de commencer à en remplir un autre. De plus, à l'intérieur de chaque batch, il considère les sommets dans l'ordre $1, 2, \dots, n$ sans revenir en arrière. C'est pourquoi il est appelé "glouton".

⁵Les résultats et preuves de la Section 3.3.1 sont issus d'une collaboration avec G. Finke, M. Queyranne et A. Sebő [76].

⁶L'algorithme de [125] est de complexité $\mathcal{O}(n + m)$ alors que le notre est en $\mathcal{O}(n \log(n))$.

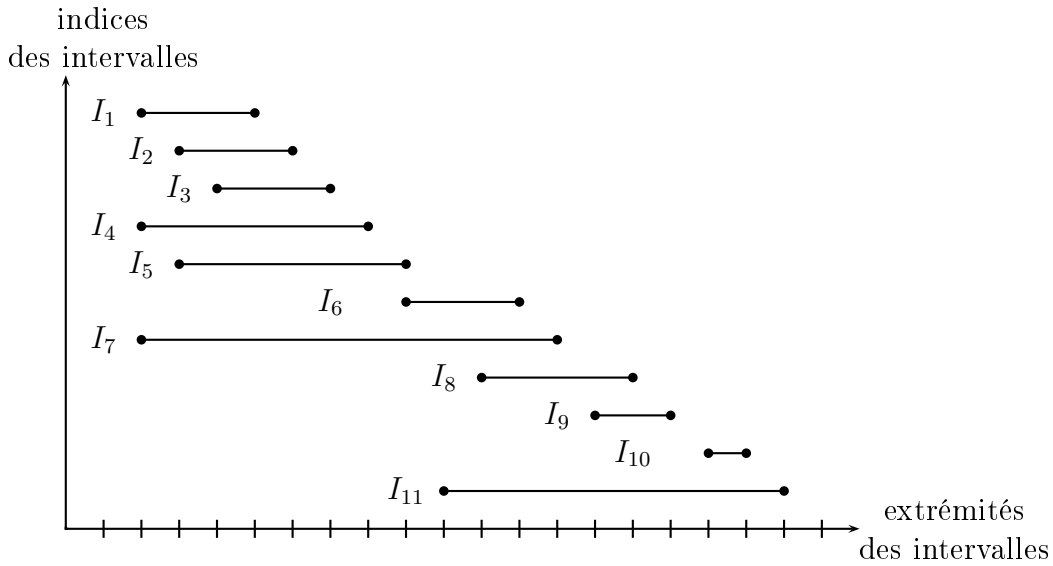


Figure 3.4 – Représentation ordonnée par b_i croissants d'un graphe d'intervalle G

L'algorithme GAC : Construire un batch par itération jusqu'à ce que tous les intervalles soient placés dans des batchs. Au cours de la i -ème itération ($i = 1, \dots, l$), on ouvre un nouveau batch B_i et on lui affecte le premier intervalle $I_j = [a_j, b_j]$ qui n'a pas encore été placé dans un batch. Après avoir placé I_j dans B_i , on affecte les b premiers intervalles intersectant b_j (ou tous, s'il y en a moins que b).

Appelons *étiquette* de B_i , le premier intervalle placé dans le batch B_i . Les points terminaux des étiquettes sont non-décroissants par construction.

Notons aussi que l'algorithme suivant fournit la même solution et montre que GAC peut être vu comme une version de *first-fit*, algorithme classique pour le problème de bin-packing [96], modifié pour prendre en compte les contraintes de compatibilité :

L'algorithme GAC' : On commence avec aucun batch. On considère les intervalles un par un dans l'ordre ($j = 1, \dots, n$). On insère l'intervalle I_j dans le batch insaturé B_i , ayant le plus petit indice i , et qui est compatible avec I_j . S'il n'y a pas de tel batch (insaturé et compatible), on en crée un nouveau et I_j est mis dedans.

La solution de GAC est illustrée sur un exemple dans la Figure 3.5.

Proposition 3.33 [76] Pour tout graphe d'intervalles G et tout $b \in \mathbb{N}$, GAC donne une $\bar{\chi}_b(G)$ partition en b -cliques de G . GAC' a une complexité en temps $\mathcal{O}(n \log(n))$.

Preuve On procède par induction sur le nombre d'intervalles dans la séquence B_1, B_2, \dots, B_l de batchs construite par GAC. Puisque l'algorithme procède de manière gloutonne, la séquence B_2, \dots, B_l est sélectionnée par GAC si on met le graphe

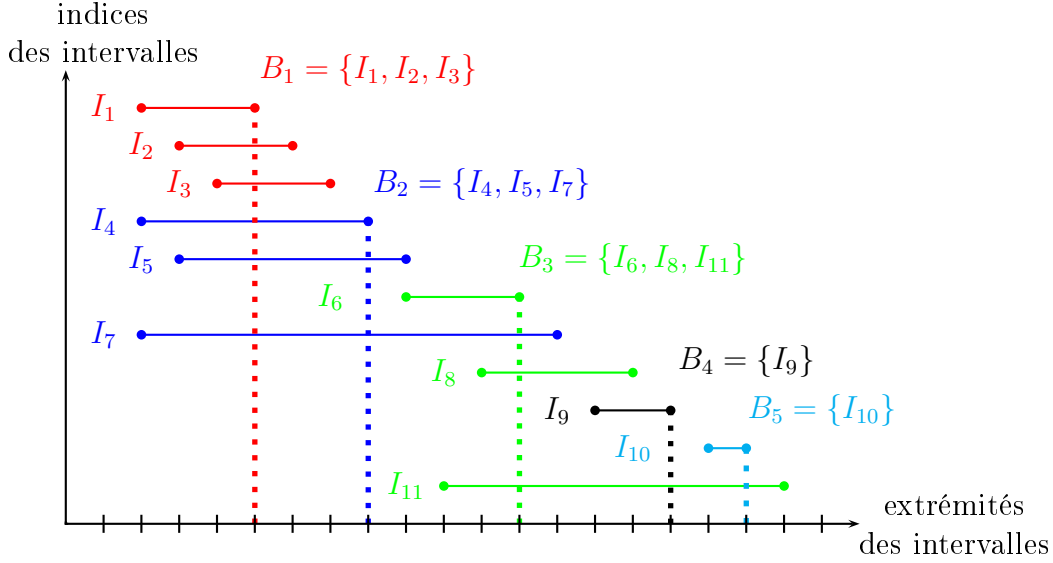


Figure 3.5 – Solution optimale fournie par GAC pour $\bar{\chi}_3(G)$.

$G[V - B_1]$ en entrée. Il est donc suffisant de montrer qu'il existe une partition \mathcal{B} de G en b -cliques, telle que $|\mathcal{B}| = \bar{\chi}_b(G)$ et telle que $B_1 \in \mathcal{B}$.

L'étiquette de B_1 est $I_1 = [a_1, b_1]$. Rappelons que I_1 est un sommet simplicial de G . Soit D_1 le batch contenant I_1 dans une partition optimale \mathcal{D} , telle que $|D_1 \cap B_1|$ soit maximum parmi l'ensemble des \mathcal{D} possibles. Montrons $D_1 = B_1$. Puisque $I_1 \in D_1$, tous les intervalles dans D_1 contiennent b_1 .

Si $|B_1| < b$, c'est qu'il y a moins que b intervalles compatibles avec I_1 . GAC a donc mis exactement les voisins de I_1 dans B_1 . Puisque D_1 est une clique contenant I_1 , $D_1 \subseteq B_1$. Par monotonie de $\bar{\chi}_b$ on a que

$$|\mathcal{B}| - 1 = \bar{\chi}_b(G - B_1) \leq \bar{\chi}_b(G - D_1) = |\mathcal{D}| - 1,$$

qui prouve l'optimalité de \mathcal{B} (si $|B_1| < b$).

Si $|B_1| = b$. Supposons, par contradiction, qu'il existe $I_j \in B_1 \setminus D_1$. Par monotonie de $\bar{\chi}_b$, $D_1 \subseteq B_1$ peut être exclu. Il existe donc $I_k = [a_k, b_k] \in D_1 \setminus B_1$. Puisque B_1 consiste en les premiers b intervalles de G , on a $j < k$, et donc $b_j \leq b_k$. Puisque $I_k \in D_1$, on a $b_1 \in I_k$. Effectuons l'échange suivant : d'une part $D'_1 := (D_1 \setminus \{I_k\}) \cup \{I_j\}$ et d'autre part, pour $D \in \mathcal{D}$ contenant I_j , $D' := (D \setminus \{I_j\}) \cup \{I_k\}$. Les batches ainsi redéfinis satisfont les contraintes de compatibilité. Pour D'_1 , on a $b_1 \in I_j$, et b_1 est aussi contenu dans tous les intervalles de D_1 . Pour D' , on a aussi que I_k rencontre tous les intervalles que I_j rencontre (on a que $b_j \leq b_k$, que $b_1 \in I_k$, et que tous les points de terminaisons sont au moins aussi grands que b_1).

Mais alors $|D'_1 \cap B_1| > |D_1 \cap B_1|$, ce qui contredit la définition de \mathcal{D} , et termine la preuve. \square

Construisons maintenant une solution duale, c'est-à-dire un sous-ensemble U définissant $\sigma_b(G)$. L'algorithme est glouton et traite les sommets dans l'ordre I_n, \dots, I_1 inverse de GAC. On construit une chaîne décroissante de sous-ensembles de V tels que $\bar{\chi}_b$ ne décroît pas.

L'algorithme DualGAC : *Ayant stocké la solution B_1, \dots, B_ℓ fournie par GAC, considérer les intervalles I_n, \dots, I_1 , un par un. Si I_i intersecte un intervalle I_j ($i \neq j$), étiquette d'un batch insaturé, redéfinir $G := G[V - I_i]$ et redéfinir les batchs de G comme les batchs induits sur le graphe $G[V - I_i]$.*

Montrons d'abord comment tourne cet algorithme sur l'exemple de la Figure 3.5. Le résultat est représenté dans la Figure 3.6 : l'intervalle I_{11} est enlevé puisqu'il intersecte l'étiquette du batch insaturé B_5 . I_8 est enlevé parcequ'il intersecte I_6 , étiquette du batch B_3 qui est maintenant insaturé. I_8 est aussi enlevé parcequ'il intersecte I_9 , étiquette du batch B_4 . Finalement, on obtient l'ensemble $U = \{I_1, I_2, I_3, I_4, I_6, I_9, I_{10}\}$ qui définit $\sigma_3(G)$.

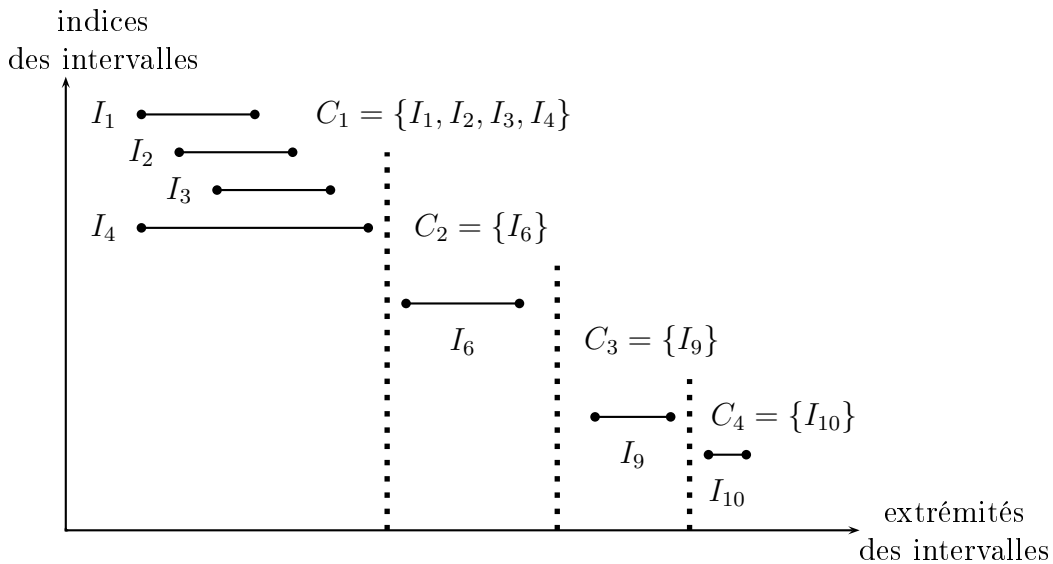


Figure 3.6 – Solution définissant $\sigma_3(G)$, fournie par DualGAC pour l'ensemble d'intervalles de la Figure 3.5

Proposition 3.34 *Pour tout graphe d'intervalle G et tout $b \geq 2$ on a $\sigma_b(G) = \bar{\chi}_b(G)$. DualGAC construit un sous-ensemble U qui vérifie*

$$\bar{\chi}_b(G) = \sum_i \left\lceil \frac{C_i(G[U])}{b} \right\rceil$$

DualGAC peut être implémenté en temps $\mathcal{O}(n \log(n))$.

Preuve On sait déjà que $\bar{\chi}_b(G) \geq \sigma_b(G)$. Montrons l'autre sens par induction sur

n :

C'est évident si $n = 1$. Supposons que ça soit vrai pour tous les graphes ayant strictement moins de sommets que G . Puisque $\bar{\chi}_b$ et σ_b sont toutes deux additives sur les composantes connexes, on peut supposer que G est connexe. Si il existe $v \in V(G)$ tel que $\bar{\chi}_b(G - v) = \bar{\chi}_b(G)$ on a fini car par hypothèse de récurrence :

$$\bar{\chi}_b(G) = \bar{\chi}_b(G - v) = \sigma_b(G - v) \leq \sigma_b(G)$$

On peut donc supposer que $\bar{\chi}_b(G - v) < \bar{\chi}_b(G)$ pour tout $v \in V(G)$. Sous cette hypothèse, on montre que les batchs définis par GAC *partitionnent* $V(G)$ en batchs de taille b et un unique batch B tel que $|B| = 1$. Si GAC ne renvoie pas au moins un batch insaturé, on a $\bar{\chi}_b(G - v) = \bar{\chi}_b(G)$ pour tout $v \in V(G)$. Soit $B \in \mathcal{B}$ le premier batch insaturé. Si $|B| \geq 2$, soit L le dernier intervalle placé dans B par GAC, et soit I l'étiquette de B . Puisque tous les intervalles placés après L sont disjoints de I , GAC renvoie les même batchs sur $G - L$ que sur G (à part que B a un intervalle de moins). Comme cette solution est optimale (Proposition 3.33), $\bar{\chi}_b(G - L) = \bar{\chi}_b(G)$, contradiction.

Donc $|B| = 1$. Soit t le point terminal de l'unique intervalle $I \in B$. Premièrement, notons que les intervalles placés après I ont un point terminal après t puisqu'ils ne contiennent pas t (sinon ils seraient placés dans B). Leur point initial est donc aussi plus grand que t .

Nous montrons que *les points terminaux des intervalles placés avant I sont avant t et donc que ces intervalles sont disjoints de ceux placés après I* . Sinon, soit J le dernier intervalle qui contient t placé avant I . Alors $\bar{\chi}_b(G - J) = \bar{\chi}_b(G)$, car aucun des intervalles placés dans ces batchs strictement après le batch de J jusqu'à I (I compris) ne peut être placé dans le batch de J : sinon GAC les auraient placés là, à la place de J . La contradiction $\bar{\chi}_b(G - J) = \bar{\chi}_b(G)$ implique que les intervalles placés avant I sont disjoints de ceux placés après.

Les intervalles placés avant I forment donc une composante de G , et puisque G est connexe, tous les intervalles de G y sont. Cela termine la preuve que $\bar{\chi}_b(G) = \sigma_b(G)$.

On peut implémenter DualGac en $O(n^2)$, en regardant, pour chaque intervalle I_j et chaque batch B_i , si I_j intersecte le point de terminaison de l'étiquette de B_i . Pour une implémentation qui traite chaque intervalle en temps $\log(n)$, on garde en mémoire une liste, disons \mathcal{U} , qui contient les batchs insaturés dans l'ordre induit par \mathcal{B} . Lors du traitement de l'intervalle I_j , on peut savoir par dichotomie sur les éléments de \mathcal{U} si I_j intersecte l'étiquette d'au moins un batch insaturé. En effet, s'il existe un tel batch, c'est en particulier le cas du batch insaturé ayant la plus grande étiquette inférieure à j . \square

3.3.2 c -couplages pour les split graphes

[PCliqB] restreint aux splits graphes se réduit au couplage biparti dans un graphe auxiliaire [42, 113], ou au flot max dans un graphe biparti orienté auxiliaire [27, 22]. Les deux réductions prouvent la polynomialité. Notre algorithme pour les triangulés, présenté dans la Section 3.3.3, généralise ces réductions mais implique de nombreuses difficultés supplémentaires. Il nous semble donc pédagogiquement important de montrer ce qui se passe pour les split graphes dans un premier temps.

Soit $G = ((S, K), E)$ un split graphe (où S est un stable et K une clique). Dans toute partition en b -cliques de G , les sommets de S sont dans des batches distincts (et donc $\bar{\chi}_b(G) \geq |S|$). Etant donnée une partition en b -cliques de G , enlevons les sommets qui appartiennent à un batch contenant un sommet de S . Il reste un ensemble $V' \subseteq K$, qui peut être partitionné avec $\lceil |V'|/b \rceil$ batches. [PCliqB] sur les split-graphes se réduit donc à trouver $|S|$ b -cliques contenant les sommets de S et un maximum de sommets de K . Cela s'exprime comme un problème de c -couplage dans le graphe biparti auxiliaire G' qui a (S, K) comme bipartition et dans lequel $s \in S$ et $k \in K$ sont reliés si et seulement si ils sont reliés dans G (voir Figure 3.7). Autrement dit, G' est obtenu à partir de G en enlevant les arêtes induites par K . On met la capacité $c(k) = 1$ pour tout sommet $k \in K$ et $c(s) = b - 1$ pour tout $s \in S$. Une partition en b -cliques optimale est ensuite obtenue à partir d'un c -couplage max comme illustré dans la Figure 3.8 : chaque sommet de S est placé dans le même batch que ses voisins dans le c -couplage et l'ensemble V' des sommets de K insaturés par le couplage sont partitionnés arbitrairement en $\lceil |V'|/b \rceil$ batches.

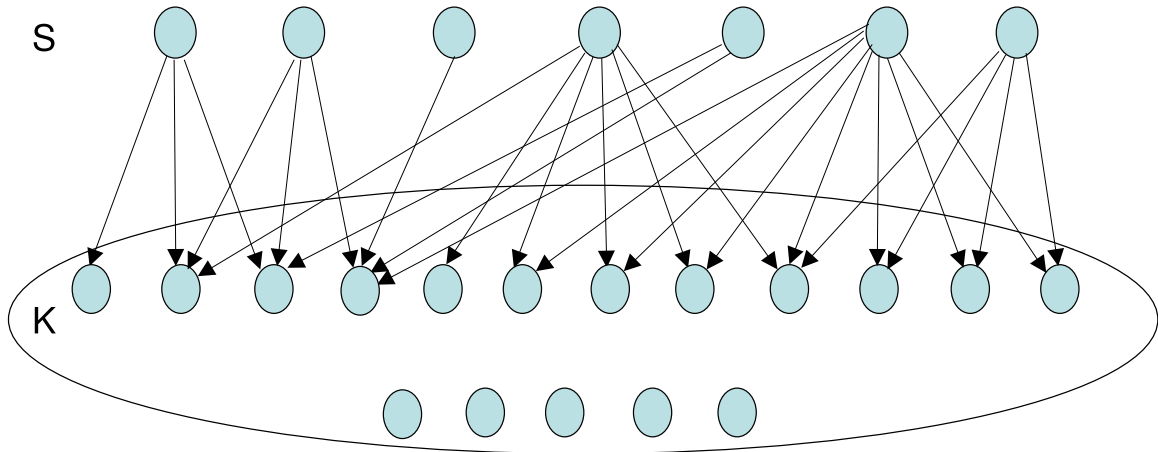


Figure 3.7 – Graphe biparti G' représentant un split graphe G (où G est le split graphe construit à partir de G' en remplaçant le stable dans l'ellipse par une clique).

Etant donné un c -couplage maximum, on peut aussi trouver un sous-ensemble de sommets qui définit $\sigma_b(G)$ avec une technique de chaînes alternées (voir [129], Chapitre 16 pour cette technique). Pour cela, on considère l'ensemble des sommets de S insaturés par le c -couplage (Figure 3.9). On recherche ensuite une chaîne

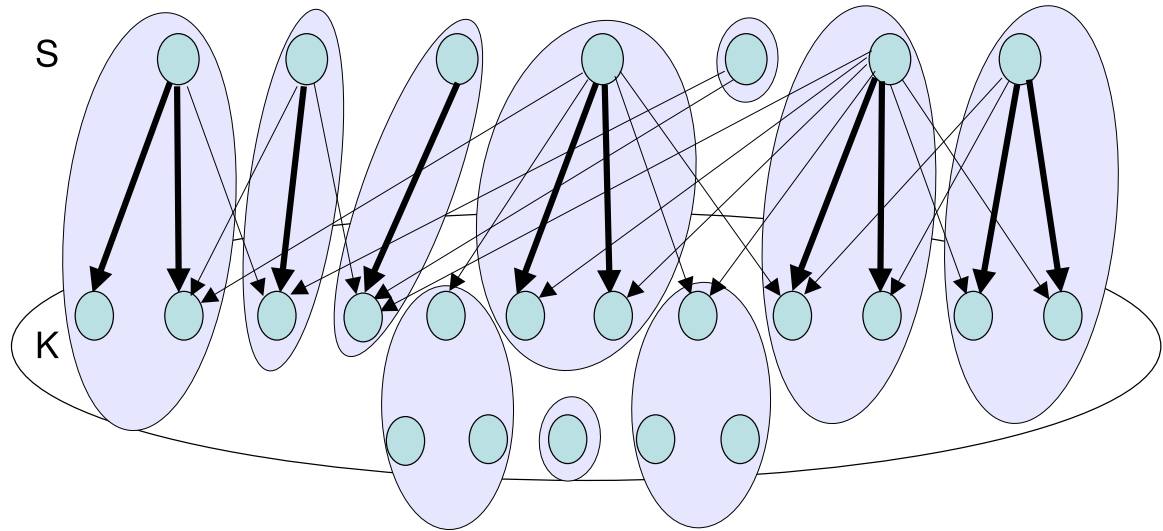


Figure 3.8 – c -couplage maximum sur G' et partition en b -cliques associée. Ici, $b = 3$. Pour $v \in S$, on a $c(v) = 2$ et pour $v \in K$, on a $c(v) = 1$.

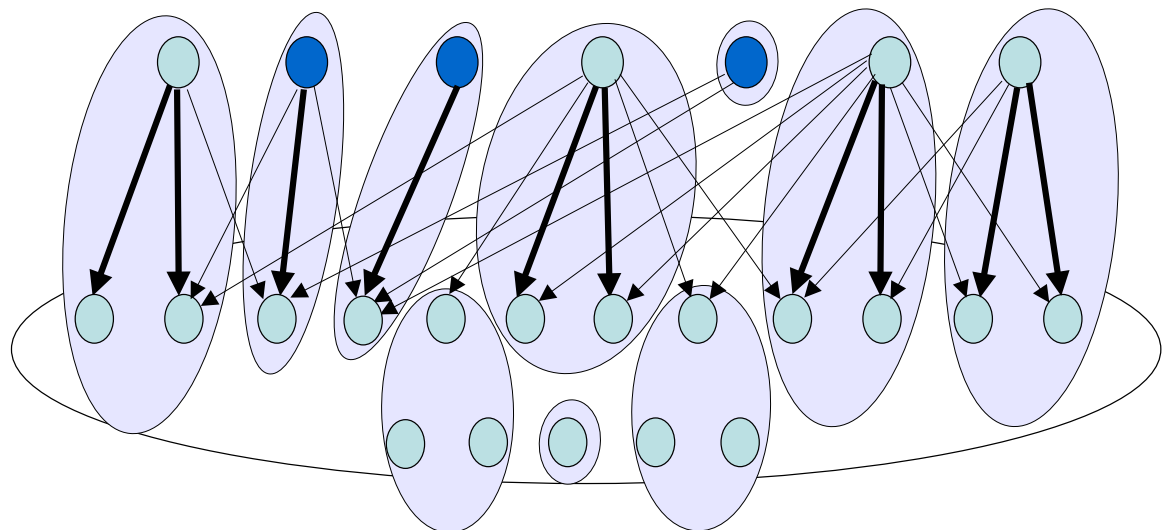


Figure 3.9 – Sommets de S insaturés par le c -couplage.

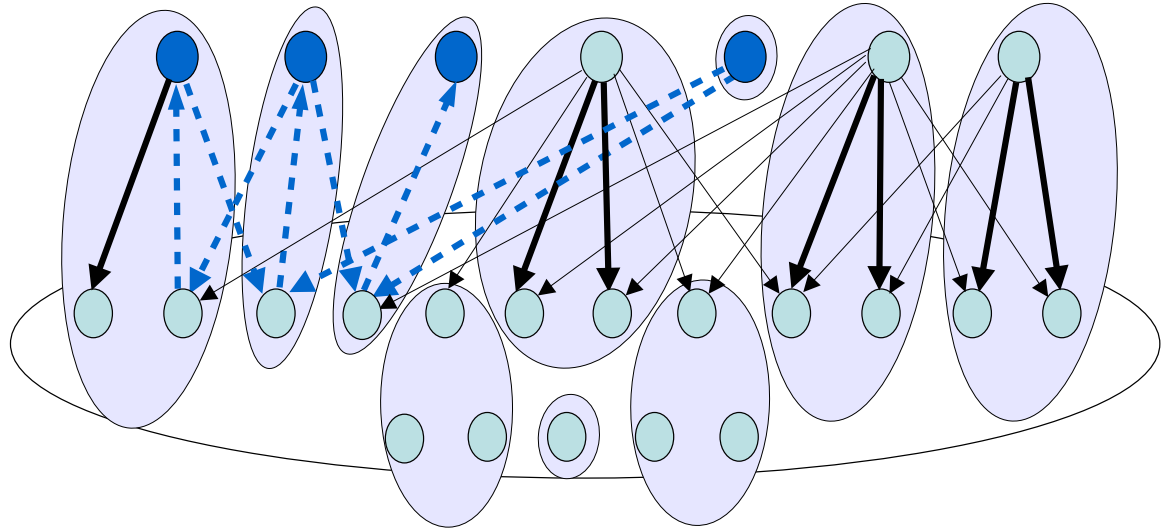


Figure 3.10 – Sommets “marqués” dans la recherche d’une chaîne augmentante.

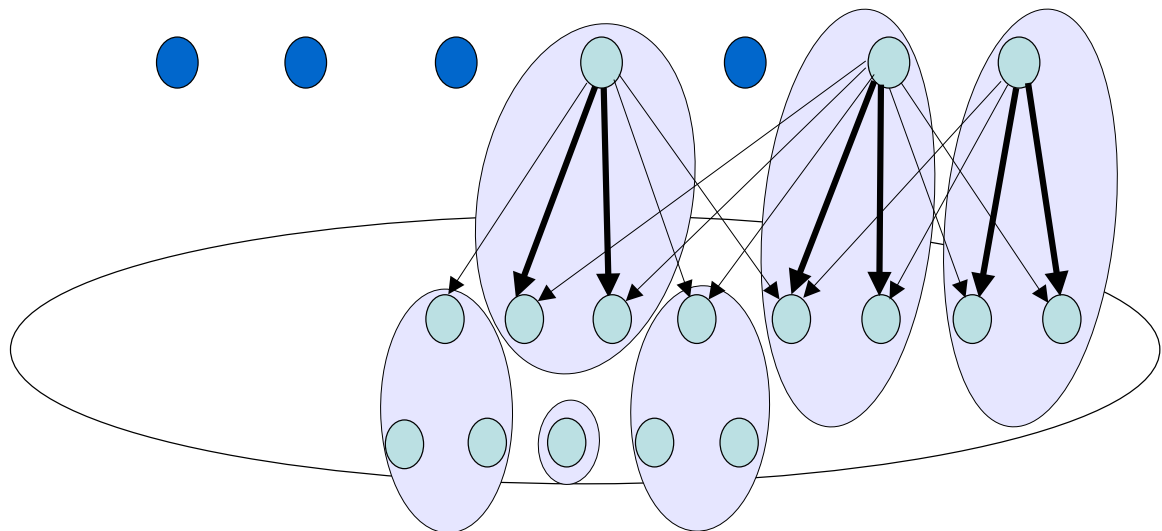


Figure 3.11 – Ensemble $S \subset V$ définissant $\sigma_3(G)$, construit à partir du couplage maximum en enlevant tous les voisins des sommets marqués.

augmentante pour le c -couplage, ce faisant on marque de nouveaux sommets de S (Figure 3.10). Soit S' l'ensemble des sommets de S ainsi marqués lors de la recherche d'une chaîne augmentante. Un ensemble définissant $\sigma_b(G)$ est obtenu en enlevant les voisins des sommets de S' (Figure 3.11). La validité de cette procédure est prouvée dans le cas plus général des graphes triangulés dans la prochaine section.

3.3.3 Réseau simplicial pour les triangulés

Dans cette section nous étudions la structure simpliciale des graphes triangulés. Nous montrons comment représenter de manière canonique un graphe triangulé par un graphe biparti et détaillons deux algorithmes (appelés S.c-M et Dual-S.c-M) :

Théorème 3.35 *Soit G un graphe triangulé et $b \in \mathbb{N}$. L'algorithme S.c-M fournit une partition $\mathcal{B} = \{B_1, \dots, B_l\}$ en b -cliques de G et Dual-S.c-M fournit un ensemble U tels que*

$$\bar{\chi}_b(G) = |\mathcal{B}| = \sum_i \left\lceil \frac{|C_i(U)|}{b} \right\rceil = \sigma_b(G)$$

Les deux solutions sont donc optimales. Les deux algorithmes sont polynomiaux.

En plus de généraliser la polynomialité de [PCliqB] sur les intervalles et sur les split, le théorème 3.35 généralise la polynomialité du problème du couplage multidimensionnel parfait [PerfectMDM] restreint aux triangulés [52] (c'est-à-dire la question de l'existence d'une partition de G en cliques, toutes de taille exactement b). En fait, la polynomialité de [PCliqB] sur les graphes triangulés peut être déduite des résultats de [52]⁷ :

Proposition 3.36 *Soit $G = (V, E)$ un graphe et b, k , deux entiers. Alors $\bar{\chi}_b(G) \leq k$ si et seulement si le graphe G' , obtenu à partir de G en ajoutant $kb - |V|$ sommets universels, a un couplage b -dimensionnel parfait. \square*

Autrement dit, [PerfectMDM] et [PCliqB] sont polynomialement équivalents sur toute classe de graphes fermée par ajout de sommets universels. Or les graphes triangulés sont fermés par ajout de sommets universels, puisque les trous n'ont pas de tels sommets.

Nous présentons quand même les algorithmes (découverts avant que nous ne prenions connaissance des résultats de [52]⁸), d'une part car l'étude simpliciale préliminaire est à notre avis intéressante en elle-même et d'autre part car nous prouvons la borne-perfection des graphes triangulés.

⁷ bien que cela ne nous soit apparu que tardivement et que Frédéric Gardi ait laissé ouvert le problème [PCliqB] sur les graphes triangulés dans sa thèse [79]. La réduction donnée dans la Proposition 3.36 est donc triviale après coup, mais le chemin qui y a mené fût bien long. . .

⁸ L'algorithme de [52] utilise aussi une technique de c -couplages successifs. L'avantage de [52] est de ne pas nécessiter la décomposition simpliciale canonique, mais un ordre simplicial quelconque.

3.3.3.1 Structure simpliciale des graphes triangulés

La caractérisation des graphes triangulés en termes de sommets simpliciaux (Théorème 2.7 i)) est équivalente au fait qu'un graphe est triangulé si et seulement si il a un ordre de décomposition simplicial. Un graphe triangulé a en général plusieurs tels ordres. On va donc raffiner l'idée d'élimination simpliciale en définissant un ordre partiel sur les sommets. L'idée de base (déjà exploitée dans [44] pour des questions d'Hamiltonicité) pour aboutir à une décomposition canonique, est d'enlever récursivement *tous* les sommets simpliciaux du graphe.

Soit G un graphe triangulé, on définit $\mathcal{S}(G)$ comme l'ensemble des sommets simpliciaux de G . Observons (voir exemple Fig. 3.12) que

Lemme 3.37 [44] $\mathcal{S}(G)$ est une union disjointe de cliques.

Preuve[nouvelle] Un graphe est une union disjointe de cliques si et seulement s'il est sans- P_3 (Section 2.2.3.2). Supposons qu'il y ait un P_3 induit dans $\mathcal{S}(G)$, alors le sommet au milieu du P_3 n'est pas simplicial, contradiction. \square

Appelons les cliques de $\mathcal{S}(G)$, les **blocs simpliciaux** de G . La **séquence simpliciale** de G est la suite G_1, \dots, G_t où $G_1 = G$, $G_t = \mathcal{S}(G_t)$ et pour tout $1 \leq i \leq t-1$, $G_{i+1} = G_i - \mathcal{S}(G_i)$ (autrement dit, G_{i+1} est le sous-graphe de G_i induit sur tous les sommets non-simpliciaux de G_i). L'ensemble \mathcal{K} de tous les blocs simpliciaux dans la séquence $\mathcal{S}(G_1), \dots, \mathcal{S}(G_t)$ forme une partition de $V(G)$.

Le Lemme suivant, nous permet de représenter un graphe triangulé par un graphe biparti de manière canonique.

Lemme 3.38 Soient L et M contenues dans $\mathcal{S}(G_i)$ et $\mathcal{S}(G_j)$ respectivement, avec $i < j$ et soit $v \in M$. Il existe une arête entre un sommet de L et v si et seulement si tous les sommets de L sont voisins de v .

Preuve Les sommets de L induisent une clique et sont simpliciaux dans $\mathcal{S}(G_i)$. \square

Définissons une relation binaire \rightarrow sur \mathcal{K} comme suit : $L, M \in \mathcal{K}$ satisfont $L \rightarrow M$ si L et M sont contenues dans $\mathcal{S}(G_i)$ et $\mathcal{S}(G_j)$ respectivement, $i < j$ et il y a une arête dans G entre un sommet de L et un sommet de M .

Grâce à la condition " $i < j$ ", la relation \rightarrow est acyclique. Sa fermeture transitive réflexive définit donc un ensemble partiellement ordonné (\mathcal{K}, \succeq) .

Pour $v \in V$, notons $\mathbf{K}(v)$, l'unique bloc de \mathcal{K} qui contient v . On utilisera aussi la notation $\mathbf{K} \rightarrow \mathbf{x}$ qui signifie que l'une des deux assertions équivalentes du Lemme 3.38 est satisfaite.

Proposition 3.39 Soient $K, L, M \in \mathcal{K}$ tels que $K \rightarrow L$ et $K \rightarrow M$. Alors soit $M \rightarrow L$ soit $L \rightarrow M$.

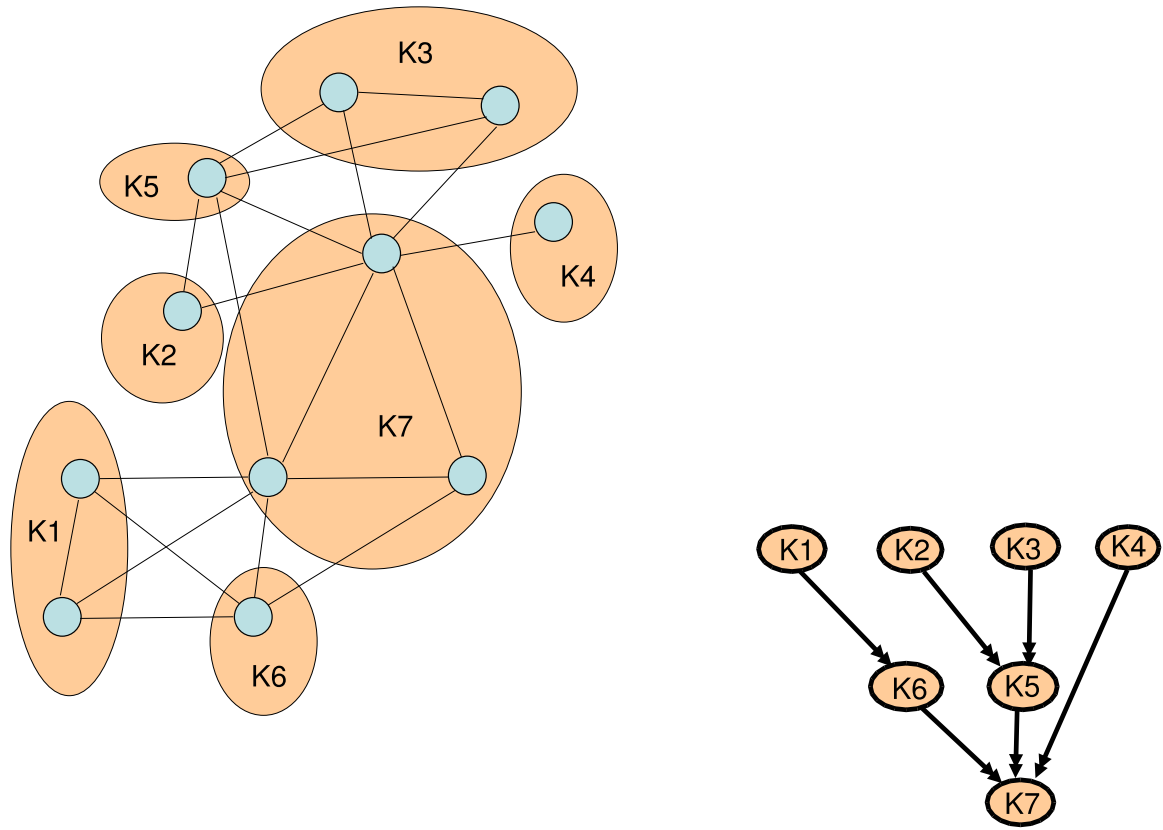


Figure 3.12 – Un graphe triangulé G dont les sommets sont partitionnés selon l'ensemble des blocs simpliciaux. Le diagramme de Hasse de (\mathcal{K}, \succeq)

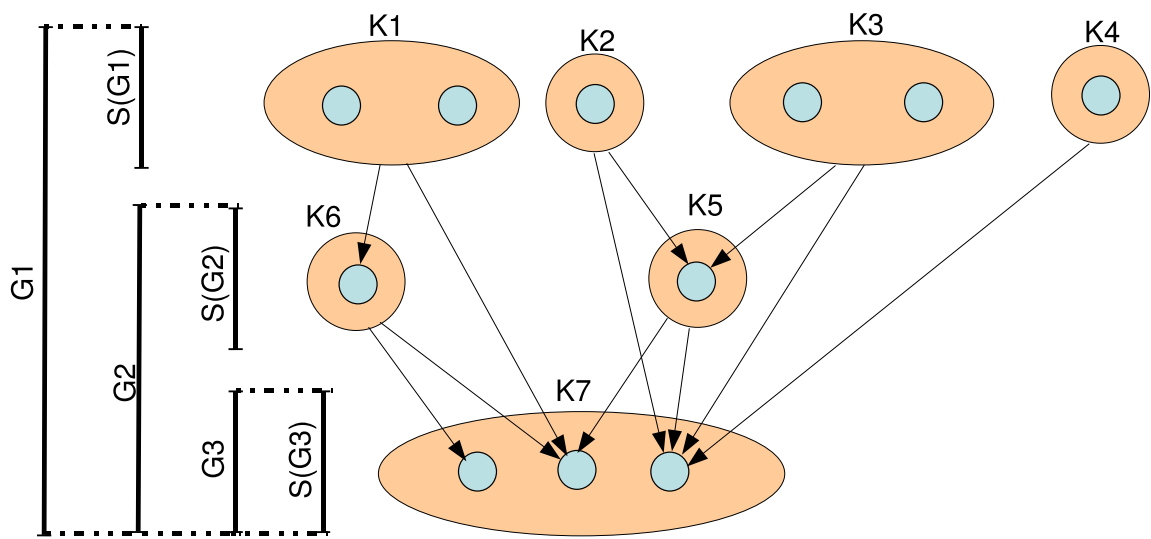


Figure 3.13 – La séquence G_1, \dots, G_t et le réseau simplicial $\mathcal{SN}(G)$

Preuve Soit i tel que $K \in \mathcal{S}(G_i)$. Puisque $K \rightarrow L$ et $K \rightarrow M$, il existe $v \in L$ et $w \in M$ tels que $K \rightarrow v$ et $K \rightarrow w$. Comme L est simpliciale dans G_i , on a $(v, w) \in E(G_i)$. On a donc soit $M \rightarrow L$ soit $L \rightarrow M$. \square

Pour $K, L \in \mathcal{K}$, notons $K \twoheadrightarrow L$ le fait que K couvre L dans le poset (\mathcal{K}, \succeq) .

Corollaire 3.40 ([44]) *Le diagramme de Hasse de (\mathcal{K}, \succeq) est une di-forêt entrante. Si G est connexe, c'est un di-arbre entrant.*

Preuve Rappelons qu'une di-forêt entrante est un digraphe acyclique dont les degrés sortants sont ≤ 1 . Supposons, par contradiction, qu'il existe $K, L, M \in \mathcal{K}$ tels que $K \twoheadrightarrow L$ et $K \twoheadrightarrow M$. D'après la proposition 3.39, soit $M \rightarrow L$ soit $L \rightarrow M$. Par symétrie, supposons $L \rightarrow M$. Cela contredit $K \twoheadrightarrow M$ puisque $K \succeq M$ peut être déduit de $K \rightarrow L$ et de $L \rightarrow M$. \square

Corollaire 3.41 *Soient $K, L, M \in (\mathcal{K}, \succeq)$ tels que $K \succeq L$ et $K \succeq M$. Alors soit $L \succeq M$ soit $M \succeq L$.*

Preuve L'ensemble $\{N \in \mathcal{K} \mid N \preceq K\}$ forme une chaîne dans (\mathcal{K}, \succeq) . L et M appartenant à cette chaîne, ils sont comparables. \square

Corollaire 3.42 ([44]) *Soit $K_1 \succeq K_j \succeq K_z \in \mathcal{K}$. Si $K_1 \rightarrow K_z$ alors $K_j \rightarrow K_z$.*

Preuve Le Corollaire 3.40 implique qu'il existe un unique chemin $K_1 \twoheadrightarrow K_2 \twoheadrightarrow \dots \twoheadrightarrow K_j \twoheadrightarrow \dots \twoheadrightarrow K_z$ dans le Diagramme de Hasse de (\mathcal{K}, \succeq) ayant K_1 et K_z comme extrémités. En particulier, pour chaque i dans l'intervalle $[1..z-1]$, on a $K_i \rightarrow K_{i+1}$. Donc d'après la Proposition 3.39, si $K_i \rightarrow K_z$ alors $K_{i+1} \rightarrow K_z$. Par induction, on a $K_j \rightarrow K_z$ pour $j \in [1..z-1]$. \square

Pour un graphe triangulé G ayant \mathcal{K} comme blocs simpliciaux, le **réseau simplicial** $\mathcal{SN}(G)$ de G est le graphe biparti ayant (V, \mathcal{K}) comme bipartition et ayant une arête entre K et v si $K \rightarrow v$. Autrement dit

$$(3.7) \quad K \rightarrow v \iff \begin{cases} K \succeq K(v) \\ (k, v) \in E(G) \quad \forall k \in K \end{cases} \iff \begin{cases} K \succeq K(v) \\ \exists k \in K \text{ tel que } (k, v) \in E(G) \end{cases}$$

La Figure 3.13 montre le réseau simplicial associé au graphe triangulé de la Figure 3.12.

Lemme 3.43 *Pour tout $K \in \mathcal{K}$, soit $A_K := \{v \in V(G) \mid v \in L \succeq K\}$, soit $N_{\mathcal{SN}(G)}(K) \subseteq V$ le voisinage de K dans $\mathcal{SN}(G)$ et $A_K^+ := A_K \cup N_{\mathcal{SN}(G)}(K)$. Alors l'ensemble $N_{\mathcal{SN}(G)}(K) \subseteq V$ déconnecte l'ensemble A_K de $V \setminus A_K^+$.*

Preuve Supposons que $(x, y) \in E(G)$ avec $x \in A_K$ et $y \in V \setminus A_K^+$. On a soit $K(y) \rightarrow x$, soit $K(x) \rightarrow y$. Si $K(y) \rightarrow x$, alors $K(y) \succeq K(x) \succeq K$, contradiction. Si $K(x) \rightarrow y$, on a $K(x) \succeq K(y)$. Comme $K(x) \succeq K$, d'après le Corollaire 3.41, on a soit $K(y) \succeq K$ soit $K \succeq K(y)$. Or $K(y) \succeq K$ est exclu par hypothèse. On

a donc $K(x) \succeq K \succeq K(y)$ et $K(x) \rightarrow K(y)$ et donc, d'après le Corollaire 3.42, $K \rightarrow K(y)$ ce qui contredit que $y \in V \setminus A_K^+$. \square

3.3.3.2 c -couplages successifs pour les triangulés

Lemme 3.44 *Soit G un graphe triangulé ayant \mathcal{K} comme blocs simpliciaux. Pour toute clique B de G , l'ensemble des cliques de \mathcal{K} qui ont une intersection non-vide avec B forme une chaîne dans le poset (\mathcal{K}, \succeq) .*

Preuve Soient $v, w \in B$ tels que $K(v) \neq K(w)$. Puisque $(v, w) \in E(G)$ mais n'appartiennent pas au même bloc simplicial, on a $v \in \mathcal{S}(G_i)$ et $w \in \mathcal{S}(G_j)$ avec $i < j$. Par définition de \rightarrow , on a $K(v) \rightarrow K(w)$ et donc $K(v) \succeq K(w)$. \square

Etant donné une clique B d'un graphe triangulé G , le Lemme 3.44 permet de définir l'**étiquette** de B comme le plus petit élément de (\mathcal{K}, \succeq) qui intersecte B .

Appelons **bien étiquetée sur** $K \in \mathcal{K}$, une partition en b -cliques $\mathcal{B} = (B_1, \dots, B_m)$ d'un graphe triangulé G telle qu'il y a au plus un B_i tel que $|B_i \cap K| \neq 0$ et $|B_i \cap K| \neq b$. La partition sera dite bien étiquetée si elle est bien étiquetée sur tous les $K \in \mathcal{K}$.

En particulier dans une partition bien étiquetée sur $K \in \mathcal{K}$, il y a au plus une b -clique qui a K comme étiquette sans être totalement incluse dans K .

Lemme 3.45 [Facultatif pour Th 3.35] *Pour tout graphe triangulé G et tout b , il existe une $\overline{\chi}_b(G)$ -partition bien étiquetée en b -cliques de G .*

Preuve Montrons comment définir une $\overline{\chi}_b$ -partition bien étiquetée en b -cliques à partir d'une partition en b -cliques $\mathcal{B} = (B_1, \dots, B_{\overline{\chi}_b})$. Supposons que la partition est déjà bien étiquetée pour chaque $K \in \mathcal{S}(G_j)$, pour $j = [1..p-1]$ et liftons la pour les $K \in \mathcal{S}(G_p)$. Pour chaque $K \in \mathcal{S}(G_p)$ l'ensemble de sommets

$$\mathcal{B}(K) := \cup_{\{B_i \mid \text{étiquette}(B_i)=K\}} B_i$$

forme une clique de G_p . On peut donc redécouper $\mathcal{B}(K)$ en $\lceil |\mathcal{B}(K)|/b \rceil$ b -cliques, toutes de taille b sauf éventuellement la dernière en affectant d'abord tous les sommets de K . On aura donc au plus une clique (notée \mathbf{B}_K si elle existe) telle que $|B_K \cap K| \not\equiv 0 \pmod{b}$. \square

L'intérêt des partitions bien étiquetées est de pouvoir être représentées par les c -couplages du réseau simplicial de G .

Remarque 3.46 *Soit un graphe triangulé G et $b \in \mathbb{N}$. A toute partition bien étiquetée $\mathcal{B} = (B_1, \dots, B_m)$ de G en b -cliques, on peut associer de manière unique une capacité c sur $\mathcal{SN}(G)$ telle que $c(v) := 1 \quad \forall v \in V(G)$ et pour tout $K \in \mathcal{K}$,*

$$c(K) := \begin{cases} b - |B_K \cap K| & \text{si } B_K \text{ existe} \\ 0 & \text{sinon} \end{cases}$$

On peut ensuite associer de manière unique un c -couplage \mathcal{M} à \mathcal{B} de la manière suivante : $(K, v) \in \mathcal{M}$ si $v \in B_K \setminus K$. \square

Un exemple de la correspondance entre c -couplages et partitions en b -cliques est donné dans la paire de Figures (3.14,3.15)

Notons que la réciproque est fautive dans le sens qu'un même c -couplage peut être défini par plusieurs partitions bien étiquetées en b -cliques. Toutefois, les seuls choix que l'on a concernent la manière de partitionner l'intérieur des blocs simpliciaux en cliques de taille b . Par exemple, à partir du c -couplage de la Figure 3.15, on peut reconstruire la partition bien étiquetée de manière unique au choix près d'un sommet de K_1 qui doit être affecté à B_{K_1} . Pour fixer les idées :

Remarque 3.47 Soit un graphe triangulé G , $b \in \mathbb{N}$, une capacité c sur $\mathcal{SN}(G)$ un c -couplage \mathcal{M} . Pour tout $K \in \mathcal{K}$, soit $K' \subseteq V$ l'ensemble des sommets insaturés par \mathcal{M} . Supposons que

$$\begin{cases} c(v) = 1 & \text{pour tout } v \in V(G) \\ 0 \leq c(K) \leq b - 1 & \text{pour tout } K \in \mathcal{K} \\ c(K) + |K'| \equiv 0 \pmod{b} & \text{pour tout } K \in \mathcal{K} \end{cases}$$

alors on peut associer une partition bien étiquetée de G en b -cliques à \mathcal{M} telle que l'application de la procédure du Lemme 3.45 redonne c et \mathcal{M} .

Preuve Il suffit d'appliquer l'algorithme S.c-M ci-dessous à partir du point iv. \square

La morale des Remarques 3.47 et 3.46 est donc que *toute l'information nécessaire pour construire une partition en b -cliques optimale peut être représentée par une capacité c et un c -couplage \mathcal{M} sur $\mathcal{SN}(G)$.*

Cette réduction aux c -couplage permet d'utiliser l'attirail d'optimisation que l'on connaît sur ceux-ci. Toutefois, il reste le problème de savoir quelles capacités affecter à chaque $K \in \mathcal{K}$.

La preuve algorithmique du Lemme 3.48 suivant va nous donner une clé pour ce problème de calcul de capacités.

Lemme 3.48 (Facultatif pour Th 3.35) Soit $G = ((A, B), E)$ un graphe biparti. Soit $c : A \cup B \rightarrow \mathbb{N}$ et une famille en chaîne, $\mathcal{A} = (A_1 \subset \dots \subset A_z)$ de sous-ensembles de A . Alors il existe un c -couplage de G qui soit simultanément maximum sur chacun des sous graphes $G[A_i, B]$ pour $i \in [1..z]$.

Preuve Directement en utilisant un algorithme de chaînes alternées : supposons que l'on ait construit un c -couplage \mathcal{M} simultanément maximum sur $G[A_j, B]$ pour $j = [1..i]$. On peut augmenter \mathcal{M} jusqu'à un c -couplage maximum sur $G[A_{i+1}, B]$ par des échanges de chaînes alternées. Pour chaque sommet v de G , au cours de chaque échange, le nombre d'arêtes incidentes à v dans \mathcal{M} ne diminue pas. \square

L'intuition qui a mené à l'algorithme S.c-M se base sur le fait que dans la preuve

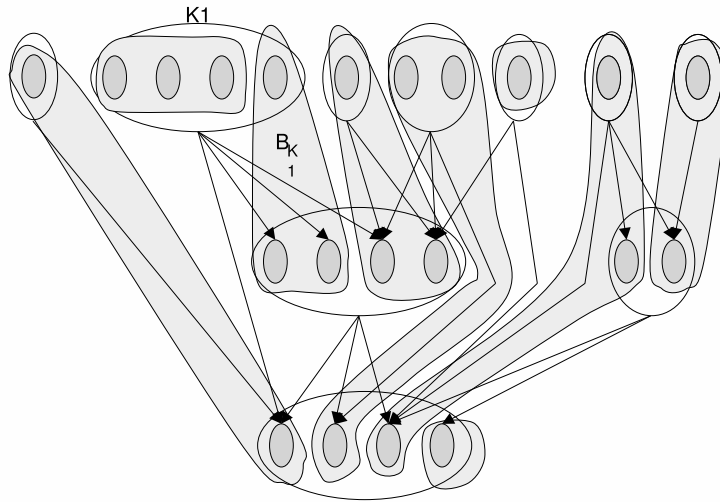


Figure 3.14 – Une partition bien étiquetée avec $b = 3$

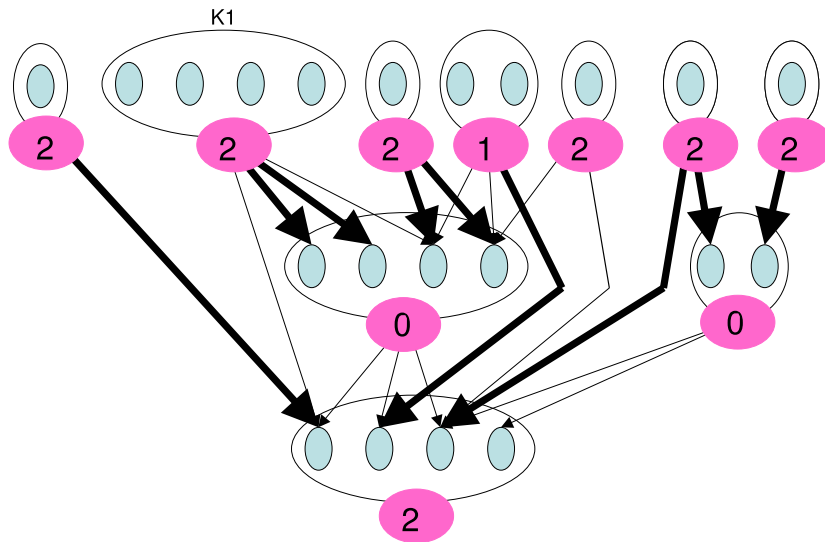


Figure 3.15 – Les capacités sur \mathcal{K} et le c -couplage correspondant à la partition bien étiquetée de la Figure 3.14

du Lemme 3.48, *on peut calculer les capacités au fur et à mesure de la construction du couplage. Il suffit juste que pour chaque i , les capacités des sommets dans A_i ainsi que les capacités dans $N(A_i)$ soient déterminées au moment de l'optimisation de \mathcal{M} sur $G[A_i, B]$.*

Le Lemme 3.48 doit être interprété dans S.c-M en prenant : $A = V$, $B = \mathcal{K}$ et $A_i = \cup_{j=1}^i \mathcal{S}(G_j)$. Définissons aussi $\mathcal{SN}_i := \mathcal{SN}(G)[A_i, B]$.

l'algorithme "Successive c-matching" (S.c-M) :

- i) Construire le graphe biparti $\mathcal{SN}(G) = ((V(G), \mathcal{K}(G)), E')$.
- ii) Donner une capacité $c(v) = 1$ à tous les sommets dans la classe $V(G)$.
- iii) Pour i allant de 1 à t
 - A) Trouver un c -couplage (noté $(\mathcal{M}(\mathcal{SN}_i(G)))$) simultanément maximum pour tous les $\mathcal{SN}_j(G)$, $1 \leq j \leq i$.
 - B) Pour chaque clique K dans $\mathcal{S}(G_{i+1})$
 - 1) Compter le nombre $|K'|$ de sommets de K insaturés par $(\mathcal{M}(\mathcal{SN}_i(G)))$.
 - 2) Donner à K la capacité $0 \leq c(K) < b$ avec $c(K) \equiv -|K'| \pmod{b}$.
- iv) Noter \mathcal{M} le c -couplage de $\mathcal{SN}(G)$ obtenu en iii).
- v) Pour chaque clique K dans \mathcal{K}
 - A) Parmi l'ensemble K' de sommets de K insaturés par \mathcal{M} , en choisir $\lfloor |K'|/b \rfloor$ et les partitionner en batchs de taille b .
 - B) Affecter les sommets restants de K' , ainsi que les sommets de $V(G)$ adjacents à K dans \mathcal{M} à un même batch.

Le déroulement S.c-M est montré sur un exemple dans la Figure 3.16.

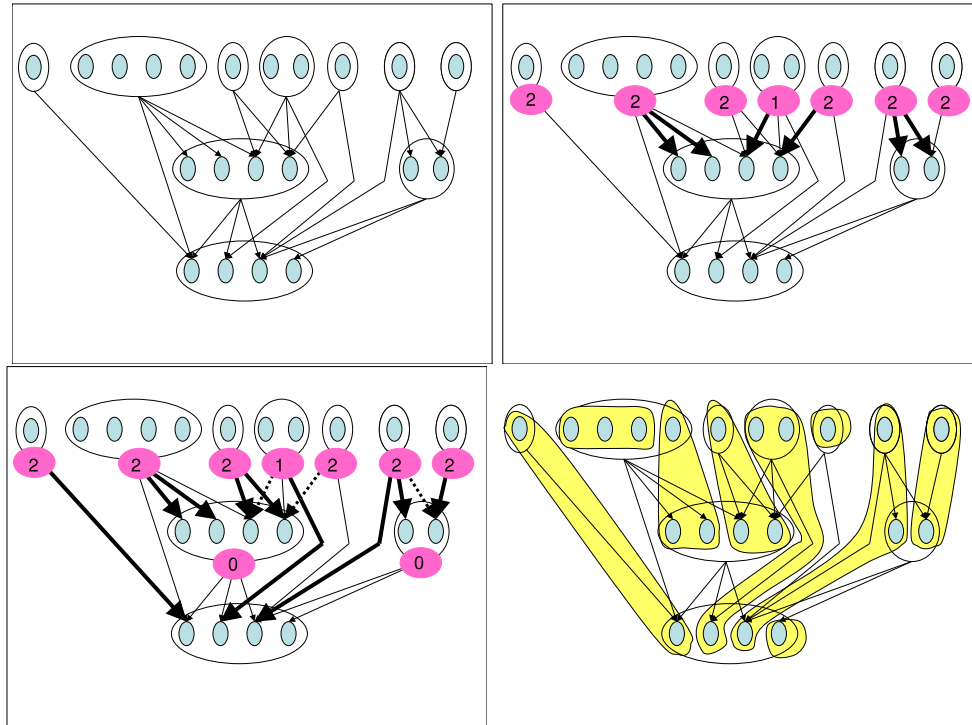
Considérons maintenant le problème "dual" de la partition en b -cliques. On cherche donc un sous-ensemble de sommets U de V qui définisse $\sigma_b(G)$ (équation 3.2). Dans la partition définie par l'algorithme S.c-M, si tous les batchs sont saturés, $U = V$ fait l'affaire. En général, il existe des batchs insaturés (marqués en bleus dans le quartier supérieur gauche de la Figure 3.17). On applique alors

l'algorithme Dual-S.c-M :

- i) Appliquer S.c-M.
- ii) Marquer toutes les sources insaturées.
- iii) Appliquer la procédure d'augmentation du couplage par des chaînes alternées et marquer les sources ainsi atteintes.
- iv) Retourner $U := V \setminus \{v \in V \mid \exists K \in \mathcal{K}, K \text{ marqué}, (K, v) \in \mathcal{SN}(G)\}$

Le déroulement Dual-S.c-M est montré sur un exemple dans la Figure 3.17.

Le Lemme 3.49 donne l'élément clé dans la preuve d'optimalité de la paire d'algorithmes S.c-M :



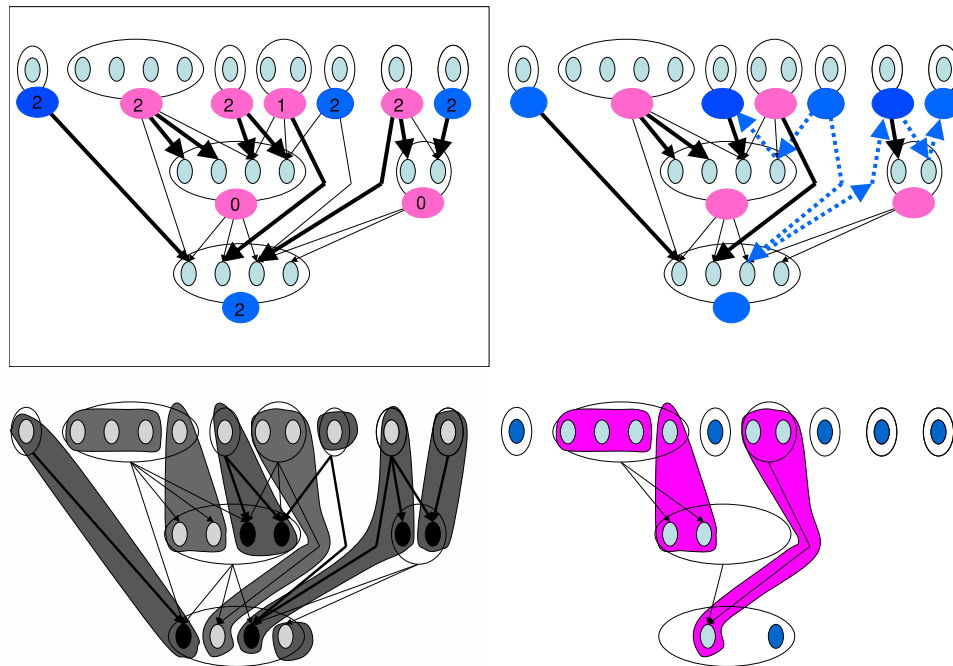
Up-left Phase i) et ii). on ne représente pas les capacités des sommets de G car elles sont unitaires.

Up-right Phase *iii*) A) A) Comme $N(\mathcal{S}(G_1)) = \emptyset$, le couplage vide est optimal.
Phase *iii*) B) On affecte les capacités (en rose) aux blocs simpliciaux de G_1 .
Phase *iii*) A) On calcule un couplage max entre $\mathcal{S}(G_1) \cup \mathcal{S}(G_2)$ et son voisinage (arcs en gras).

Down-left Phase *iii*) B) Affectation des capacités aux cliques simpliciales de G_2 .
Phase *iii*) A) Calcul d'un couplage max par un algorithme de chaînes alternées. Noter que les degrés dans \mathcal{M} des sommets de G comme des sommets de \mathcal{K} n'ont pas diminué.

Down-right Phase *iii*) B) (non-dessiné) On met la capacité 2 sur l'unique clique de G_3 . Phase *iv*) Le couplage \mathcal{M} est celui obtenu en [Down-left]. Phase *v*) A) Une seule clique a un nombre suffisant de sommets non-saturés pour former un batch. Phase *v*) B) Tous les autres batchs sont définis par le couplage.

Figure 3.16 – Déroulement de S.c-M sur un exemple



Up-left Phase *ii*).

Up-right Phase *iii*).

Down-left Phase *iv*) Les sommets remplis de noirs sont les voisins d'éléments de \mathcal{K} marqués à la phase *iii*).

Down-right Phase *iv*) Noter le fait général, qui permet de prouver l'optimalité de S.c-M et de Dual-S.c-M : *L'ensemble U renvoyé se décompose en composantes connexes, chacune d'elle intersectant au plus un batch marqué.*

Figure 3.17 – Déroulement de Dual-S.c-M sur l'exemple de la Figure 3.17

Lemme 3.49 Soit U l'ensemble renvoyé par Dual-S.c-M. Chaque composante connexe de $G[U]$ contient au plus un batch marqué.

Preuve Supposons qu'on ait un chemin $P = (v_0, \dots, v_z)$ dans $G[U]$ tel que v_0 et v_z soient dans des batchs marqués distincts. Supposons P minimal par inclusion (et en particulier induit). D'après le Lemme 3.43, $v_0 \leftarrow v_1$ et $v_{z-1} \rightarrow v_z$. Prenons un sommet $v_j \in P$ qui appartienne à un $\mathcal{S}(G_i)$ avec i minimum, puis parmi ceux-là, avec j minimum. On a alors

$$v_{j-1} \leftarrow v_j - \dots - v_{j+\alpha} \rightarrow v_{j+\alpha+1}$$

avec $\alpha \geq 0$. On a clairement que $K(v_j) = \dots = K(v_{j+\alpha})$, donc $v_j \rightarrow v_{j+\alpha+1}$ et donc $\alpha = 0$ par minimalité de P . Mais alors, par simplicialité de v_j , on a $(v_{j-1}, v_{j+1}) \in E(G)$, ce qui contredit la minimalité de P . \square

Preuve du Théorème 3.35 Soit U l'ensemble renvoyé par Dual-S.c-M. Notons premièrement que les batchs induits sur $G[U]$ sont des cliques et sont donc inclus dans une seule composante connexe de $G[U]$. Deuxièmement les sommets de $V \setminus U$ étant des successeurs de sources marquées, ils sont tous dans des batchs marqués. Chaque composante connexe C_i de $G[U]$ comporte donc un certain nombre de batchs saturés et au plus un batch insaturé. La quantité $\lceil |C_i|/b \rceil$ est donc égale au nombre de batchs contenus dans C_i . Chaque batch étant représenté par au moins un sommet dans U , le nombre $\sum_i \lceil |C_i(U)|/b \rceil$ est égal au nombre de batchs de la solution renvoyée par S.c-M, ce qui termine la preuve. \square

3.4 Pour aller plus loin

3.4.1 Coloration avec bornes non-uniformes

Nous étudions ici le problème de la coloration bornée *non-uniforme* dans lequel les couleurs n'ont pas toutes la même cardinalité maximum :

[ColBNU] Coloration bornée non-uniforme

Données : Un graphe G , une liste $\{b_i\}_{i=1}^k$ de nombres entiers (avec $k \leq |V|$ et $|V| \geq b_1 \geq \dots \geq b_k \geq 1$).

Résultat : Existe-t-il une k -coloration de G telle que la couleur i soit utilisée pour au plus b_i sommets.

Remarque 3.50 La supposition qu'il y a au plus $|V|$ couleurs disponibles ($k \leq |V|$) et qu'elles ont toutes une cardinalité maximum majorée par $|V|$ ne réduit pas la généralité du problème (Si il y a plus que $|V|$ couleurs, on peut toujours partitionner et une capacité plus grande que $|V|$ est inutile). Cette supposition assure que la taille de l'instance est polynomiale dans la taille du graphe. \square

Malheureusement, [ColBNU] se trouve être NPC, même pour des classes très pauvres de graphes, dans lesquelles [ColB] est trivial. Avant d'étudier la frontière de polynomialité observons que [ColBNU] ne peut pas être formulé comme un problème de couverture des sommets d'un hypergraphe par un nombre minimum d'arêtes (alors que [ColB], si). En effet, si un seul b_i est grand, deux couleurs réalisables séparément peuvent ne pas être réalisables simultanément.

Notons que [ColBNU] contient le problème de la k -coloration partielle maximum (défini dans le Chapitre 5) :

Proposition 3.51 *Pour toute classe \mathcal{G} de graphes, [ColPart] restreint à \mathcal{G} se réduit polynomialement à [ColBNU] lui aussi restreint à \mathcal{G} .*

Idée de preuve Pour tout graphe G et tout $k, t \in \mathbb{N}$, il existe une k -coloration partielle de G qui contient au moins t sommets si et seulement si G admet une coloration $\{b_i\}$ -bornée avec

$$(3.8) \quad b_i := \begin{cases} |V| & \text{si } i \leq k \\ 1 & \text{si } k < i \leq k + |V| - t \\ 0 & \text{si } k + |V| - t < i \end{cases}$$

□

Théorème 3.52 *[ColBNU] est NP-complet même pour les graphes*

- i) co-sans- P_3 .*
- ii) split.*

Idée de preuve i) Considérons le problème équivalent de partition en cliques $\{b_i\}$ -bornées sur les graphes sans- P_3 . On réduit [3-PARTITION] à ce problème en considérant le graphe qui a m cliques, chacune de cardinalité B avec la liste de capacités b_1, \dots, b_{3m} donnée par la liste des $3m$ valeurs $s(a)$ des éléments $a \in A$.
ii) Le problème de k -coloration partielle étant NP-complet pour les split graphes [138], il suffit d'appliquer la Proposition 3.51 □

Théorème 3.53 *Le problème [ColBNU] est polynomial pour les graphes*

- i) co-bipartis (couplage biparti)*
- ii) sans- P_3*

Idée de preuve i) Ce cas se réduit facilement au couplage max : puisque les cliques ont taille au plus 2 dans un biparti, on peut réduire l'information contenue dans la liste des b_i à seulement deux paramètres. Ces deux paramètres sont le nombre t de b_i tels que $b_i \geq 2$ et le nombre k (de b_i tels que $b_i \geq 1$). Le problème a une solution si et seulement si $\min\{\nu(G), t\} + k \geq |V|$.

ii) La polynomialité est une conséquence directe du théorème d'union des matroïdes, car les stables d'un graphe sans- P_3 définissent les indépendants d'un ma-

troïde sur V (voir Proposition 3.75 pour une extension). La polynomialité a aussi été établie par de Werra [57] qui a formulé un problème plus général que [ColBNU] par un programme linéaire dont la totale unimodularité de la matrice de contraintes caractérise les graphes sans- P_3 . \square

Remarque 3.54 *Il semble que l'algorithme glouton suivant résout aussi [ColBNU] pour les graphes sans- P_3 . Le graphe est composé de α cliques disjointes, ordonnées par taille décroissante. Considérer les b_i dans un ordre arbitraire. La première couleur est composée de $\min\{\alpha, b_1\}$ sommets (un sommet par clique, pour les $\min\{\alpha, b_1\}$ premières cliques de G). Enlever les sommets coloriés, réordonner les cliques par taille décroissante et recommencer pour chacun des b_i .*

La question suivante semble spécialement digne d'intérêt autant du point de vue des applications que du point de vue théorique.

Question 3.55 *Quelle est la complexité de [ColBNU] sur les line-graphes de bipartis⁹ ?*

Nous adressons bien sûr la même question pour les classes de graphes pour lesquelles à la fois [ColB] et [PartialColor] sont polynomiaux comme les threshold et surtout les forêts. La question pour les forêts mérite probablement aussi une attention particulière car plusieurs articles discutent de [ColB] sur cette classe de graphe (voir Section 3.4.5). De plus, [ColBNU] est polynomial sur les forêts quand le nombre de couleurs est fixé [16] avec une méthode qui ne semble pas s'étendre à un nombre variable de couleurs.

3.4.2 Approximabilité et complexité à paramètres fixés

Dans l'étude du problème [ColB], la complexité de la restriction à des classes de graphes peut être complétée par plusieurs autres paradigmes. Par exemple, on peut fixer la valeur d'un des paramètres b et k . On peut aussi chercher à donner des solutions avec de bonnes garanties de performance. Nous évoquons ces deux approches.

3.4.2.1 Complexité à paramètres fixés

Théorème 3.56 *[ColB] est polynomial si l'on se restreint aux graphes*

i) sans- P_4 et à b fixé [22]

⁹Il ne semble pas que le principe de l'échange bichromatique permette de résoudre [ColBNU] sur les line-graphes de bipartis. En effet l'échange bichromatique permet seulement d'équilibrer la cardinalité des couleurs et non d'accentuer les déséquilibres. Si [ColBNU] est polynomial cela cache probablement une propriété d'échange intéressante que satisfont les lines de bipartis et qui mériterait d'être mise en avant.

- ii) sans-PA et à k fixé [22]
- iii) d'intervalles et à k fixé [22]
- iv) bipartis et à b fixé [22]
- v) quelconques et à $k \leq 2$ fixé [9]

Nous commentons le point v), que nous avons découvert indépendamment de [9]. Le nombre chromatique d'un graphe G est inférieur ou égal à 2 si et seulement si G est biparti. Mais dans une bipartition d'un graphe, une des deux couleurs peut avoir beaucoup plus de sommets que l'autre. Si G est connexe, cette bipartition est unique et ce déséquilibre est intrinsèque à G . En revanche, si G n'est pas connexe, le déséquilibre peut parfois être compensé d'une composante à l'autre. Supposons que l'on colorie les composantes connexes de G une par une. La marge de manœuvre que nous avons se résume à choisir quel côté de la bipartition est affecté à quelle couleur. L'influence de ce choix sur le déséquilibre ne dépend que de la différence de cardinalité entre les deux composantes connexes. Ces observations mènent à :

Proposition 3.57 [9] *Le problème de la 2-coloration b -bornée est polynomial.*

Idée de preuve Clairement, $\chi_b(G) \leq 2$ implique que G est biparti. Supposons donc que G soit biparti. Notons $\mathcal{C} = \{C_1, \dots, C_t\}$ les composantes connexes de G . Pour $i \in [1..t]$, notons (c_i, d_i) la cardinalité des deux composantes connexes de C_i ainsi que $q_i := \min\{c_i, d_i\}$ et $p_i := \max\{c_i, d_i\} - q_i$. Notons $b' := b - \sum_{i=1}^t q_i$. On a que $\chi_2(G) \leq 2$ si et seulement l'ensemble $[1..t]$ peut être partitionné en deux sous-ensembles tels que la somme des p_i sur chacun est inférieur à b' . Cela est une instance du problème "Partition" qui est pseudopolynomial. La polynomialité de [ColB] découle du fait que les nombres $\{c_i\}$ et $\{d_i\}$ sont codés en unaire au travers des sommets du graphe. \square

3.4.2.2 Minimisation du nombre de couleurs et approximabilité

[ColB] peut être redéfini en deux versions optimisation : ayant fixé l'un de b ou k dans l'entrée, on cherche à minimiser l'autre dans l'algorithme. Nous ne décrivons qu'une de ces possibilités avec le problème :

[ColB] Coloration b -bornée (version optimisation de k)

Données : Un graphe G , un entier b .

Résultat : Une k -coloration b -bornée de G

Mesure : k (à minimiser).

Comme corollaire des résultats de la Section 3.1.2, on a :

Corollaire 3.58 *Soit \mathcal{G} une classe de graphe contenue dans les sans- $K_{1,3}$. La version optimisation de k de [ColB] et de [Color] restreints à \mathcal{G} ont le même ratio et*

le même delta d'approximabilité. \square

Idée de preuve [ColB] ne peut pas être mieux approximé que [Color] puisque [Color] est un cas spécial (pour $b \geq |V|$).

Pour un graphe G sans- $K_{1,3}$ et une k -coloration de G (donnée par un algorithme quelconque pour [Color]) on déduit une $\max\{k, \lceil |V|/b \rceil\}$ -coloration de G en temps polynomial avec l'algorithme "Gaspillage - Echange bichromatique". Rappelons que $\chi_b(G) = \max\{\chi(G), \lceil |V|/b \rceil\}$. On vérifie que

$$\max\{k, \lceil |V|/b \rceil\} - \chi_b(G) \leq k - \chi(G)$$

et que

$$\max\{k, \lceil |V|/b \rceil\} / \chi_b(G) \leq k / \chi(G)$$

en considérant les trois cas possibles, selon comment s'insère $\lceil |V|/b \rceil$ dans l'inégalité $k \geq \chi(G)$. \square

Pour les graphes généraux, considérons l'algorithme suivant :

Algorithme "Découpage" :

Données : Un graphe G , un entier b et une k -coloration $\{C_1, C_2, \dots, C_k\}$ de G .

Résultat : Une coloration b -bornée de G .

Recolorier chaque couleur C_i avec $\lceil |C_i|/b \rceil$ couleurs de cardinalité au plus b de telle manière qu'au plus une nouvelle couleur utilisée pour C_i ait cardinalité strictement inférieure à b .

Théorème 3.59 *Pour toute classe de graphe \mathcal{G} , $\rho([ColB(\mathcal{G})]) \leq 1 + \rho([Color(\mathcal{G})])$. Cela est atteint par l'algorithme "Découpage" et $1 + \rho$ est une borne asymptotiquement serrée pour la garantie de performance de l'algorithme "Découpage".*

Preuve Etant donnée une k -coloration d'un graphe G , soit k' le nombre de couleurs utilisées par "Découpage". On a $k' \leq \chi_b(G) + k$ car après recoloriage, il y a au plus k couleurs qui ont une cardinalité strictement inférieure à b . On a

$$k' \leq \chi_b(G) + k \leq \chi_b(G) + \rho \chi(G)$$

et donc $k' / \chi_b(G) \leq 1 + \rho$.

La borne est asymptotiquement serrée, comme le montre les graphes complémentaires de la famille de graphes $\{G_b = (V_b, E_b)\}_{b \in \mathbb{N}}$, dont V_b est formé de b ensembles de cardinalité $b + 1$, chacun induisant une clique, tels que chacun des b ensembles contient un sommet spécial unique tels que l'ensemble des sommets spéciaux induise aussi une clique. Il n'y a pas d'autre arêtes dans G_b (voir G_3 dans la Figure 3.18). On a $\bar{\chi}(G_b) = b$ et $\bar{\chi}_b(G_b) = b + 1$. "Découpage" donne une $2b$ -partition en b -cliques de G_b . \square

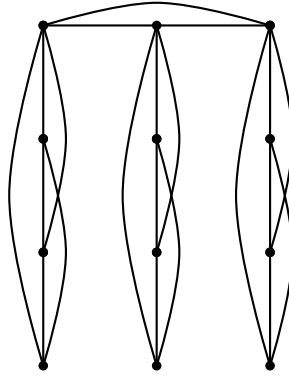


Figure 3.18 – Graphe G_3 de la preuve du Théorème 3.59.

Proposition 3.60 *[ColB] est approximable avec*

- i) $\rho = 2$ sur les graphes parfaits*
- ii) $\rho^* = 4/3$ et $\delta^* = 1$ sur les bipartis.*
- iii) $\rho^* = 4/3$ et $\delta^* = 1$ sur les line-graphes.*

Preuve i) voir Théorème 3.59.

ii) Rappelons qu’il est NP-dur de décider si $\chi_b \leq 3$ pour les graphes bipartis, cela donne $\delta^* \geq 1$ et $\rho^* \geq 4/3$. Soit $G = ((A, B), E)$ un graphe biparti et b un entier. On peut décider si $\chi_b \leq 2$ avec la Proposition 3.57. Si $\chi_b \geq 3$, l’application de l’algorithme “Découpage” avec la coloration (A, B) en entrée donne une coloration qui utilise au plus une couleur de plus que $\chi_b(G)$, d’où $\delta^* = 1$ et $\rho^* \leq 4/3$.

iii) Rappelons qu’il est NP-dur de décider si $\chi \leq 3$ dans les line-graphes, mais que $\delta^*[Color] = 1$ et $\rho^*[Color] = 4/3$ sont atteints par l’algorithme de Vizing ([129], Chapitre 28). Il suffit alors d’appliquer le Corollaire 3.58. \square

3.4.3 Programmes linéaires basés sur BT-mod

Dans cette section, nous montrons comment formuler [PCliqB] par un problème de recouvrement des sommets par les arêtes d’un hypergraphe, et donc un PLNE du type $\min\{\mathbf{1}y \mid Ay \geq \mathbf{1}, y \in \{0, 1\}\}$. Nous observons que ce PLNE n’a pas de bonnes propriétés d’intégralité, mais que la borne BT-mod peut être interprétée comme une famille de coupes valides (et intéressantes) pour cette formulation¹⁰. Le seul usage que nous faisons de cette formulation est de donner un certificat que certains graphes ne satisfont pas BT-mod avec égalité, grâce à une paire de solutions optimales pour les programmes primaux et duaux relaxés.

¹⁰Cette traduction polyédrale ne semble pas digne d’intérêt, car elle mène à une propriété de dualité entière seulement pour des fonctions objectifs spécifiques : même avec la famille de coupes associées à BT-mod, le système considéré ne définit presque jamais un polyèdre entier.

3.4.3.1 Formulation comme un problème de recouvrement

Notons $\mathcal{C}_b(G)$ l'ensemble des b -cliques de G . On peut représenter [PCliqB] par un PLNE de type partition : pour $y \in \mathbb{R}^{\mathcal{C}_b(G)}$, on a $\bar{\chi}_b(G) = \min \mathbf{1}^T y$ tel que

$$(3.9) \quad \begin{aligned} \text{(i)} \quad & y_C \geq 0 \text{ pour tout } C \in \mathcal{C}_b(G) \\ \text{(ii)} \quad & y(v) = 1 \text{ pour tout } v \in V \end{aligned}$$

On peut aussi représenter [PCliqB] par un PLNE de type recouvrement. Soit $(\mathcal{C}_b(G))^{\max}$, le clutter des b -cliques de G qui ne sont pas contenues dans d'autres b -cliques de G . On a $\bar{\chi}_b(G) = \min \mathbf{1}^T y$ tel que

$$(3.10) \quad \begin{aligned} \text{(i)} \quad & y_C \geq 0 \text{ pour tout } C \in (\mathcal{C}_b(G))^{\max} \\ \text{(ii)} \quad & y(v) \geq 1 \text{ pour tout } v \in V \end{aligned}$$

Les relaxations linéaires de ces formulations, en plus de faire intervenir un grand nombre de variables, ne semblent pas valoir la peine d'être étudiées¹¹ après examen de l'Exemple 3.61.

Exemple 3.61 *Pour $G = K_3$ et $b = 2$, on a $\bar{\chi}_2(K_3) = 2$ mais $\min\{1x \mid x \text{ sujet à (3.10)}\} = 3/2$.*

3.4.3.2 Interprétation polyédrale de BT-mod

Reprenons la formulation (3.10) de la couverture de V par des b -cliques, et ajoutons les inégalités BT-mod :

$$(3.11) \quad \begin{aligned} \text{(i)} \quad & y_C \geq 0 \text{ pour tout } (C \in \mathcal{C}_b(G))^{\max} \\ \text{(ii)} \quad & y(U) \geq \lceil |U|/b \rceil \text{ pour tout } U \subseteq V \end{aligned}$$

Les inégalités (3.10)(ii) ne sont plus écrites explicitement puisqu'elles sont des cas particuliers de (3.11)(ii).

Explicitons la validité de la formulation (3.11) :

¹¹On peut vérifier que ces formulations donnent des solutions optimales entières pour tout sous-graphe induit et pour tout b si et seulement si G est biparti.

Proposition 3.62 *Pour tout graphe G et tout $b \in \mathbb{N}_+$, $y \in \{0, 1\}^{\mathcal{C}_b(G)}$ est le vecteur caractéristique d'un recouvrement de G par des b -cliques si et seulement si y satisfait (3.11).*

Preuve Si y entier est solution de (3.11), il est aussi solution de (3.10) et est donc le vecteur caractéristique d'une couverture de V par des b -cliques. Réciproquement, dans une couverture en b -cliques il y a au moins $\lceil |U|/b \rceil$ cliques incidentes à U puisqu'une b -clique n'est incidente qu'à au plus b sommets. \square

Observons que la formulation (3.11) est intimement liée à la borne BT-mod :

Proposition 3.63 *Pour tout graphe G et $b > 0$, on a $\bar{\chi}_b(G) = \sigma_b(G)$ si et seulement si le programme linéaire $\min\{\mathbf{1}^T y \mid y \text{ satisfait (3.11)}\}$ a la propriété de dualité entière. \square*

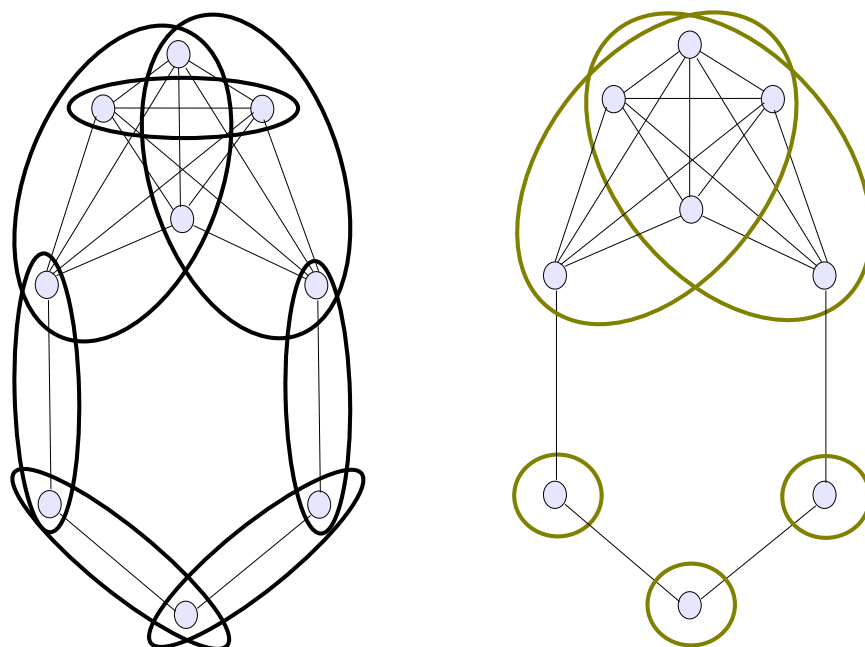


Figure 3.19 – $G = C_6$ avec un sommet répliqué $b = 4$ fois. Solutions primales et duales optimales pour la formulation (3.11). Les ellipses décrivent les variables non-nulles (toutes valent $1/2$). A gauche, une couverture fractionnaire en b -cliques. A droite, un empilement fractionnaire de sous-ensembles de sommets (noter que les deux grandes ellipses ont une valeur objectif de $2 = \lceil 5/b \rceil$).

3.4.4 Applications et complexité de la (co)-réplication

Le sens de la réplication dans le cadre de la partition en b -cliques est le suivant : Un sommet répliqué r fois doit être considéré comme r copies d'un sommet dont les copies peuvent, mais ne doivent pas nécessairement, appartenir au même batch.

De manière analogue si un sommet est co-répliqué, ses copies doivent être mises dans des batchs distincts. Notons que la co-réplication pour la partition en b -cliques correspond à la réplication pour la coloration b -bornée et vice-versa.

L'utilisation explicite de la réplication peut être utile quand les sommets du graphe ne représentent plus des tâches, mais des types de tâches. Le nombre de copies de v définit alors le nombre de tâches de type v . La réplication appliquée à [PCliqB] dans le cas des graphes d'intervalles a été utilisée dans [107] dans le contexte de l'industrie métallurgique. L'Exemple 3.64 donne une application mathématiquement proche de celle de [107]¹².

Exemple 3.64 *L'Université Libre Mondiale (U.L.M.) dispose d'une richesse unique au monde : Un de ses professeurs émérites, Marla Tex, a mis au point cinq copie du premier prototype d'ordinateurs disposant d'un moteur d'Intelligence Emotionnelle Artificielle. Ces ordinateurs sont capables de donner un feed-back en temps réel aux programmeurs sur les erreurs sémantiques qu'ils introduisent dans leurs programmes. Pour cela, l'ordinateur confronte les spécifications implicites à la réalisation du code¹³. De nombreuses SS2I ont payé une fortune pour envoyer quelques employés faire des stages dans l'U.L.M. Chaque SS2I envoie quelques employés en mission, tous pendant un même intervalle de temps. Marla Tex s'est engagée à offrir une formation d'une journée à tous les employés qui effectuent un séjour dans l'U.L.M. Mais son temps est précieux et elle préfère former plusieurs chercheurs à la fois, de manière à minimiser le nombre total de jours de formation. L'université lui laisse la priorité dans l'élaboration du planning des stagiaires. La formation étant principalement constituée d'un TP, elle ne peut encadrer plus de 5 personnes à la fois (ses ordinateurs ne supportent pas encore les incohérences dues aux binômes). Pour minimiser le nombre de jours de formation, elle doit trouver une partition en cliques de taille au plus 5 du graphe ayant les séjours comme sommets (intervalles), ces derniers étant répliqués autant de fois qu'il y a de stagiaires affectés à ce séjour.*

Les graphes d'intervalles étant stables par réplication, Le modèle de l'Exemple 3.64 est donc polynomial si les sommets répliqués sont donnés un par un, mais cela n'est que pseudo-polynomial si les facteurs de réplifications sont codés en binaire. Les algorithmes GAC et S.c-M sont cependant facilement adaptables en algorithmes semi-fortement polynomiaux¹⁴

Voyons maintenant un cas de graphe qui reste borné-parfait par co-réplication et l'application de cela pour des problèmes d'emploi du temps.

Exemple 3.65 *Dans l'Université Libre Mondiale, il y a un ensemble A de professeurs, un ensemble D de groupes d'étudiants (promotions) et un ensemble $E \subset$*

¹²J'ai trouvé plus amusant de décrire une application farfelue. . .

¹³Non, désolé, je n'ai pas trouvé plus farfelu.

¹⁴On commence par remplacer chaque facteur de réplication par son reste modulo b , puis la généralisation est directe.

$A \times D$ de cours (supposés de même durée) qui doivent être donnés. Supposons qu'on cherche le nombre minimum de créneaux horaires pour que tous les cours soient donnés. Cela est équivalent à trouver une $\chi(L(G))$ -coloration du line-graphe $L(G)$ du graphe biparti $G = ((A, D), E)$. Si l'on ne dispose que de b salles, le problème revient à trouver une coloration b -bornée de $L(G)$. Si un même professeur donne plusieurs cours à un même groupe, on peut le modéliser par un multigraphe en parallélisant l'arête $(a, b) \in E$ autant de fois que a doit donner de cours à b . La parallélisation des arêtes de G se traduit dans $L(G)$, par la réplique des sommets.

Les line-graphes de bipartis étant parfaits sans- $K_{1,3}$, le graphe de l'Exemple 3.65 reste parfait sans- $K_{1,3}$ après réplique¹⁵. Le modèle est donc pseudo-polynomial, mais l'existence d'un algorithme semi-fortement polynomial est ouvert et, à notre avis, intéressante pour des raisons similaires à la Question 3.55 page 166.

Voyons maintenant un cas où la borne-perfection est perdue par réplique.

Exemple 3.66 *Les ligues Françaises et Irlandaises de Scrabble organisent une semaine de vacances commune dans une gigantesque station balnéaire. Plusieurs clubs de chaque pays participent à l'évènement. Le premier but étant de faire se rencontrer les membres des deux pays en vue de lancer un championnat multi-langue et de diversifier nos programmes télévisuels. La fédération Française a remarqué que les membres de chaque club français entretiennent une forte aversion envers les membres des autres clubs Français (de même pour les Irlandais entre eux). L'autre but de l'évènement est donc d'inciter les membres de différents clubs à enterrer la hache de guerre en terrain amical. Finalement, pour loger tout ce beau monde, la station balnéaire dispose de k chambre-dortoirs pouvant chacune accueillir au plus b -personnes. Ne voulant pas prendre trop de risques quant à l'expression de l'animosité de leurs membres, les fédérations décident de ne pas mettre deux scrabbleurs venant du même pays mais de clubs différents dans une même chambre. On peut modéliser le problème de répartition dans les chambres par [PCliqB] dans le graphe biparti (complet) ayant l'ensemble des clubs Français et Irlandais respectivement comme cotés de la bipartition, chaque Club étant répliqué autant de fois qu'il a de membres¹⁶.*

Les bipartis complets n'étant pas triangulés, leur répliqués ne sont pas bornés-parfaits. En fait, on a pire : le modèle de l'Exemple 3.66 est fortement NP-dur (cf. Théorème 3.8).

3.4.5 [ColB] pour les arbres et borne-perfection faible

Hansen et al. [93] ont noté la non-trivialité de la complexité de [ColB] restreint aux arbres. Il existe au moins quatre articles apportant une réponse significative à

¹⁵Autrement dit, le complément de ce graphe reste borné-parfait après co-réplique.

¹⁶Non il n'est pas artificiel ce problème!!!

cette question [43, 9, 102, 23].

Proposition 3.67 [9] [ColB] est polynomiale sur les forêts.

Indépendamment [102] prouve la polynomialité pour les arbres ¹⁷. Par ailleurs, [43] donne une formule pour calculer χ_b sur les arbres :

Proposition 3.68 [43] Soit $G = ((A, B), E)$ un arbre, v un sommet de degré maximum dans G et b un entier. Alors

- $\chi_b(G) = 2$ si $b \geq \max\{|A|, |B|\}$
- $\chi_b(G) = \max\{3, \lceil |V|/b \rceil, 1 + \lceil (|V| - \alpha(G - N(v)))/b \rceil\}$ si $b < \max\{|A|, |B|\}$

Dans la formule de la Proposition 3.68, on voit apparaître une idée nouvelle par rapport aux ingrédients de la borne BT-mod (les compléments d'arbres ne sont pas bornés-parfaits comme le montre le graphe G_1 dans la Figure 3.1 page 139). L'essence de cette idée semble être :

Ingrédient 3.69 Soit G un graphe, b un entier et $v \in V$.

$$(3.12) \quad \bar{\chi}_b(G) \geq 1 + \left\lceil \frac{|V| - \omega(N(v))}{b} \right\rceil$$

où $N(v)$ désigne le sous-graphe induit par le voisinage étendu de v .

Preuve $\omega(N(v))$ est un majorant de la cardinalité du batch qui contient v . Il faut donc au moins $\lceil (|V| - \omega(N(v)))/b \rceil$ batches pour les sommets qui ne sont pas dans le même batch que v . \square

Nous proposons une borne inférieure pour $\bar{\chi}_b$ qui généralise à la fois BT-mod et l'Ingrédient 3.69 en vue de donner une formule min-max pour [PCliqB] commune pour les co-arbres et les graphes bornés-parfaits. En fait nous proposons un ingrédient plus fort que 3.69.

Ingrédient 3.70 Pour tout graphe $G = (V, E)$, tout stable S de G et tout b ,

$$(3.13) \quad \bar{\chi}_b(G) \geq \left\lceil \frac{|V| - |\sum_{v \in S} \omega(N(v))|}{b} \right\rceil + |S|$$

Preuve Il faut $|S|$ b -cliques pour couvrir S . Pour $v \in S$, la b -clique qui couvre v , a une taille inférieure ou égal à $\omega(N(v))$. Ensemble, les $|S|$ b -cliques qui couvrent S ne couvrent donc pas plus que $|\sum_{v \in S} \omega(N(v))|$ sommets. \square

L'utilisation de l'Ingrédient 3.70 et son avantage par rapport à 3.69 sont illustrés dans la Figure 3.20.

¹⁷Par la suite un algorithme pour les graphes de largeur d'arbre bornée a été donné [23], mais il ne nous paraît pas clair que l'algorithme soit polynomial en $\log(b)$.

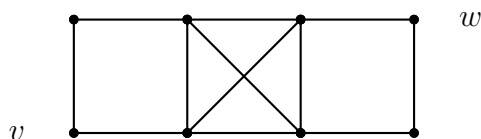


Figure 3.20 – Graphe G satisfaisant $\bar{\chi}_3(G) = 4 > 3 = \sigma_3(G)$ mais dont $\bar{\chi}_3(G)$ est obtenu en appliquant l’Ingrédient 3.70 au stable $\{v, w\}$.

Pour $U \subseteq V$, notons $C_1(U), C_2(U), \dots$ les sous-ensembles de sommets des composantes connexes de $G[U]$. Définissons

$$(3.14) \quad \varsigma_b(G) = \max_{U \subseteq V} \sum_i \max_{S_i \subseteq C_i(U)} \left\{ \left\lceil \frac{|C_i(U)| - |\sum_{v \in S_i} \omega(N_{C_i(U)}(v))|}{b} \right\rceil + |S_i| \right\}$$

où S_i est un stable de $G[C_i(U)]$ et $N_{C_i(U)}(v)$ désigne le sous-graphe induit par le voisinage étendu de v dans $G[C_i(U)]$.

Les Ingrédients 3.70 et 3.69 ne mènent pas forcément à de bonnes caractérisations car ils utilisent le paramètre ω . Cependant,

Proposition 3.71 (Borne BT-mod-faible) *Pour tout graphe G et tout b ,*

$$(3.15) \quad \bar{\chi}_b(G) \geq \varsigma_b(G)$$

De plus décider $\varsigma_b(G) \geq t$ est dans NP si l’on se restreint à une classe de graphes fermés par passage à l’induit, dont la reconnaissance est dans NP et dans laquelle décider $\omega(G) \leq p$ est dans NP (c’est la cas pour les graphes parfaits).

En raison des précautions énoncées dans la Proposition 3.71, nous proposons d’appeler **faiblement borné-parfait**, un graphe G qui est parfait et pour lequel $\bar{\chi}_b(G[U]) = \varsigma_b(G[U])$ pour tout $U \subseteq V$ et tout $b \in \mathbb{N}$. Nous n’effectuons pas d’étude pour ces graphes, mais posons quelques questions :

Question 3.72 *Est-ce que les co-forêts, les graphes clique-séparables et les line-graphes de bipartis (et plus généralement les classes de graphes non-borné-parfaits mais pour lesquels [PCliqB] n’a pas été prouvé être NP-complet) sont faiblement bornés-parfaits ?*

La Figure 3.21 donne un exemple de graphe minimalement non-faiblement borné-parfait.

Mais si $\text{NP} \neq \text{co-NP}$, il devrait aussi exister de petites obstructions, ou du moins, des obstructions appartenant à des classes pauvres de graphes parfaits. En vue d’obtenir une caractérisation des graphes faiblement bornés-parfaits, ou simplement de savoir quelles classes sont susceptibles d’en être des sous-classes, il convient d’étudier la question “duale” de la Question 3.72 :

Question 3.73 *Quels sont les graphes minimaux non-faiblement bornés-parfaits ? Que peut-on dire sur ces obstructions si l’on se restreint aux classes suivantes*

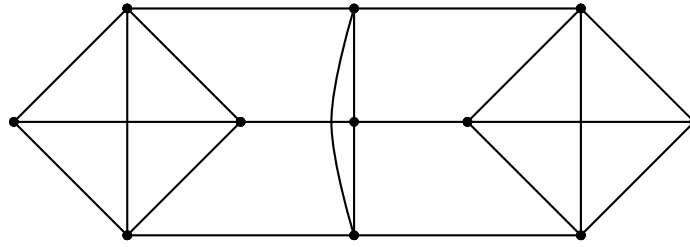


Figure 3.21 – Graphe G tel que $\bar{\chi}_3(G) = 5 > 4 = \varsigma_3(G)$.

- *co-trivialement parfaits*
 - *co-bipartis*
- et plus généralement aux classes pour lesquelles $[PCliqB]$ est NP-complet ?

Un travail de nettoyage important reste donc à effectuer pour y voir plus clair sur la possibilité d’une amélioration de la borne BT-mod¹⁸.

3.4.6 Extensions aux hypergraphes

Nous proposons deux extensions du problème de la coloration bornée aux hypergraphes et nous les discutons dans le cas des matroïdes. Dans la Section 3.4.6.1, nous remplaçons le graphe à colorier par un hypergraphe. Dans la Section 3.4.6.2, nous remplaçons les contraintes de cardinalité par l’appartenance aux arêtes d’un hypergraphe et discutons le problème de la coloration minimum et de l’indépendant commun maximum.

3.4.6.1 Coloration bornée des hypergraphes

Etant donné un hypergraphe $\mathcal{H} = (V, \mathcal{E})$ et un entier b , le nombre chromatique b -borné $\chi_b(\mathcal{H})$ de \mathcal{H} est le nombre minimum de couleurs de cardinalité au plus b qu’il faut pour colorier \mathcal{H} . De manière équivalente $\chi_b(\mathcal{H})$ est le nombre minimum d’arêtes de taille au plus b dans l’hypergraphe $(\text{col}(\mathcal{H}))^\perp$ qu’il faut pour partitionner V .

Dans le cas où \mathcal{E} est l’ensemble des circuits d’un matroïde \mathcal{M} , $(\text{col}(\mathcal{H}))^\perp$ est l’hypergraphe des indépendants. Le théorème d’union des matroïdes implique alors que l’on peut résoudre le problème de “coloration bornée non-uniforme pour les matroïdes”.

¹⁸D’autant plus que l’on pourrait imaginer un ingrédient encore plus fort que 3.70, en remplaçant $|\sum_{v \in S} \omega(N(v))|$ par la cardinalité maximum de l’union de $|S|$ cliques couvrant S . Pour que la borne inférieure sur $\bar{\chi}_b$ ainsi dérivée donne lieu à un bon certificat, il faudrait alors se restreindre à une classe de graphe dans laquelle $[PartialColor]$ est bien soluble, comme les graphes de (co-)comparabilité (voir Section 5.2 pour le problème $[PartialColor]$).

Commençons par la coloration bornée uniforme. L'hypergraphe donné par les indépendants de taille au plus b d'un matroïde \mathcal{M} est encore un matroïde, appelé *b -troncation* de \mathcal{M} . Si r est le rang de \mathcal{M} , le rang de la b -troncation de \mathcal{M} est donné, pour tout $U \subseteq V$ par $\min\{b, r(U)\}$.

On peut donc donner le nombre chromatique borné d'un matroïde :

Proposition 3.74 *Soit $\mathcal{M} = (V, r)$ un matroïde et $b \in \mathbb{N}$. Le nombre minimum $\chi_b(\mathcal{M})$ d'indépendants de taille au plus b qui couvrent V est donné par*

$$\chi_b(\mathcal{M}) = \max_{U \subseteq V} \left\lceil \frac{|U|}{\min\{b, r(U)\}} \right\rceil$$

Preuve Directement en utilisant le théorème de couverture d'un matroïde par des bases ([129], Corollary 42.1c), appliqué à la b -troncation de \mathcal{M} . \square

Pour le cas de bornes non-uniformes,

Proposition 3.75 *Soit $\mathcal{M} = (V, r)$ un matroïde et $\{b_i\}_{i=1}^k$ une liste d'entiers. Il existe une partition de V en k indépendants I_1, \dots, I_k tels que $|I_i| \leq b_i$ si et seulement si*

$$\sum_{i=1}^k \min\{r(U), b_i\} \geq |U|$$

pour tout $U \subseteq V$

Preuve En utilisant le Théorème d'union des matroïdes ([129], Corollary 42.1c) appliqué aux matroïdes $\mathcal{M}_1, \dots, \mathcal{M}_k$, où $\mathcal{M}_i = (V, \min\{r, b_i\})$ est la b_i troncation de \mathcal{M} . \square

3.4.6.2 Indépendants communs d'un graphe et d'un matroïde

La contrainte de cardinalité maximum (au plus b) peut être reformulée comme une contraintes d'indépendance dans le matroïde uniforme \mathcal{U}_n^b . Etant donné un graphe $G = (V, E)$ et un matroïde $\mathcal{M} = (V, r)$, une \mathcal{M} -*clique* est un sous-ensemble de V qui est à la fois une clique de G et un indépendant dans \mathcal{M} . Nous posons les questions de trouver une partition de V en un nombre minimum de \mathcal{M} -cliques ainsi que de trouver la cardinalité maximum d'une \mathcal{M} -clique.

Couverture minimum Notons $\bar{\chi}_{\mathcal{M}}(G)$ le nombre minimum de \mathcal{M} -cliques qu'il faut pour partitionner V . Nous discutons la possibilité d'une formule min-max et la complexité du calcul de $\bar{\chi}_{\mathcal{M}}(G)$.

Nous avons interprété la formule de Berge-Tutte-Gallai (pour la couverture minimum par des arêtes) comme une formule modulo 2. Nous l'avons étendue au cas

modulo- b pour un entier b quelconque en montrant qu'elle donnait une borne inférieure pour $\bar{\chi}_b(G)$. Nous montrons maintenant comment l'étendre du cas modulo- b au cas matroïdal. Pour cela nous utilisons :

Ingrédient 3.76 *Pour tout hypergraphe $\mathcal{H} = (V, \mathcal{E})$ et tout $C \subseteq V$, le nombre minimum d'arêtes de \mathcal{H} nécessaires pour couvrir C est au moins $\lceil |C|/r(C) \rceil$ \square*

Pour tout graphe $G = (V, E)$ et tout matroïde $\mathcal{M} = (V, r)$, notons

$$(3.16) \quad \sigma_{\mathcal{M}}(G) := \max_{U \subseteq V} \sum_i \left\lceil \frac{|C_i(U)|}{r(C_i)} \right\rceil$$

Une raison pour étudier la borne (3.17) est qu'elle donne, dans les cas où elle est serrée, une bonne caractérisation du problème de partition en \mathcal{M} -cliques :

Proposition 3.77 *Pour tout graphe $G = (V, E)$ et tout matroïde $\mathcal{M} = (V, r)$:*

$$(3.17) \quad \bar{\chi}_{\mathcal{M}}(G) \geq \sigma_{\mathcal{M}}(G)$$

De plus, décider $\sigma_{\mathcal{M}}(G) \geq k$ est dans NP (dans le modèle où \mathcal{M} est décrit par un oracle d'indépendance).

Preuve Pour tout $U \subseteq V$, on a

$$\bar{\chi}_{\mathcal{M}}(G) \geq \bar{\chi}_{\mathcal{M}}(G[U]) = \sum_i \bar{\chi}_{\mathcal{M}}(G[C_i(U)]) \stackrel{(3.76)}{\geq} \sum_i \left\lceil \frac{|C_i(U)|}{r(C_i(U))} \right\rceil$$

Puisque les inégalités ci-dessus sont vraies pour tout U , elles le restent en passant au maximum sur $U \subseteq V$.

Un certificat NP consiste en un ensemble $U \subseteq V$ qui réalise $\sigma_{\mathcal{M}}(G)$. Sur chaque composante connexe $C_i(U)$ de $G[U]$, on peut alors calculer un indépendant maximum (*i.e.* de rang $r(C_i)$) avec l'algorithme glouton. \square

La proposition 3.78 rassemble plusieurs formules min-max sous une même forme :

Proposition 3.78 *L'inégalité de "Berge-Tutte matroïdale" (3.17) est serrée dans les cas suivants :*

- i) G est complet et M est quelconque (théorème de couverture d'un matroïde par des bases.)*
- ii) G est borné-parfait et M est uniforme (coloration bornée, borne BT-mod)*
- iii) G est parfait et M est libre (coloration, inégalité $\bar{\chi} \geq \alpha$)*
- iv) G est quelconque et M est uniforme de rang 2 (couverture des sommets d'un graphe par des arêtes, formule de Berge-Tutte-Gallai)*

Malheureusement, cette formule ne semble pas donner une formule min-max dans d'autres cas que ceux de la proposition 3.78 : l'exemple de la Figure 3.22, pour lequel nous sommes redevables à Jacint Szabó et Zsolt Fekete [133] semble exclure toute autre paire de classes (graphe, matroïde) intéressantes. Le cas des matroïdes de partitions généralisées (caractérisés par l'exclusion de $\mathcal{M}(G_2)$ comme mineur) mériterait toutefois une inspection plus détaillée :

Question 3.79 *La borne (3.17) est-elle serrée pour les graphes bornés-parfaits et les matroïdes de partition généralisées ?*

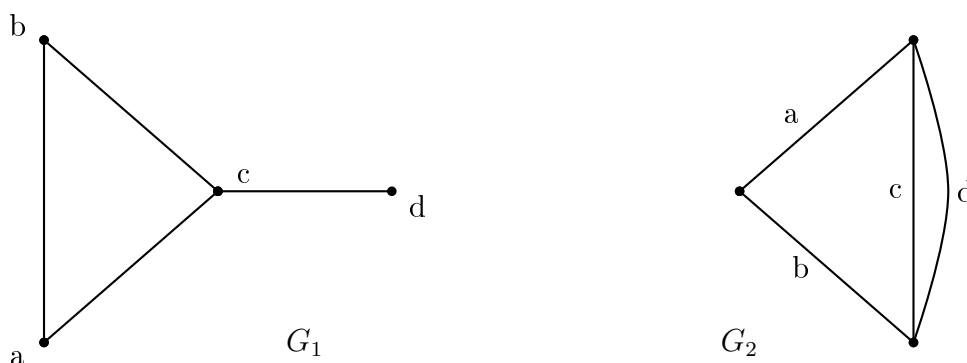


Figure 3.22 – Graphe G_1 et matroïde $\mathcal{M} = \mathcal{M}(G_2)$ (le matroïde des cycles de G_2) pour lesquels $\bar{\chi}_{\mathcal{M}}(G) = 3 > 2 = \sigma_{\mathcal{M}}(G)$.

Bien que la borne (3.17) ne mène pas à une formule min-max, la polynomialité n'est pas nécessairement condamnée pour les matroïdes généraux (et il est donc envisageable que d'autres ingrédients mènent à une formule min-max) :

Proposition 3.80 [133] *Soit $G = (V = (S, K), E)$, un split graphe et $\mathcal{M} = (V, \mathcal{I})$ un matroïde. Le problème de partitionner V en un nombre minimum de \mathcal{M} -cliques se réduit polynomialement au théorème d'union des matroïdes.*

Preuve Si \mathcal{M} a une boucle, la partition est impossible. Sinon, il faut exactement une \mathcal{M} -clique pour couvrir chacun des sommets de S . La \mathcal{M} -clique qui couvre le sommet $s \in S$ est incluse dans l'unique clique qui contient le voisinage étendu de s . Considérons donc pour chaque $s \in S$, le matroïde \mathcal{M}_s , comme la restriction de \mathcal{M} à $N_V^+(s)$. Considérons aussi le matroïde \mathcal{M}_K défini par la restriction de \mathcal{M} à K . On a que $\bar{\chi}_{\mathcal{M}}(G)$ est égal à $|S|$ plus le minimum des $0 \leq k \leq |K|$ tel que l'union des \mathcal{M}_s avec k copies de \mathcal{M}_K soit de rang V . Ce plus petit k peut être déterminé en temps polynomial grâce au théorème d'union des matroïdes. \square

Nous laissons donc comme sujet ouvert de décrire les paires intéressantes (classe de graphes, classe de matroïdes) telles que $\bar{\chi}_{\mathcal{M}}(G)$ peut être déterminé en temps polynomial. En particulier :

Question 3.81 *Le problème de la partition en \mathcal{M} -cliques est-il polynomial pour les graphes triangulés et les matroïdes généraux ?*

Indépendant commun maximum Nous discutons ici le problème de trouver une \mathcal{M} -clique maximum. On peut aisément résoudre ce problème pour les classes de graphes dont on peut énumérer en temps polynomial toutes les cliques maximales : on calcule un indépendant maximum de la restriction de \mathcal{M} à chacune de ces cliques et on compare ensuite les solutions obtenues. D'autres cas moins triviaux sont polynomiaux, comme le poids maximum d'une \mathcal{M} -clique pour un graphe de comparabilité et un matroïde uniforme [131]¹⁹.

En guise de résultat négatif pour le problème de la \mathcal{M} -clique maximum, nous adaptons la réduction de [HamiltonianPath] à [3MatroidInters] :

Proposition 3.82 *Le problème de déterminer une \mathcal{M} -clique de cardinalité maximum est NP-complet même si l'on restreint G aux compléments de line-graphes de bipartis et \mathcal{M} aux matroïdes graphiques (et même planaires).*

Idée de preuve On réduit l'existence d'un chemin Hamiltonien dans un graphe planaire à ce problème. Etant donné un digraphe $D = (V, A)$ on construit le graphe biparti $G = ((V', V''), E)$ dans lequel V' et V'' sont deux copies de V et dont (v', w'') est une arête ssi $(v, w) \in A$ (notons que $A \equiv E$). Soit $\mathcal{M} = (E, \mathcal{I})$ le matroïde des cycles de D dont l'indépendance est exprimée sur E . On vérifie qu'il y a un couplage de G , indépendant dans \mathcal{M} , de cardinalité $|V| - 1$ ssi D a un chemin Hamiltonien. \square

De même, la réduction de [3DM] à [3MatroidInters] nous guide vers :

Proposition 3.83 *Le problème de déterminer une \mathcal{M} -clique de cardinalité maximum est NP-complet même si l'on restreint G aux compléments de line-graphes de multigraphes bipartis et \mathcal{M} aux matroïdes de partition.*

Notons que la Proposition 3.83 n'est pas plus forte que la Proposition 3.82, bien les matroïdes de partition soient des cas particuliers des matroïdes graphiques, car nous ne savons pas si l'on peut se restreindre aux line-graphes de graphes simples dans la Proposition 3.83.

Preuve On réduit la restriction conforme de [3DM] à ce problème. Etant donné trois ensembles V_1, V_2, V_3 de même cardinalité, et $T \subseteq V_1 \times V_2 \times V_3$, soit $G = ((V_1, V_2), T)$ le multigraphe biparti tel que pour tout $t = (v_1, v_2, v_3) \in T$, t est une arête de G incidente à v_1 et v_2 . Soit \mathcal{M} le matroïde de partition sur T dont les classes sont définies par l'incidence entre T et V_3 . On vérifie que G a un couplage indépendant dans \mathcal{M} si et seulement si (V_1, V_2, V_3) admet une partition en éléments de T . \square

¹⁹Problème que l'on peut résoudre grâce à l'algorithme de Ford-Bellman pour trouver le plus long chemin de longueur inférieur ou égal à k dans un graphe orienté acyclique.

3.4.7 Utilisation des forêts clique-connectantes

Dans cette section, nous montrons que les constellations peuvent servir de socle pour modéliser le problème [PCliqB]. Plus précisément, [PCliqB] sur G se réduit au problème d'un indépendant commun du graphe des arêtes dépendantes $D(G, \prec)$ et d'un matroïde de partition généralisée.

La formulation des partitions en b -cliques à l'aide des forêts cliques-connectantes se base sur l'observation suivante : une partition en cliques associée à une forêt clique-connectante est une partition en b -cliques si et seulement si chaque arbre de la forêt contient au plus $b - 1$ arêtes. Supposons donc que l'on ait un ordre quelconque \prec sur $V(G)$. Rappelons qu'une forêt clique-connectantes stellaire bien ordonnée (selon \prec) est aussi appelée constellation de (G, \prec) . De plus, $F \subseteq E$ est une constellation de (G, \prec) si et seulement si c'est un stable de $D(G, \prec)$. Les arbres d'une constellation contiennent au plus $b - 1$ arêtes si et seulement si le degré de chaque sommet dans la forêt est inférieur ou égal à $b - 1$. De cette observation on déduit :

Proposition 3.84 *Pour tout graphe $G = (V, E)$ tout ordre total \prec sur V et tout entier b non-nul,*

$$|V| = \bar{\chi}_b(G) + r(\text{col}(D(G, \prec)) \cap \bigoplus_{v \in V} \mathcal{U}_{\delta_{out}(v)}^{b-1})$$

où $r(\text{col}(D(G, \prec)) \cap (\bigoplus_{\delta_{out}(v)}^{b-1}))$ est la cardinalité maximum d'un indépendant commun du graphe $D(G, \prec)$ et du matroïde de partition généralisé défini comme la somme directe des matroïdes uniformes de rang $b - 1$ définis par l'ensemble des arcs vw pour chaque sommet $v \in V$.

La Proposition 3.84 donne donc un champ d'application au problème de l'indépendant commun de cardinalité maximum dans un graphe et un matroïde discuté dans la Section 3.4.6. A notre connaissance, aucune classe de graphes \mathcal{G} n'est connue pour laquelle cette réduction donne un algorithme polynomial pour [PCliqB] sur \mathcal{G} . Cependant, cette approche et celle proposée dans la Section 4.4.2 pour le problème de la max-coloration mériteraient une inspection ultérieure car elles peuvent être généralisées en une approche commune pour le problème de la max-coloration b -bornée (voir [Pintro] dans le Chapitre 0), motivation originelle de ce mémoire.

Chapitre 4

Batches avec économie d'échelle

Etant donné un graphe $G = (V, E)$ et une fonction $f : \mathcal{P}(V) \rightarrow \mathbb{R}_+$, on cherche une partition en cliques $\{Q_1, Q_2, \dots, Q_k\}$ de G qui minimise $\sum_{i=1}^k f(Q_i)$. De manière équivalente, on cherche à minimiser le makespan d'un ordonnancement par batch avec contraintes de compatibilité données par G et dans lequel le temps d'opération d'un batch est donné par la fonction f . On parle d'économie d'échelle lorsque $f(A \cup B) < f(A) + f(B)$ pour deux ensembles de tâches $A, B \subseteq V$ disjoints. Nous proposons une formulation linéaire en nombres entiers pour le problème ci-dessus, par un problème de partition de coût minimum avec variables sur les cliques de G . Ce programme donne une formule min-max pour certaines classes de graphes et de fonctions d'ensembles. En particulier, on obtient la formule $\bar{\chi}(G) = \alpha(G)$ (si G est parfait et $f \equiv \mathbf{1}$), mais aussi le théorème d'intersection des polymatroïdes, les formules pour les c -couplages bipartis ainsi que la formule de Greene-Kleitman (si f est le rang d'un matroïde uniforme). On obtient aussi de nouveaux cas, notamment pour la fonction $f(U) := \max_{v \in U} p(v)$ qui est le temps d'opération d'une machine max-batch.

Nous traitons indépendamment les graphes d'intervalles, pour lesquels cette approche polyédrale est mise en échec, mais dans lesquels une partition en cliques de coût minimum peut être obtenue par programmation dynamique si f est max-batch ou probabiliste.

Nous discutons finalement le problème de partition de coût minimum avec contraintes de cardinalité, polynomial si f est max-batch mais NP-dur si f est sous-modulaire.

Mots clés : max-coloration, fonction sous-modulaire, graphe d'intervalle, line-graphe de biparti, coeur d'un jeu avec coopération restreinte aux cliques d'un graphe, système (box-)TDI.

Autres objets utilisés : coloration probabiliste, entropie chromatique, graphe parfait, formule de Greene-Kleitman, tournées de véhicules sur un réseau en arbre.

Menu du jour

4.1	L'économie d'échelle : une problématique transversale	184
4.1.1	Philosophie : l'économie d'échelle	185
4.1.2	Max-coloration comme une motivation introductive . . .	188
4.1.3	Coûts de batchs généraux et PLNE	190
4.1.4	Le problème dual : jeux à coopération restreinte	191
4.1.5	Troncation sous-additive d'une fonction partielle	192
4.2	Succès et limites de l'approche jeux coopératifs	193
4.2.1	Fonctions quelconques et graphes bipartis	194
4.2.2	Fonctions sous-modulaires et line-graphes de bipartis . .	195
4.3	Approches "classiques" pour [PCliqBatchGen]	201
4.3.1	Complexité de [PCliqBatchGen] pour des coûts valeur- polymatroïdaux	201
4.3.2	Programmation dynamique sur les graphes d'intervalles	203
4.3.3	TSP-enraciné sur un réseau en arbre	206
4.4	Pour aller plus loin	207
4.4.1	Une approche alternative pour [MaxColor]	207
4.4.2	Le polytope des constellations	208
4.4.3	Cardinalité bornée et max-coloration des matroïdes . . .	210

4.1 L'économie d'échelle : une problématique transversale

Le but de cette section est d'introduire le lecteur aux problèmes d'ordonnement par batchs avec temps d'opérations donnés par une fonction d'ensemble. Nous commençons par évoquer en quoi les temps d'opérations donnés par une fonction d'ensemble élargissent considérablement l'applicabilité des modèles d'ordonnement par batchs (Section 4.1.1). Nous illustrons ensuite le problème général traité dans ce chapitre avec le problème d'ordonnement sur une machine max-batch avec contraintes de compatibilité entre les tâches (Section 4.1.2). Nous proposons une formulation linéaire en nombres entiers pour la minimisation du coût d'une partition en cliques, ce qui est équivalent à la minimisation du makespan d'un ordonnancement par batch avec contraintes de compatibilité (Section 4.1.3). Nous interprétons la dualité forte de la relaxation linéaire de cette formulation comme une loi d'équilibre des prix en théorie des jeux coopératifs avec coopération restreinte (Section 4.1.4).

4.1.1 Petite philosophie à l'usage des (non)-spécialistes : l'optimalité comme un compromis entre la simplicité et l'économie d'échelle

Dans cette section, nous expliquons de manière informelle en quoi l'optimisation d'un ordonnancement par batch résulte d'un équilibre entre des influences contradictoires. La première influence, l'*économie d'échelle*, incite à créer de gros batches, tandis que de nombreuses autres influences tendent au contraire à en limiter la taille. Dans la prochaine sous-section, nous évoquons l'universalité de ce phénomène et dans la sous-section suivante nous le spécialisons au cadre particulier que nous explorons dans ce chapitre.

4.1.1.1 Un phénomène d'équilibre universel

Exemple 4.1 *Le "Café Archimède" dispose d'une grande terrasse. L'unique serveuse est poussée à optimiser l'efficacité de son service car le nombre de clients est en perpétuelle augmentation. En particulier, pour minimiser la distance qu'elle parcourt, elle livre souvent plusieurs boissons à la fois. Évidemment, plus elle sert de tables avant de retourner au comptoir, plus elle économise sur la distance totale parcourue. Mais d'autres facteurs l'obligent au contraire à effectuer de petites tournées :*

- *Son plateau ne peut porter qu'un nombre limité de boissons à la fois,*
- *Plus elle porte de boissons, plus la gestion de la tournée devient complexe,*
- *Si elle porte trop de boissons, elle court le risque de les renverser,*
- *Le délai entre une commande et le service doit rester raisonnable.*

En pratique, on observe que la "bonne stratégie" de service suit la loi suivante :

Le nombre de boissons servies par tournée augmente avec l'intensité des commandes.

Evidemment, au-delà d'une certaine intensité de commandes, une unique serveuse ne sera plus capable de toutes les satisfaire et il faudra en engager une autre. Supposons maintenant qu'il y ait n serveurs qui travaillent. Si l'intensité des commandes augmente, la patronne du café aura recours à une autre stratégie avant d'engager un $n + 1$ ème serveur : la spécialisation. En effet, à nombre fixé de boissons servies par tournées, l'économie de distance augmente quand le nombre de tables visitées diminue ainsi que quand ces tables sont proches les unes des autres. Autrement dit, plus il y a de serveurs, plus la zone que chacun couvrira sera petite :

La spécialisation augmente avec l'intensité des commandes.

Un *batch* est une tournée effectuée par le serveur entre deux retours au comptoir. Une *tâche* est une boisson qui doit être servie. L'augmentation de la taille des batches et la spécialisation des batches en groupes de tâches similaires sont deux

aspects d'un concept plus général : "l'économie d'échelle". Pour schématiser, on peut formaliser l'économie d'échelle de la manière suivante : le temps (ou coût) d'opération $f(A \cup B)$ de deux ensembles A et B disjoints de tâches est plus petit que $f(A) + f(B)$. La validité d'une telle inégalité ne dépend pas du fait que l'organisation en charge du traitement des tâches ait pour vocation d'extraire des matières premières, de produire des voitures, de transporter des abricots, de traiter des données informatiques, de créer des logiciels ou même des théorèmes. En fait, l'économie d'échelle est un des facteurs expliquant la réussite économique des grandes organisations.

Cependant, un serveur n'attend pas toujours que son plateau soit plein pour effectuer une tournée, même pendant une période de pointe. De même, l'efficacité d'une entreprise ne dépend pas que de son aptitude à tirer parti de l'économie d'échelle. Dans l'industrie automobile par exemple, au succès du "Fordisme" a succédé celui du "juste-à-temps". Si dans le premier c'est l'économie d'échelle qui fait force de Loi, dans le second, on cherche plutôt à améliorer la flexibilité et la réactivité. Ce n'est pas parce que le "juste-à-temps" a été promu après le "Fordisme" que c'est une meilleure doctrine. Il semble plutôt que le Fordisme ait été une réponse adaptée à un contexte de forte croissance et que le juste-à-temps soit une réponse adaptée à des contextes économiques plus diversifiés et moins prévisibles.

En général, la réussite économique dépend donc d'un équilibre entre des forces antagonistes. Si la simplicité, la flexibilité et la réactivité d'une entreprise sont des facteurs déterminants de viabilité économique, ces qualités sont difficiles à préserver quand la taille de l'organisation augmente et donc souvent incompatibles avec la maximisation de l'économie d'échelle. Cet équilibre se traduit souvent par une forme de spécialisation jumelée au partenariat avec d'autres entreprises à vocation complémentaire (par exemple, la chaîne de relations fournisseurs-clients, la sous-traitance ou encore les filiales par zones géographiques...)¹.

Ce type d'équilibre apparaît aussi dans l'organisation du travail des chercheurs. L'attrait pour l'amélioration, la généralisation et l'unification des théories pousse à la transdisciplinarité et aux projets ambitieux, qui permettent des avancées majeures... lorsque ces projets aboutissent. D'un autre côté, les limites de temps et de communication poussent à la spécialisation des connaissances et à la simplification des questionnements, lesquelles assurent une production régulière de résultats... mais qui peut, à long terme, dégénérer en consanguinité stérile.

¹Dans d'autres contextes, on peut aussi chercher le bon compromis entre l'économie d'échelle générée par une direction globale (par exemple nationale) et les avantages de souplesse que permet un système décentralisé...

4.1.1.2 Et l'ordonnement par batches dans tout ça ?

Un équilibre similaire entre forces antagonistes a lieu dans les problèmes d'optimisation combinatoire² ou de recherche opérationnelle, en particulier en ordonnancement par batch. En effet, si l'économie d'échelle incite à ordonnancer de gros batches, d'autres facteurs poussent au contraire à en limiter la taille. Les facteurs limitants que nous discutons dans ce chapitre, sont des "contraintes dures", comme l'incompatibilité du traitement simultané de certaines tâches (*i.e.* les contraintes de coloration), ou encore la capacité de traitement des machines (*i.e.* les contraintes de cardinalité, voir Section 4.4). D'autres facteurs participent à la limitation de la taille des batches, la réactivité par exemple, qui s'exprime au travers d'une fonction objectif comme $\sum C_j$ ou $\sum U_j$ (cela est évoqué dans le Chapitre 5).

Dans le présent chapitre, nous étudions la minimisation du makespan C_{max} sous contraintes d'incompatibilité. L'utilisation de cet objectif traduit une motivation simple : la minimisation des coûts opérationnels. De manière intéressante, l'optimisation de cet objectif est lié à une forme d'équilibre : la loi de l'offre et de la demande. Modulo certaines simplifications, cette loi s'exprime plus précisément en théorie des jeux coopératifs comme suit. *Dans un contexte où la concurrence et l'information sont parfaites, le prix maximum que l'on peut faire payer aux clients est égal au coût minimum nécessaire à une entreprise pour les servir.* Techniquement parlant, cela revient à généraliser le théorème de Bondareva-Shapley au contexte d'une coopération restreinte aux arêtes d'un hypergraphe. Si la version continue de cette généralisation (*i.e.* le problème de partition fractionnaire de coût minimum) n'est qu'une simple application de la dualité en programmation linéaire [74], c'est la version entière cependant (*i.e.* le problème de partition de coût minimum) qui est pertinente dans les applications. Bien que sujette à de nombreuses obstructions, la dualité forte dans le cas entier offre un formalisme commun pour exprimer plusieurs résultats de combinatoire polyédrale (Section 4.2). On montre que la validité de cette dualité forte ((4.3) page 191) implique de se restreindre aux hypergraphes conformes. De plus, avec une perte de généralité qui semble relativement faible on peut dire que cette dualité forte nécessite aussi de se restreindre aux hypergraphes héréditaires, donnés par les cliques d'un graphe parfait ainsi qu'aux fonctions de coût sous-modulaires.

Curieusement, l'approche étudiée ici (*i.e.* coopération restreinte aux cliques d'un graphe) pour obtenir une formule min-max à l'aide du coeur des jeux coopératifs ne semble pas avoir été proposée dans la littérature. D'autres approches ont

²Le terme d'équilibre peut paraître inapproprié car l'optimisation combinatoire traite d'optimisation dans un domaine de réalisabilité. Mais un autre point de vue modélise les contraintes fortes en les relaxant en contrepartie d'un coût (très élevé si la contrainte est très forte). Le phénomène d'équilibre apparaît donc lorsque l'on considère la relaxation Lagrangienne des problèmes considérés.

par exemple été proposées pour une coopération restreinte aux indépendants d'un matroïde [19]. Le lien entre les jeux coopératifs et les flots sous-modulaires est mentionné dans [74].

4.1.2 Max-coloration comme une motivation introductive

Une machine *max-batch* est une machine par batches dont le temps $f(U)$ de traitement d'un ensemble de tâches $U \subseteq V$ est donné par $f(U) := \max_{u \in U} p(u)$. Soit $G = (V, E)$ un graphe et $p : V \rightarrow \mathbb{N}$. Le problème de la *max-coloration* consiste dans le traitement des sommets de V sur une machine max-batch avec des contraintes d'incompatibilité données par les arêtes de G . L'objectif est donc de trouver une coloration $\{S_1, S_2, \dots, S_k\}$ de G qui minimise le makespan. Celui-ci s'exprime comme la somme des temps d'opération des différents batches :

$$C_{\max} := \sum_{i=1}^k f(S_i)$$

Le makespan ne dépend donc pas de l'ordre dans lequel les batches sont ordonnancés, mais seulement de la manière dont on partitionne V . Ce problème d'ordonnancement est donc un problème de partition de coût minimum. Le problème [PCliqMax] est équivalent au problème de la max-coloration en passant au graphe complémentaire.

[PCliqMax] Partition en cliques sur une machine max-batche

Données : Un graphe $G = (V, E)$ et une fonction $p : V \rightarrow \mathbb{N}$.

Résultat : Une partition en cliques $\{C_1, C_2, \dots, C_k\}$ de G

Mesure : $\sum_{i=1}^k (\max_{v \in C_i} p(v))$ à minimiser.

Commentaires : Le problème de coloration équivalent en passant au graphe complémentaire est noté [MaxColor].

Notons $\bar{\chi}(G, p)$ la valeur optimum d'une partition en cliques pour le problème [PCliqMax]. $\bar{\chi}(G, p)$ est lié à la taille d'un stable de poids maximum dans le graphe pondéré (G, p) :

Proposition 4.2 [65] *Pour tout graphe pondéré (G, p) , on a*

$$(4.1) \quad \bar{\chi}(G, p) \geq \alpha(G, p)$$

Preuve Soit $(G = (V, E), p)$ un graphe pondéré et S un stable de G , on a :

$$\bar{\chi}(G, p) \geq \bar{\chi}(G[S], p) = \sum_{v \in S} p(v)$$

où la première inégalité vient du fait que toute partition en cliques de G en induit une de coût inférieur ou égal sur tout sous-graphe induit. On obtient (4.1) en choisissant un stable de poids maximum pour S . \square

Cependant, même dans les graphes sans- P_4 (pour lesquels [PCliqBatchGen] est polynomial [65]), la borne (4.1) n'est pas serrée [65] (voir aussi Figure 4.1).

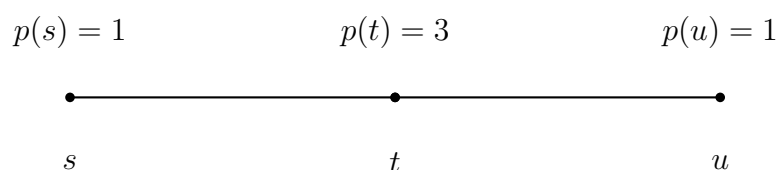


Figure 4.1 – Une pondération du graphe P_3 telle que $4 = \bar{\chi}(P_3, p) > \alpha(P_3, p) = 3$.

Nous revenons sur l'inégalité (4.1) dans la Section 4.4.1. Un des buts de ce chapitre est de développer une meilleure borne inférieure pour $\bar{\chi}(G, p)$. Cette borne inférieure se base sur une autre interprétation de la formulation linéaire en nombres entiers associée à la théorie des graphes parfaits : le problème [PCliqMax] peut être écrit comme un problème de partition de coût minimum ayant l'ensemble des cliques de G comme variables, l'ensemble des sommets de G comme contraintes et tel que le coût $f(C)$ d'une clique C est donné par $\max_{v \in C} p(v)$. Nous écrivons maintenant ce programme linéaire pour le graphe pondéré de la Figure 4.1.

$$\begin{aligned}
 & \min y_s + 3y_{st} + 3y_t + 3y_{tu} + y_u \\
 & y_s + y_{st} = 1 \\
 & y_{st} + y_t + y_{tu} = 1 \\
 & y_{tu} + y_u = 1 \\
 & y_C \in \{0, 1\} \text{ pour toute clique } C \text{ de } G
 \end{aligned}$$

Une solution optimale de ce programme est donnée par $y_s = y_{tu} = 1$. Après avoir relaxé les contraintes $y_C \in \{0, 1\}$ en $y_C \geq 0$ le programme dual donne un certificat d'optimalité de la partition $\{\{s\}, \{t, u\}\}$: le vecteur $x_s = x_u = 1$ et $x_t = 2$, qui est une solution optimale du programme :

$$\begin{aligned}
& \max x_s + x_t + x_u \\
& x_s \leq 1 \\
& x_s + x_t \leq 3 \\
& x_t \leq 3 \\
& x_t + x_u \leq 3 \\
& x_u \leq 1
\end{aligned}$$

4.1.3 Coûts de batchs généraux et PLNE

Le problème [PCliqBatchGen] suivant consiste à généraliser le problème [PCliqMax] au cas de temps d'opérations quelconques. Pour un graphe G , on note $\mathcal{C}(G)$ l'ensemble des cliques (non-vides) de G .

[PCliqBatchGen] Partition en cliques sur machine par batchs

Données : Un graphe $G = (V, E)$ et un oracle de valeurs, qui fournit $f(C)$ en temps constant pour chaque $C \in \mathcal{C}(G)$.

Résultat : Une partition en cliques $\{C_1, C_2, \dots, C_k\}$ qui minimise $\sum_{i=1}^k f(C_i)$.

Nous formulons [PCliqBatchGen] par un programme linéaire en nombres entiers et plus précisément comme un problème de partition.

$$\begin{aligned}
(4.2) \quad (i) \quad & \min f^T y \\
(ii) \quad & \sum_{C \ni v} y_C = 1 \text{ pour tout } v \in V \\
(iii) \quad & y_C \in \{0, 1\} \text{ pour tout } C \in \mathcal{C}(G)
\end{aligned}$$

Dans la plupart des applications, f est non-négative et non-décroissante. Dans ce cas, on peut à loisir relaxer les contraintes (4.2)(ii) en $\sum_{C \ni v} y_C \geq 1$ et les contraintes (4.2)(iii) en $y_C \in \mathbb{N}$ sans changer la valeur de l'optimum de (4.2). En effet, si y est une solution de ce programme relaxé, on peut réaffecter $y_A := 1$ pour tout $A \in \mathcal{C}(G)$ tel que $y_A > 1$. La solution reste réalisable et puisque f est non-négative, la valeur objectif n'augmente pas. Ensuite, si $y_A = y_B = 1$, avec $A, B \in \mathcal{C}(G)$ tels que $A \cap B \neq \emptyset$ alors $B \setminus A$ est encore une clique de G et nous pouvons réaffecter $y_B := 0$ et $y_{B \setminus A} := 1$. La solution reste réalisable et puisque f est non-décroissante, la valeur objectif n'augmente pas.

4.1.4 Le problème dual : jeux à coopération restreinte

Indépendamment des propriétés de f , on peut relaxer les contraintes (4.2)(iii) en $y_C \in \mathbb{N}$, car les contraintes (4.2)(ii) impliquent que l'ensemble des solutions est inchangé. Après cette transformation, le dual de la relaxation linéaire de (4.2) est donc de maximiser $\mathbf{1}^T x$ sujet à :

$$(4.3) \quad \sum_{v \ni C} x_v \leq f(C) \text{ pour tout } C \in \mathcal{C}(G)$$

Si f est non-négative et non-décroissante, on peut relaxer (4.2), sans changer la valeur objectif de la manière suivante :

$$(4.4) \quad \begin{aligned} \text{(i)} \quad & \min f^T y \\ \text{(ii)} \quad & \sum_{C \in v} y_C \geq 1 \text{ pour tout } v \in V \\ \text{(iii)} \quad & y_C \in \mathbb{N} \text{ pour tout } C \in \mathcal{C}(G) \end{aligned}$$

Le dual de la relaxation linéaire de (4.4) est de maximiser $\mathbf{1}^T x$ sujet à :

$$(4.5) \quad \begin{aligned} \text{(i)} \quad & \sum_{v \in C} x_v \leq f(C) \text{ pour tout } C \in \mathcal{C}(G) \\ \text{(ii)} \quad & x_v \geq 0 \text{ pour tout } v \in V(G) \end{aligned}$$

Si G est parfait et $f \equiv 1$, (4.5) est TDI. Aussi, si G est complet et f est sous-modulaire (4.3) est box-TDI. Dans les deux cas, on obtient donc une formule min-max pour [PCliqBatchGen] à partir de la relaxation linéaire de (4.2). Dans la Section 4.2, nous explorons d'autres paires (G, f) pour lesquelles le système (4.3) a de bonnes propriétés d'intégralité.

Le système (4.5) peut être interprété en théorie des jeux coopératifs dans le cadre d'une coopération restreinte : seuls les ensembles de joueurs qui induisent une clique peuvent coopérer (ou de manière équivalente, la valeur d'une coalition qui n'induit pas une clique est $+\infty$). Une répartition des coûts $x \in \mathbb{R}^V$ est donc stable si et seulement si elle respecte les contraintes (4.3). Si le coeur de la fonction f est vide dès que G n'est pas une clique (car alors $f(V) = +\infty$), le système (4.3) est pourtant TDI dans de nombreux cas où G n'est pas une clique. Comme nous l'avons argumenté dans les Sections (1.4.1.4) et (1.4.1.5), la stabilité (au sens de

la théorie des jeux) des solutions du système (4.3) lorsque celui-ci est TDI incite à étudier les cas dans lesquels les troncations sous-additives et totalement-équilibrées de f sont égales (plutôt que le simple sous-cas où f est totalement-équilibrée).

4.1.5 Troncation sous-additive d'une fonction partielle

Dans cette section, nous associons deux fonctions auxiliaires à f définies sur $\mathcal{P}(V)$: la troncation sous-additive et la troncation totalement-équilibrée. Nous expliquons pourquoi l'égalité de ces troncations est fondamentale aussi bien en théorie des jeux coopératifs qu'en combinatoire polyédrale. Nous supposons que $f : \mathcal{C}(G) \rightarrow \mathbb{R}$ est non-négative et non-décroissante³.

Pour $U \subseteq V$, nous notons $\widehat{f}_G(U)$ le coût minimum d'une partition en cliques de $G[U]$:

$$(4.6) \quad \widehat{f}_G(U) := \min \left\{ \sum_{P \in \mathcal{P}} f(P) \mid \mathcal{P} \text{ est une partition de } G[U] \text{ en cliques} \right\}$$

De manière équivalente, on peut considérer l'extension $f' : \mathcal{P}(V) \rightarrow \mathbb{R} \cup \{+\infty\}$ de f telle que $f'(U) := f(U)$ pour $U \in \mathcal{E}$ et $f'(U) := +\infty$ pour $U \notin \mathcal{E}$.

$$(4.7) \quad \widehat{f}'(U) := \min \left\{ \sum_{P \in \mathcal{P}} f'(P) \mid \mathcal{P} \text{ est une partition de } U \text{ en ensembles non-vides} \right\}$$

Clairement, $\widehat{f}_G(U) = \widehat{f}'(U)$ pour tout $U \subseteq V$. D'après la Proposition 1.40, \widehat{f}' est la plus grande fonction sous-additive inférieure ou égale à f' , c'est pourquoi nous l'appelons la troncation sous-additive de f_G . De manière similaire, pour $U \subseteq V$, on note $\overline{f}(U)$ le coût minimum d'une partition fractionnaire de $G[U]$ en cliques.

$$(4.8) \quad \overline{f}_G(U) := \min \left\{ \sum_{P \in \mathcal{P}} f(P) \mid \mathcal{P} \text{ est une partition fractionnaire de } G[U] \text{ en cliques} \right\}$$

Clairement, on a aussi $\overline{f}_G = \overline{f}'$. D'après la Proposition 1.39, \overline{f}' est la plus grande fonction totalement-équilibrée inférieure ou égale à f' , c'est pourquoi nous l'appelons la troncation totalement-équilibrée de f_G .

Du point de vue de la complexité algorithmique, la question adressée par ce chapitre est la difficulté du calcul de $\widehat{f}_G(V) = \overline{\chi}(G, f)$. La question adressée habituellement par la combinatoire polyédrale est de déterminer pour quelles paires (G, f) le système (4.3) est TDI ou box-TDI. Une autre question naturelle est de déterminer

³Supprimer ces hypothèses nécessiterait probablement un examen minutieux et sûrement des définitions plus générales qui compliquent l'interprétation en termes de théorie des jeux. D'autre part, comme nous l'avons vu dans la Section 1.4.1.3, traiter parallèlement les résultats pour B_f et EB_f mène plutôt à des lourdeurs qu'à une meilleure compréhension.

les cas où le coût minimum d'une partition de G en cliques est égal à $\max \mathbf{1}^T x$ sujet à (4.3) (x n'étant pas nécessairement entier). En effet, cela semble mener à une bonne caractérisation pour le problème [PCliqBatchGen]. Cependant, même dans le cas où $f = \mathbf{1}$, cela ne donne pas une bonne caractérisation puisque c'est équivalent à l'égalité $\bar{\chi}(G) = \bar{\chi}^*(G)$ (Proposition 1.45). Demander l'égalité pour tout sous-graphe induit est donc une question naturelle à la fois en combinatoire polyédrale et en théorie des jeux coopératifs :

Remarque 4.3 Soit G un graphe et $f : \mathcal{C}(G) \rightarrow \mathbb{R}_+$ non décroissante. Les conditions suivantes sont équivalentes :

- $\bar{\chi}(H, f) = \max \mathbf{1}^T x$ sujet à (4.5) pour tout sous-graphe induit H de G ,
- Les troncations sous-additive \widehat{f}_G et totalement équilibrée \overline{f}_G de f sont égales.

4.2 Succès et limites de l'approche jeux coopératifs

Dans cette section, nous étudions plusieurs classes de fonctions. Pour chaque classe \mathcal{F} , nous cherchons à déterminer les graphes G pour lesquels le système (4.3) donne une formule min-max pour le coût minimum $\bar{\chi}(G, f)$ d'une partition de G en cliques pour tout $f \in \mathcal{F}$. Le Théorème type que nous visons est illustré dans la Proposition 4.4. Il s'agit d'une équivalence entre des propriétés de théorie des graphes et des propriétés polyédrales liées à la classe \mathcal{F} considérée. L'équivalence se base sur la construction d'une chaîne d'implications qui met en sandwich des propriétés polyédrales entre des propriétés relevant de la théorie des graphes. Le pas difficile (lorsqu'il est possible), consiste à boucler la chaîne d'implications à l'aide d'une méthode de théorie des graphes. Idéalement donc, la première condition est de type *décomposition* et la dernière est de type *sous-graphe exclus*⁴. Nous discutons quelques classes de fonctions \mathcal{F} relativement naturelles, qui pourraient mener à des classes de graphes intéressantes. Malheureusement, dans la plupart des cas, les résultats sont largement inachevés et nous ne pouvons pas encore ne serait-ce que formuler de conjectures raisonnables.

L'intérêt des résultats discutés dans la Section 4.2 repose en partie sur le fait que pour certaines classes de fonctions \mathcal{F} , on a pour tout graphe G :

$$\widehat{f}_G = \overline{f}_G \text{ pour tout } f \in \mathcal{F} \iff (4.5) \text{ est TDI pour tout } f \in \mathcal{F}$$

L'équivalence de ces deux propriétés a par exemple lieu pour la classe $\mathcal{F} = \{\mathbf{1}\}$

⁴C'est le schéma que semble avoir tenté d'appliquer Paul Seymour pour plusieurs des preuves monumentales auxquelles il a contribué (matrices TTU [128], hypergraphes binaires Mengériens [129], graphes parfaits). On a parfois tendance à mentionner (*i.e.* dans les enseignements) ce genre de caractérisation seulement sous l'angle des obstructions (graphes parfaits, hypergraphes Mengériens) ou seulement sous l'angle des décompositions (matrice TTU), alors que les deux points de vue sont largement complémentaires.

et cela caractérise les graphes parfaits (Corollaire 1.46 page 65, et Théorème 2.4 page 101).

4.2.1 Fonctions quelconques et graphes bipartis

Dans cette section, nous observons que les graphes bipartis sont intimement liés à la classe des fonctions quelconques.

Proposition 4.4 *Soit G un graphe, les conditions suivantes sont équivalentes :*

- i) G est biparti,*
- ii) La matrice des contraintes du système (4.3) est TTU,*
- iii) (4.3) est box-TDI pour tout $f : \mathcal{C}(G) \rightarrow \mathbb{R}$,*
- iv) $\widehat{f}_G = \overline{f}_G$ pour tout $f : \mathcal{C}(G) \rightarrow \{0, 1, 2\}$ non-décroissante,*
- v) G n'a pas de cycle impair (induit).*

Preuve $v) \iff i) \iff ii)$ voir [129].

$i) \implies ii)$ La matrice d'incidence sommets-arêtes d'un graphe biparti est totalement unimodulaire [129]. L'ensemble des cliques d'un graphe biparti étant égal à l'ensemble de ses arêtes plus l'ensemble de ses sommets, la matrice d'incidence sommets-cliques est elle aussi totalement unimodulaire.

$ii) \implies iii)$ Voir Théorème 1.22 page 44.

$iii) \implies iv)$ est trivial.

$iv) \implies v)$ Supposons que G contienne un cycle impair C_n . Considérons la fonction f_n définie sur les cliques de C_n par

$$(4.9) \quad f(U) := \begin{cases} 0 & \text{si } U = \emptyset \\ 1 & \text{si } U \text{ est une clique de taille 1 ou 2} \\ 2 & \text{si } U = K_3 \end{cases}$$

Le fonction f est définie pour chaque clique K de G par $f(K) := f_n(C_n \cap K)$. On a alors que $\max\{\mathbf{1}^T x \mid x \text{ satisfait (4.3)}\} = n/2$ est atteint par $x = 1/2 \times \mathbf{1}_{V(C_n)}$. \square

De plus, dans les graphes bipartis, [PCliqBatchGen] est équivalent à trouver un couplage de poids maximum (en donnant le poids $f(\{u\}) + f(\{v\}) - f(\{u, v\})$ à l'arête (uv)). Plus généralement, on peut énoncé un résultat qui a été publié tant de fois qu'il serait fastidieux de le créditer à un groupe d'auteurs :

Proposition 4.5 (Folklore) *Soit $G = ((A, B), E)$ un graphe biparti, $f : E \cup A \cup B \rightarrow \mathbb{N}$ et $w : A \cup B \rightarrow \mathbb{N}$. Alors le coût minimum d'une w -couverture de $A \cup B$ par des cliques est égal au maximum de $w^T x$ sujet à (4.5). Une couverture en cliques de coût minimum est équivalente à trouver un w -couplage de poids maximum pour la pondération $f(\{u\}) + f(\{v\}) - f(\{u, v\})$ associée à chaque arête (uv) .*

Preuve La formule min-max est une simple reformulation de la propriété TDI du système (4.5). Montrons la réduction au problème du c -couplage de poids maximum. Un ensemble de cliques qui couvre $w(v)$ fois chaque sommet peut être vu comme une fonction $y : A \cup B \cup E \rightarrow \mathbb{N}$ qui satisfait

$$y_v + \sum_{e \ni v} y_e = w(v) \text{ pour tout } v \in A \cup B$$

Autrement dit, on a $y_v = w(v) - \sum_{e \ni v} y_e$ pour tout v . Une w -couverture s'exprime donc indépendamment des variables y_v pour $v \in A \cup B$. De plus $y_v \geq 0$ si et seulement si $\sum_{e \ni v} y_e \leq w(v)$ c'est-à-dire si y est un w -couplage. En choisissant les poids annoncés pour les arêtes, on a que le poids de toute w -couverture y plus le poids du w -couplage défini par y est constant ($\sum_{a \in A} w(a)f(a) + \sum_{b \in B} w(b)f(b)$). En effet, le coût d'une w -couverture est

$$\begin{aligned} & \sum_{a \in A} y_a f(a) + \sum_{b \in B} y_b f(b) + \sum_{e \in E} y_e f(e) \\ &= \sum_{a \in A} \left(w(a) - \sum_{e \ni a} y_e \right) f(a) + \sum_{b \in B} \left(w(b) - \sum_{e \ni b} y_e \right) f(b) + \sum_{e \in E} y_e f(e) \\ &= \left(\sum_{a \in A} w(a)f(a) + \sum_{b \in B} w(b)f(b) \right) + \sum_{ab \in E} (f(ab) - f(a) - f(b)) y_{ab} \end{aligned}$$

□

4.2.2 Fonctions sous-modulaires et line-graphes de bipartis

Dans cette section, nous étudions la classe des fonctions sous-modulaires. Si les line-graphes de multi-graphes bipartis jouent un rôle fondamental relativement à cette classe de fonctions, quelques graphes exotiques se comportent bien vis-à-vis du système (4.3). A défaut de caractériser ces graphes, nous offrons une liste (que nous conjecturons complète) de graphes pour lesquels (4.3) n'a pas de bonnes propriétés d'intégralité.

4.2.2.1 Fonction sous-modulaire sur une famille héréditaire

Proposition 4.6 *Soit \mathcal{E} une famille héréditaire sur V et $f : \mathcal{E} \rightarrow \mathbb{R}$. Les conditions suivantes sont équivalentes :*

- i) Il existe $g : \mathcal{P}(V) \rightarrow \mathbb{R}$ sous-modulaire telle que $f(A) = g(A)$ pour tout $A \in \mathcal{E}$.*
- ii) $f(S \cap T) + f(S \cup T) \leq f(S) + f(T)$ pour tous $S, T \subseteq V$ tels que $(S \cup T) \in \mathcal{E}$.*

iii) $f(S) \leq f(S - t) + f(S - u) - f(S - t - u)$ pour tous $S \subseteq \mathcal{E}$ et $t \neq u \in S$.

Preuve i) \implies ii) : Si g est une extension sous-modulaire de f , alors g vérifie en particulier la condition de sous-modularité sur les paires (S, T) dont l'union est dans \mathcal{E} .

ii) \implies iii) : Appliquer ii) aux ensembles $S + t$ et $S + u$.

iii) \implies i) : S.p.d.g. f est définie sur tous les singletons de V (sinon, en prenant une valeur quelconque, par exemple $f(s) := 0$, iii) reste vérifiée). Définissons une fonction f' de la manière suivante. Soit $<$ une extension linéaire de l'ordre \subseteq sur $\mathcal{P}(V)$, telle que \mathcal{E} soit un idéal inférieur de $(\mathcal{E}, <)$. Définissons les valeurs de f' récursivement selon l'ordre $<$.

(4.10)

$$f'(S) := \begin{cases} f(S) & \text{si } |S| \leq 1 \\ \min\{f(S), \min_{t \neq u \in S} \{f'(S - t) + f'(S - u) - f'(S - t - u)\}\} & \text{sinon} \end{cases}$$

D'après le Théorème 1.1 page 19, pour toute fonction f , la fonction f' est une fonction sous-modulaire⁵. Par ailleurs, si f vérifie iii), on a $f'(S) = f(S)$ pour tout $S \in \mathcal{E}$, donc f' est une extension de f . \square

Si \mathcal{E} est une famille héréditaire, une fonction $f : \mathcal{E} \rightarrow \mathbb{R}$ est **sous-modulaire (sur \mathcal{E})** si elle vérifie les conditions de la Proposition 4.6.

4.2.2.2 Les line-graphes de bipartis

Nous montrons que le système (4.3) est TDAU dans le cas où G est un line-graphe de multigraphe biparti et f est une fonction sous-modulaire. En fait nous montrons un peu plus, de sorte que, par analogie avec les flots sous-modulaires ([129], Chapitre 60), on pourrait qualifier les objets combinatoires définis dans le Théorème 4.7 de **couplages sous-modulaires (bipartis)**⁶. La Section 1.3.3.5 contient des prérequis pour la preuve du Théorème 4.7.

Théorème 4.7 Soit $G = ((A, B), E)$ un multigraphe biparti et pour tout $v \in V := A \cup B$, soit f_v une fonction sous-modulaire sur l'ensemble $\delta(v)$ des arêtes incidentes à v . Alors le système suivant est TDAU (et donc box-TDI) :

$$(4.11) \quad x(U) \leq f_v(U) \text{ pour tout } v \in V \text{ et tout } U \subseteq \delta(v)$$

⁵Une construction équivalente a été donnée par Edmonds pour associer une fonction sous-modulaire à toute fonction d'ensemble [129] page 784.

⁶Le cas non-biparti, (auss appelé "matchoid", cf. [129] page 765) donne lieu à un problème NP-dur.

Notons que le Théorème 4.7 contient le Théorème 1.31 (et donc le théorème d'intersection des polymatroïdes) qui correspond au cas où $A = \{a\}$ et $B = \{b\}$ (et E ne contient que des arêtes parallèles entre a et b). Nous donnons une idée de preuve pour le Théorème 4.7 en le réduisant au Théorème 1.31. Nous donnons ensuite une preuve plus standard.

Idée de preuve Soit f_A la fonction d'ensemble sur E définie pour $F \subseteq E$ par :

$$(4.12) \quad f_A(F) := \sum_{a \in A} f_a(\delta(a) \cap F)$$

Définissons f_B de manière analogue. Notons que f_A et f_B sont des fonctions sous-modulaires (comme somme de fonctions sous-modulaires). D'après le Théorème 1.31 page 53, le système suivant est donc box-TDI :

$$(4.13) \begin{aligned} \text{(i)} \quad & x(F) \leq f_A(F) \text{ pour tout } F \in \mathcal{P}(E) \setminus \emptyset \\ \text{(ii)} \quad & x(F) \leq f_B(F) \text{ pour tout } F \in \mathcal{P}(E) \setminus \emptyset \end{aligned}$$

Comme les inégalités (4.13) s'écrivent comme sommes à coefficients entiers des inégalités de (4.11), ce dernier système est lui aussi box-TDI (et les deux systèmes définissent le même polyèdre). \square

Nous donnons un schéma de démonstration plus détaillé pour le Théorème 4.7. **Preuve** Soit $w : E \rightarrow \mathbb{R}$ et y une solution duale optimale de la maximisation de $\mathbf{1}^T x$ sur (4.11). Décroisons y en y^* sur chaque $\delta(v)$, indépendamment pour chaque $v \in A \cup B$. D'après [129], Théorème 48.1, on peut choisir y^* tel que les sous-ensembles de $\delta(v)$ correspondants aux composantes non-nulles de la restriction de y^* à $\delta(v)$ forme une famille laminaire (de sous-ensembles d'arêtes de $\delta(v)$). L'union sur $v \in A$ de ces variables actives forme encore une famille laminaire (et il en va de même pour l'union sur $v \in B$). Les variables actives de y^* peuvent donc être écrite comme l'union de deux familles laminaires, dont la matrice d'incidence est TTU d'après [129], Théorème 41.11. Le système (4.11) est donc TDAU (et donc box-TDI d'après [129], Théorème 5.35). \square

4.2.2.3 Obstructions pour les graphes sous-modulairement parfaits

Dans cette section nous étudions les graphes “qui se comportent bien” vis-à-vis des fonctions sous-modulaires. Des exemples de tels graphes sont les bipartis, leurs line-graphes ainsi que la fermeture par sommet d'articulation de ces deux classes. Nous donnons une liste d'obstructions à la “perfection sous-modulaire”. Les résultats sont encore trop épars pour motiver une définition formelle de cette notion. Nous regroupons plutôt sous ce nom les différentes notions que l'on peut définir par l'intégralité du système (4.3). Un des espoirs est que ces différentes notions

n'en donnent qu'une seule. Nous présentons quelques graphes pour lesquels le système (4.3) ne donne pas une formule min-max pour certains coûts sous-modulaires (Figures 4.2, 4.3 et 4.4).

Remarque 4.8 *La classe des graphes pour lesquels le système (4.3) est TDAU pour tout f sous-modulaire est fermée par sommet d'articulation.* \square

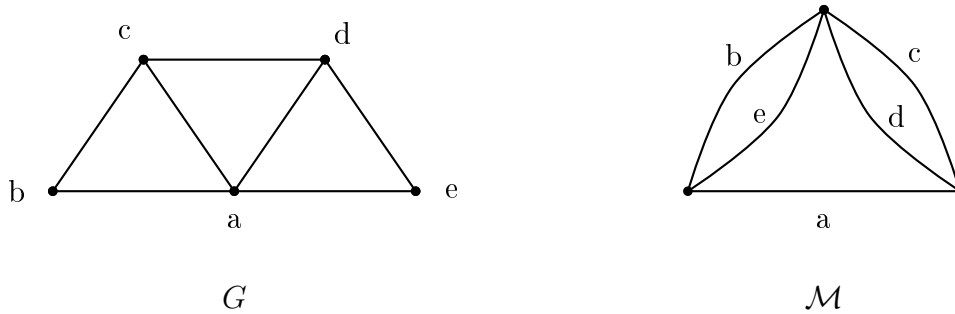


Figure 4.2 – Graphe G et matroïde graphique \mathcal{M} sur un même ensemble à 5 éléments. Une couverture fractionnaire optimale est donnée par $1/2 \times (\{b\} + \{a, b, c\} + \{c, d\} + \{a, d, e\} + \{e\})$ de coût $7/2$ ($\{a, b, c\}$ et $\{a, d, e\}$ ont rang 2 dans \mathcal{M}). Un empilement fractionnaire optimal est donné par $x_b = x_e = 1$ et $x_a = x_c = x_d = 1/2$

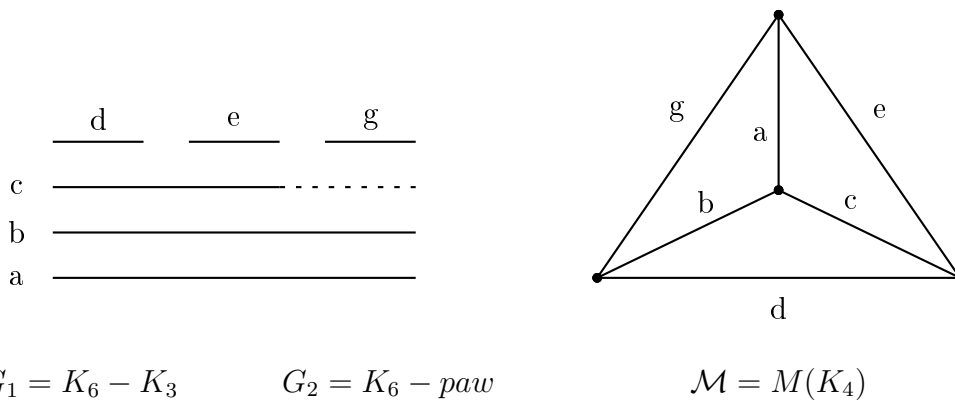
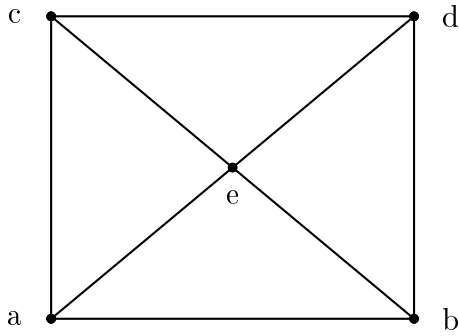


Figure 4.3 – Représentation par intervalles de deux graphes trivialement parfaits G_1 et G_2 (qui diffèrent seulement par le fait que c est adjacent ou non à g) et un matroïde $\mathcal{M} = M(K_4)$ sur un même ensemble à 6 éléments. Dans G_1 comme dans G_2 , $\max \mathbf{1}^T x$ sujet à (4.5) est atteint par $x_a = x_b = x_c = 1/2$ et $x_d = x_e = x_g = 1$. Une couverture fractionnaire optimale est donnée par $y_C = 1/2$ pour $C \in \{\{a, b, g\}, \{a, c, e\}, \{b, c, d\}, \{d\}, \{e\}, \{g\}\}$ et $y_C = 0$ sinon.



G

Figure 4.4 – Graphe G dont la partition fractionnaire $y_C = 1/2$ pour $C \in \{\{a, b, g\}, \{a, c, e\}, \{b, c, d\}, \{d\}, \{e\}, \{g\}\}$ et $y_C = 0$ sinon est optimale pour la fonction f de (4.14). Le coût de la partition est $17/2$. Un empilement fractionnaire optimal est donné par $x_a = x_b = 7/4$ et $x_c = x_d = 5/4$ et $x_e = 5/2$

$$(4.14) \quad f(K) := \begin{cases} 2 & \text{si } K \in \{\{a\}, \{b\}, \{c\}, \{d\}\} \\ 3 & \text{si } K \in \{\{e\}, \{a, c\}, \{b, d\}\} \\ 4 & \text{si } K \in \{\{a, b\}, \{c, d\}\} \\ 5 & \text{si } K \in \{\{a, e\}, \{b, e\}, \{c, e\}, \{d, e\}, \{c, d, e\}\} \\ 6 & \text{si } K \in \{\{a, b, e\}, \{a, c, e\}, \{b, d, e\}\} \end{cases}$$

4.2.2.4 Obstructions à la max-perfection

Dans cette Section, nous présentons trois graphes pour lesquels le système (4.4) ne donne pas une formule min-max pour le cas des fonctions de coût max-batch (Figures 4.5, 4.6 et 4.7).

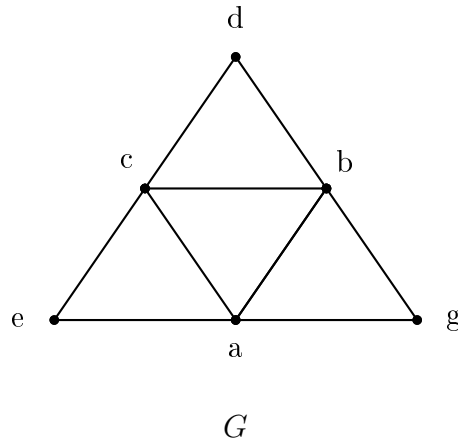


Figure 4.5 – Soit $p(a) = p(b) = p(c) = 2$, $p(d) = p(e) = p(g) = 1$ et f la fonction max-batch définie par p . Une solution optimale pour la relaxation linéaire de (4.5) est donnée par $y_C = 1/2$ si $C \in \{\{d\}, \{e\}, \{g\}, \{a, b, g\}, \{a, c, e\}, \{b, c, d\}\}$ et $y_C = 0$ sinon. Le coût de la partition est $9/2$. L'optimalité peut être prouvée à l'aide d'un x qui maximise $\mathbf{1}^T x$ sous les contraintes (4.5), par exemple, $x_a = x_b = x_c := 1/2$ et $x_d = x_e = x_g := 1$.

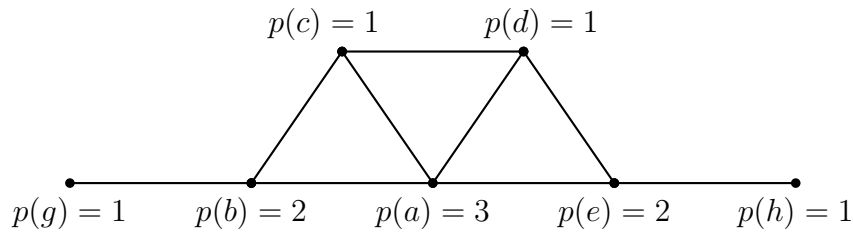


Figure 4.6 – Soit f la fonction max-batch définie par p . Une solution optimale pour la relaxation linéaire de (4.5) est donnée par $y_C = 1/2$ si $C \in \{\{g\}, \{b, g\}, \{a, b, c\}, \{a, d, e\}, \{c, d\}, \{e, h\}, \{h\}\}$ et $y_C = 0$ sinon. Le coût de la partition est $13/2$. L'optimalité peut être prouvée à l'aide d'un x qui maximise $\mathbf{1}^T x$ sous les contraintes (4.5), par exemple, $x_a = 3/2$, $x_c = x_d := 1/2$ et $x_b = x_e = x_g = x_h := 1$.

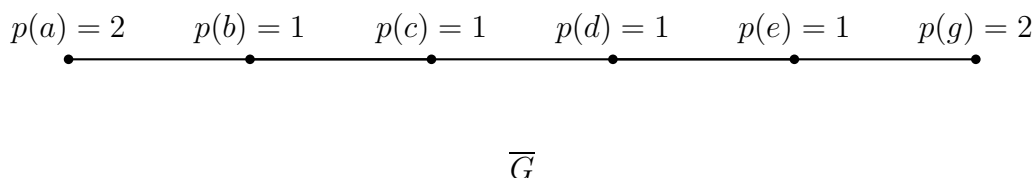


Figure 4.7 – Soit f la fonction max-batch définie par p . Une solution optimale pour la relaxation linéaire de (4.5) est donnée par $y_C = 1/2$ si $C \in \{\{a, c, g\}, \{a, d, g\}, \{b, e\}, \{c, e\}, \{b, d\}\}$ et $y_C = 0$ sinon. Le coût de la partition est $7/2$. L’optimalité peut être prouvée à l’aide d’un x qui maximise $\mathbf{1}^T x$ sous les contraintes (4.5), par exemple, $x_a = 1$, $x_b = x_c = x_d = x_e = x_g = 1/2$.

4.3 Approches “classiques” pour [PCliqBatchGen]

Dans cette section nous attaquons [PCliqBatchGen] par un autre angle que la borne (4.4). Nous commençons par donner quelques limites imposées par la complexité algorithmique (Section 4.3.1). Le résultat principal de la Section 4.3 est un algorithme de programmation dynamique qui permet de résoudre [PCliqBatchGen] sur les graphes d’intervalles quand on se restreint aux coûts valeur-polymatroïdaux (Section 4.3.2).

4.3.1 Complexité de [PCliqBatchGen] pour des coûts valeur-polymatroïdaux

Dans cette section, nous mentionnons quelques résultats de complexité pour [PCliqBatchGen] restreint à quelques classes de fonctions valeurs polymatroïdaux et quelques classes de graphes. Les différents résultats de NP-complétude discréditent d’éventuelles tentatives pour étendre les résultats de polynomialité discutés dans la Section 4.3.2.

Mis-à-part les coûts définis par un TSP enraciné sur un réseau en arbre, les classes de fonctions considérées sont toutes “valeur-polymatroïdales” et définies dans la Section 2.4.1.

4.3.1.1 Maximum

La restriction de [PCliqBatchGen] aux fonctions max-batch, a fait l'objet de plusieurs articles. Le problème correspondant de coloration sur le graphe complémentaire est aussi appelé "max-coloring". Nous y référons par [MaxColor]. Un traitement plus détaillé et d'autres références sont présentées dans [70] (en particulier pour l'approximabilité). [MaxColor] est fortement NP-complet sur les split graphes [65, 27], sur les graphes bipartis [65] et sur les graphes d'intervalles [71]. De plus, [MaxColor] est polynomial sur les graphes sans- P_4 [65] et les compléments de graphes d'intervalles [76, 68, 14]. Notons finalement que [MaxColor] admet un PTAS sur les split graphes et les arbres (pour lesquels l'appartenance à P ou à NPC est ouverte) [70].

4.3.1.2 Fonction cardinale tronquée et q -coloration partielle

Soit $q \in \mathbb{N}$. Considérons [PCliqBatchGen] restreint aux coûts définis pour chaque clique K par $f(K) := \min\{|K|, q\}$. Un coût f de ce type est *cardinal tronqué*. [PCliqBatchGen] restreint aux coût cardinaux tronqués intervient dans la relation de Greene-Kleitman qui donne la cardinalité maximum de l'union de q stables pour certains graphes (voir le problème de la coloration partielle [ColPart], Section 5.2). Bien que [ColPart] soit NP-complet dans les split graphes [138],

Remarque 4.9 *Le problème [PCliqBatchGen] est polynomial si on se restreint aux coûts cardinaux tronqués et aux split graphes.*

Idée de preuve Si $q = 1$, le problème est équivalent à trouver $\bar{\chi}(G)$. Sinon, toute partition des sommets dont une classe est composée d'une clique de cardinalité maximum et chacun des autres sommets est dans une classe particulière est optimale. \square

La simplicité de cette remarque nous pousse à demander pour quelles classes de graphes [PCliqBatchGen] restreint aux coûts cardinaux tronqués est polynomial. Par ailleurs, n'y aurait-il pas une formule min-max simple pour ce problème (autre que la formule de Greene-Kleitman) ?

4.3.1.3 Fonction taille-définie concave

Comme nous l'avons vu, ce type de coût s'exprime comme somme de fonctions cardinales tronquées. Pour ce type de coûts cependant, [PCliqBatchGen] est NP-dur dans les split graphes [84].

4.3.1.4 Probabilités indépendantes

[PCliqBatchGen] restreint aux coûts probabilistes est fortement NP-complet dans les split graphes [64]. Plus de références sont proposées dans [70].

4.3.1.5 Entropie Chromatique

Soit $p : V \rightarrow [0, 1]$ et pour $U \subseteq V$, posons

$$(4.15) \quad c_U := \sum_{u \in U} p(u)$$

$$(4.16) \quad f'(U) := -c_U \log(c_U).$$

Si $c_V = 1$, f' est une fonction *d'entropie chromatique*. Bien que f' ne soit pas valeur-polymatroïdale (elle n'est pas non-décroissante), la fonction $f := f' + c$ est valeur-polymatroïdale, ce qui découle de la concavité de la fonction $x \mapsto x - x \log(x)$ [4]. Puisque pour toute partition en cliques $V = K_1 \cup \dots \cup K_k$, on a $\sum_i f(K_i) = \sum_i f'(K_i) + c(V)$, les deux fonctions f' et f donnent les mêmes partitions optimales.

[PCliqBatchGen] restreint à ce type de coût est étudié dans [4, 41] et est fortement NP-complet sur les compléments de graphes d'intervalles [41].

4.3.2 Programmation dynamique sur les graphes d'intervalles

Dans cette Section, nous montrons que [PCliqBatchGen] restreint aux coûts valeur-polymatroïdaux et aux graphes d'intervalles est Polynomial. En fait, toujours pour les coûts valeur-polymatroïdaux, nous avons montré que cet algorithme peut être étendu pour trouver une partition en Helly-cliques optimum pour les familles d'arcs circulaires⁷ [84].

L'intérêt de la propriété valeur-polymatroïdale⁸ est :

⁷Ce qui pourrait avoir des applications en ordonnancement cyclique.

⁸Suite à une discussion avec Maurice Queyranne, une définition un peu plus générale a émergé : une fonction est *totale sous-modulaire* s'il existe un ordre total \succ sur V tel que pour tous $S \prec T$ et tout $u \in V \setminus (T \cup S)$, on a $f(S+u) + f(T) \leq f(S) + f(T+u)$. La totale sous-modularité peut prendre la place de la valeur-polymatroïdalité dans les Propositions 2.40 et 4.10. Par ailleurs, les coûts définis par l'entropie chromatique (Section 4.3.1.5) sont totalement sous-modulaires. La totale sous-modularité met peut-être encore mieux le doigt sur ce qui joue un rôle dans la preuve des Propositions 2.40 et 4.10. La définition de la totale sous-modularité est plus générale, mais elle me semble intuitivement moins claire dans le contexte de l'économie d'échelle, c'est pourquoi j'ai mené la discussion autour des fonctions valeur-polymatroïdales.

Proposition 4.10 *Pour tout graphe G et toute fonction valeur-polymatroïdale f sur $V(G)$, il existe une partition \mathcal{P} de coût $\bar{\chi}(G, f)$ dans laquelle un des batchs de \mathcal{P} est une clique maximale de G .*

Preuve Voir Proposition 2.40. □

Dans [76], [PCliqBatchGen] est résolu sur les compléments de graphes d'intervalles en temps $\mathcal{O}(n^3)$ pour les fonctions de coûts de type maximum. L'algorithme est basé sur l'observation que (pour tout graphe et tout coût $p : V \rightarrow \mathbb{R}_+$) il existe une partition en cliques optimale dans laquelle un des batchs contenant un sommet de coût maximum est une clique maximale par inclusion. Cela n'est plus vrai pour les coûts valeur-polymatroïdaux comme le montre l'exemple de la Figure 4.8. Cependant, on peut donner une légère généralisation de l'algorithme proposé dans [76], qui donne une solution optimale pour tous les coûts valeur-polymatroïdaux.

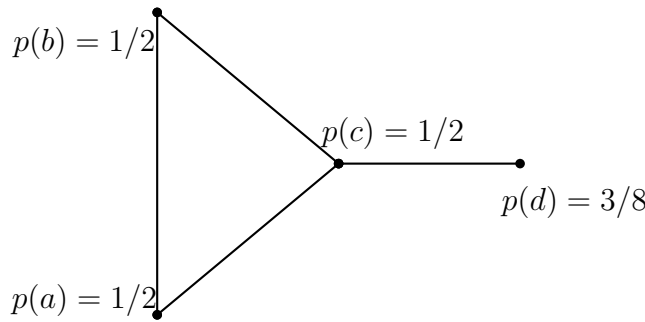


Figure 4.8 – Soit f le coût probabiliste défini par p . Le sommet d a un coût maximum $f(\{d\}) = 1 - p(d) = 5/8$. Cependant, il n'y a pas de partition en cliques optimale dans laquelle d est dans une clique maximale puisque $25/16 = f(\{a, b\}) + f(\{c, d\}) > \bar{\chi}(G, f) = f(\{a, b, c\}) + f(\{d\}) = 12/8$.

Théorème 4.11 *Pour tout graphe d'intervalles $G = (V, E)$ et toute fonction valeur-polymatroïdale f sur V donnée par un oracle de valeur, on peut trouver une partition en cliques de G de coût $\bar{\chi}(G, f)$ en temps $\mathcal{O}(n^3)$.*

Preuve Soit $\{I_i = [a_i, b_i]\}_{i=1, \dots, n}$ un ensemble d'intervalles représentant le graphe G . Considérons l'ensemble X des **extrémités** des intervalles.

$$= \{a_i\}_{i=1, \dots, n} \cup \{b_i\}_{i=1, \dots, n} = \{1, \dots, q\}$$

Le **sous-problème** $\mathcal{I}(i, j)$ consiste en l'ensemble des intervalles complètement contenus dans l'intervalle fermé $[i, j]$. Pour toute paire $i \leq j \in X$, soit $F(i, j) := \bar{\chi}(G[\mathcal{I}(i, j)], f)$, le coût minimum d'une partition en cliques du sous-graphe induit

par $\mathcal{I}(i, j)$ (par définition, $F(i, j) = 0$ si $\mathcal{I}(i, j) = \emptyset$). L’approche par programmation dynamique est basée sur le Lemme 4.12 qui implique que l’on peut séparer le problème restreint à $\mathcal{I}(i, j)$ en deux sous-problèmes.

Lemme 4.12 *Pour tout $i, j \in X$, il y a une partition en cliques optimale de $G[\mathcal{I}(i, j)]$ dans laquelle au moins un des batchs induit une clique maximale de $G[\mathcal{I}(i, j)]$.*

Preuve Directement à partir de la Proposition 4.10 □

Etant donné $i < z < j \in X$, soit $K_{i,j}^z$ l’ensemble des intervalles de $\mathcal{I}[i, j]$ contenant le point z . Notons que $K_{i,j}^z$ est une clique pour tout $i \leq z \leq j \in X$.

Lemme 4.13 *Pour toute paire $i < j$ dans X , on a la relation de récurrence suivante :*

$$(4.17) \quad F(i, j) = \min_{z \in [i, j]} \{f(K_{i,j}^z) + (F(i, z - 1) + F(z + 1, j))\}$$

Preuve D’après le Lemme 4.12, il y a une partition en cliques optimale de $G[\mathcal{I}(i, j)]$ dans laquelle un des batchs est une clique maximale B^* . Toutes les cliques maximales de $G[\mathcal{I}(i, j)]$ sont considérées lorsque l’on considère le minimum dans (4.17). Donc $B^* = K_{i,j}^{z^*}$ pour un certain z^* . Etant donné un tel point z^* , tout interval dans $\mathcal{I}[i, z^* - 1]$ a son extrémité terminale avant l’extrémité initiale de tout intervalle dans $\mathcal{I}[z^* + 1, j]$. Par conséquent, le graphe $G(\mathcal{I}[i, j] \setminus B^*)$ se décompose en deux sous-graphes déconnectés : $G(\mathcal{I}[i, z^* - 1])$ et $G(\mathcal{I}[z^* + 1, j])$. On peut donc résoudre les deux problèmes associés à ces sous-graphes indépendamment. □

L’algorithme de programmation dynamique part des conditions initiales :

$$F(i, i) = f(\mathcal{I}[i, i]) \quad \text{pour tout } i = 1, \dots, q$$

En appliquant la relation de récurrence (4.17) en augmentant la taille $j - i$ des sous-problèmes, il calcule une partition optimale

$$S(x_i, x_j) = \begin{cases} \emptyset & \text{si } \mathcal{I}[i, j] = \emptyset; \\ S(i, z^* - 1) \cup B^* \cup S(z^* + 1, j) & \text{sinon.} \end{cases}$$

La valeur optimale est $\bar{\chi}(G, f) = F(1, q)$ et $S(1, q)$ est une solution optimale. Puisqu’il y a $O(q^2) = O(n^2)$ sous-problème et $O(q) = O(n)$ valeurs candidates pour z dans chaque sous-problème, l’algorithme de programmation dynamique résout le problème en temps $\mathcal{O}(n^3)$. Cela termine la preuve du Théorème 4.11. □

4.3.3 TSP-enraciné sur un réseau en arbre

Les coûts décrits par le TSP sur un graphe enraciné sont définis dans la Section 1.4.2.4. D'après le Théorème 2.28, si le graphe est un arbre, une telle fonction de coût est polymatroïdale (cela est expliqué plus simplement dans la Figure 4.9). L'Exemple 2.3 page 120 illustre toutefois que ce type de coût n'est en général pas valeur-polymatroïdal. En l'occurrence, [PCliqBatchGen] restreint aux graphes d'intervalles pour la compatibilité et aux coûts donnés par un TSP enraciné sur une étoile est NP-complet (bien que 2-approximable) [14].

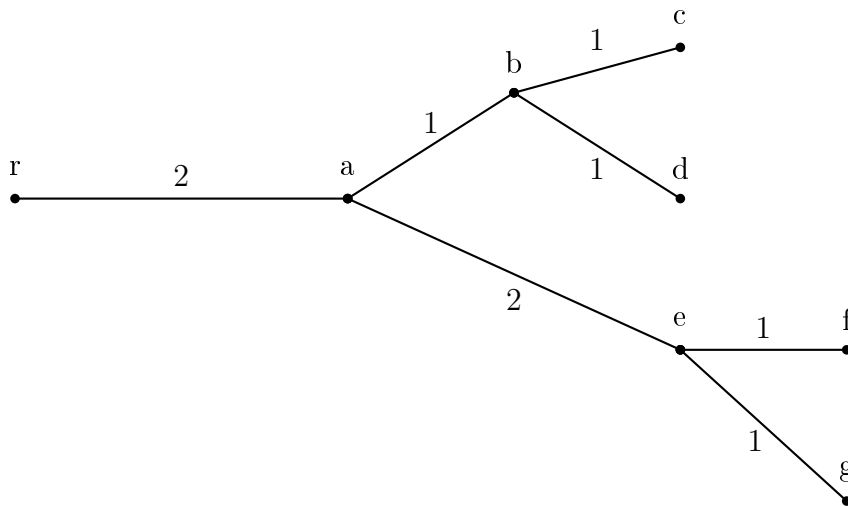


Figure 4.9 – Un arbre enraciné pondéré par $l : E \rightarrow \mathbb{R}$. La fonction de coût $GTSP$ d'un plus court tour de Steiner couvrant $U + r$ est égal 2 fois le coût de l'arbre couvrant $U + r$. Par exemple, pour $U = \{a, b, f\}$, le coût est de 12 et pour $U = \{c, d, e, f\}$, le coût est de 16.

[PCliqBatchGen] restreint aux graphes d'intervalles et coûts donnés par un TSP enraciné sur un arbre a des applications dans certains problèmes de tournées de véhicules avec fenêtre de temps (où la longueur $l(e)$ représente le coût du trajet sur l'arête e et nous supposons que les temps de transport sont négligeables comparés à la taille des fenêtres [68]).

Une éventuelle propriété d'échange intéressante de ce type de coûts, plus forte que la sous-modularité reste envisageable, ce que nous faisons transparaître au travers de la question suivante. Bien que les graphes sans- P_4 ne soient pas sous-modulairement parfaits⁹,

Question 4.14 *Le système (4.5) est-il TDI pour tout graphe G sans- P_4 et toute fonction de coût f définie par un TSP enraciné sur un réseau en arbre ? [PCliqBatchGen] est-il polynomial pour ce type d'instances ?*

⁹Voir Exemple 4.4, ou encore, $\overline{3K_2}$ avec un coût donné par le rang de $\mathcal{M}(K_4)$.

4.4 Pour aller plus loin

4.4.1 Une approche alternative pour [MaxColor]

Les papiers qui étudient [MaxColor] se concentrent presque exclusivement sur les classes de graphes. Certains résultats s'intéressent aussi au nombre de valeurs différentes que prend la fonction $p : V \rightarrow \mathbb{N}$. Nous illustrons que l'interaction entre la structure du graphe et la structure des coûts donne une prise algorithmique alternative dans certains cas. Cette conclusion est obtenue par le biais d'une caractérisation des paires (G, p) pour lesquelles un stable de poids maximum donne une formule min-max pour le coût minimum d'une partition en cliques.

Dans le problème [PCliqMax], si tous les poids sont égaux, on retombe naturellement sur la théorie des graphes parfaits :

Proposition 4.15 *Si (G, p) est un graphe pondéré et $p(v) = p(w) > 0$ pour tous $v, w \in V(G)$ alors $\bar{\chi}(G[U], p) = \alpha(G[U], p)$ pour tout $U \subseteq V$ si et seulement si G est parfait.* \square

D'un autre côté l'inégalité de stable de poids maximum ne donne en général une formule min-max que pour les graphes sans- P_3 .

Proposition 4.16 *Soit G un graphe. On a $\bar{\chi}(G, p) = \alpha(G, p)$ pour tout $p : V \rightarrow \mathbb{N}$ si et seulement si G est sans- P_3 .*

Preuve Si G contient un P_3 , on choisit p comme dans la Figure 4.1 page 189 et $p = 0$ pour les sommets en dehors de ce P_3 . On a alors $4 = \bar{\chi}(G, p) > \alpha(G, p) = 3$. La suffisance vient du fait qu'un graphe sans- P_3 est une union disjointe de clique (Théorème 2.22). Si $G = (V, E)$ est l'union disjointe de k cliques : $V = \{C_1, C_2, \dots, C_k\}$, considérons un sommet s_i de poids maximum dans chacune des cliques. Clairement l'ensemble des s_i forme un stable de poids égal au coût de la partition de V donnée par G lui-même : (C_1, C_2, \dots, C_k) . \square

Notons que dans la Figure 4.1, le graphe fait obstruction car le sommet du milieu du P_3 est de coût supérieur aux deux autres. Cet ordre sur les poids est nécessaire pour faire obstruction, comme le montre la proposition suivante :

Proposition 4.17 *Soit (G, p) un graphe pondéré tel qu'il existe un ordre simplicial qui étend l'ordre partiel des p décroissant. Alors $\bar{\chi}(G[U], p) = \alpha(G[U], p)$ pour tout $U \subseteq V$.*

Preuve Par induction sur le nombre de sommets du graphe. C'est trivial si G a un seul sommet. Soit (G, p) un graphe pondéré, v un sommet simplicial de G qui a un poids maximum et $H = G - N^+(v)$. Comme on peut mettre v et tous ces voisins dans une seule clique, on a :

$$(4.18) \quad \bar{\chi}(G, p) \leq \bar{\chi}(H, p) + p(v)$$

Comme v est isolé dans $H + v$, on a :

$$(4.19) \quad \alpha(H, p) + p(v) \leq \alpha(G, p)$$

Par hypothèse d'induction, $\bar{\chi}(H, p) = \alpha(H, p)$ et donc $\bar{\chi}(G, p) \leq \alpha(G, p)$. On a donc $\bar{\chi}(G, p) = \alpha(G, p)$ en utilisant la Proposition 4.2 page 188. \square

En fait, la simplicialité de l'ordre des poids décroissants caractérise la validité de cette formule min-max :

Proposition 4.18 *Soit (G, p) un graphe pondéré tel que les coûts soient tous différents et non nuls. Alors $\bar{\chi}(G[U], p) = \alpha(G[U], p)$ pour tout $U \subseteq V$ si et seulement si l'ordre décroissant des p est un ordre simplicial.*

Preuve Si l'ordre décroissant des p n'est pas simplicial, alors il existe trois sommets x, y, z tels que $xy \in E$, $yz \in E$, $xz \notin E$, $p(y) > p(x) > 0$ et $p(y) > p(z) > 0$. On a alors $\bar{\chi}(G[\{x, y, z\}], p) > \alpha(G[\{x, y, z\}], p)$ (voir Figure 4.1). \square

Nous laissons ouverte la question de caractériser les graphes pondérés pour lesquels $\bar{\chi}(G[U], p) = \alpha(G[U], p)$ pour tout $U \subseteq V$ dans le cas général où les $p(v)$ ne sont ni tous égaux ni tous différents.

On peut aussi espérer aller encore plus loin avec cette approche hybride, en résolvant [PCliqMax] avec des outils plus puissants que la borne inférieure donnée par le poids maximum d'un stable. Par exemple, si l'on considère la borne (4.4) donnée par la théorie des jeux, on pourrait tout aussi bien tirer parti des interactions entre la structure du graphe et l'ordre des coûts¹⁰. Avant de laisser de côté ces pistes prometteuses, notons qu'il serait intéressant de suivre le paradigme de la Section 2.1.1.1, en étudiant les applications pour savoir, la structure des graphes mis en jeu est susceptible d'être liée à celle des coûts. Une réponse positive a déjà émergée dans ce sens pour le problème [Pintro] :

Proposition 4.19 [105] *Dans le cas où les intervalles $[p_v, q_v]$ du graphe de compatibilité décrivent des temps minimaux (p_v) et maximaux (q_v) d'opération, l'algorithme GAC de la Section 3.3.1, **adapté en considérant un ordre décroissant des p_v** donne une solution optimale pour toute contrainte de capacité b .*

Les contraintes de temps maximaux d'opération interviennent par exemple dans les applications de type *Hoist Scheduling Problem* (HSP).

4.4.2 Le polytope des constellations

Dans cette section, nous proposons une approche polyédrale pour le problème [PCliqMax] basée sur les constellations. Cela confirme la pertinence du lien observé dans

¹⁰Les graphes des Figures 4.5, 4.6 et 4.7 nécessitent des coûts particuliers pour faire obstruction au fait que (4.4) donne une formule min-max.

la Section 4.4.1 entre la structure du graphe et la structure des coût. En particulier, nous montrons une réduction de [PCliqMax] au problème du stable de poids maximum dans un graphe auxiliaire.

Soit $G = (V, E)$ un graphe et $p : V \rightarrow \mathbb{R}$. Pour tout $xy \in E$ tels que $p(x) \geq p(y)$, affectons le poids $w(xy) := p(x) - p(y)$ à l'arête xy . Informellement, ce poids correspond à la diminution du coût d'une partition en cliques en plaçant y dans la même clique que x plutôt que tout seul. Plus précisément, si \prec est une extension linéaire de l'ordre non-croissant des p , le coût minimum d'une partition en cliques peut être réinterprété en terme du poids maximum d'une constellation de (G, \prec) . Autrement dit,

Proposition 4.20 *Pour tout graphe $G = (V, E)$, tout $p : V \rightarrow \mathbb{R}_+$ et tout ordre total \prec sur V tel que $x \prec y$ pour tous $x, y \in V(G)$ tels que $p(x) > p(y)$,*

$$(4.20) \quad \sum_{v \in V} p(v) = \bar{\chi}(G, p) + \alpha(D(G, \prec, w))$$

où w est la pondération des arêtes mentionnée ci-dessus.

A notre connaissance, aucune classe de graphes \mathcal{G} n'est connue pour laquelle cette réduction donne un algorithme polynomial pour [PCliqMax]. Cependant, notons que les forêts clique-connectantes ne semblent pas posséder l'expressivité nécessaire pour modéliser d'autres sous-problèmes de [PCliqBatchGen] mentionnés dans les Sections 4.3.1 et 4.3.3. Cela est plutôt une bonne nouvelle si l'on s'intéresse à la pertinence applicative des constellations relativement au problème [PCliqMax].

La NP-complétude de déterminer une constellation de poids maximum dans les split graphes, les compléments d'intervalles et les compléments de bipartis découle des résultats discutés dans la Section 4.3.1.1. Cela implique que le polytope des constellations n'est pas descriptible dans ces classes de graphes (à moins que NP=co-NP [129], Section 5.12). Notons cependant que cette preuve de NP-difficulté de la description du polytope des constellation ne tient que si l'on s'intéresse à l'ensemble des ordres possibles sur les sommets. Suite à la discussion menée dans la Section 4.4.1, il reste envisageable que le polytope des constellations soit descriptible dans les split graphes ou les compléments d'intervalles, si l'on se restreint à des ordres sur les sommets "compatibles avec la structure du graphe". Mais revenons à une problématique dont la solution à court terme semble plus tangible : la description du polytope des constellations des graphes complets est encore loin d'être triviale¹¹.

Finalement, observons que l'approche par le polytope des constellations et l'approche proposée dans la Section 3.4.7 (pour le problème de la coloration b -bornée) peuvent être combinées en une approche commune pour le problème de la max-coloration b -bornée. Ce dernier problème se réduit donc à trouver un indépendant

¹¹Tout du moins pour la poignée de spécialistes qui se sont penchés sur la question, c'est-à-dire, Denis Cornaz, Jean Fonlupt et moi-même [48].

de poids maximum, commun au graphe des arêtes dépendantes et à un matroïde de partition généralisée.

4.4.3 Cardinalité bornée et max-coloration des matroïdes

Dans cette section nous discutons le problème de partition de coût minimum avec batchs restreint aux indépendants d'un matroïde. Cela contient les problématiques associées aux contraintes de cardinalité (qui correspondent aux matroïdes uniformes). Nous donnons une borne inférieure sur le coût minimum d'une partition en indépendants d'un matroïde. Cette borne donne une formule min-max si f est max-batch dans le cas d'un système d'indépendance donné par un matroïde uniforme. Cette borne n'est cependant pas serrée pour les matroïdes généraux. En l'occurrence, la polynomialité pour les matroïdes généraux semble ouverte si f est max-batch. D'un autre côté, le problème de partition de coût minimum en indépendants est NP-dur pour les matroïdes uniformes même si f est sous-modulaire (et même si f est donnée par un TSP enraciné sur un enraciné).

Contrairement aux problèmes traités dans les Chapitres 3 et 5, le Théorème d'union des matroïdes ne semble pas permettre de résoudre le problème de partition de coût minimum en indépendants. Cependant, le Théorème de couverture par des bases (que nous exprimons en terme de partition en indépendants pour des raisons précisées après) permet de développer une borne inférieure intéressante.

Théorème 4.21 (de couverture d'un matroïde par des bases [129]) *Pour tout matroïde \mathcal{M} , le nombre minimum d'indépendants qu'il faut pour partitionner V est égal au minimum de*

$$(4.21) \max_{U \subseteq V} \left\lceil \frac{|U|}{r(U)} \right\rceil$$

L'inégalité $\min \geq \max$ dans le Théorème 4.21 découle d'un argument de comptage élémentaire :

Ingrédient 4.22 *Pour tout hypergraphe héréditaire $H = (V, \mathcal{I})$, le nombre minimum d'indépendants nécessaire pour partitionner V est supérieur ou égal à*

$$\max_{U \subseteq V} \left\lceil \frac{|U|}{r(U)} \right\rceil$$

Preuve La cardinalité de l'union de k arêtes de H incluses dans U est inférieure à $kr(U)$. □

L'ingrédient 4.22 s'exprime aussi par la dualité faible en programmation linéaire en nombre entiers : considérons la paire de programmes linéaires duaux (où $x \in \mathbb{R}^{\mathcal{I}}$ et $y \in \mathbb{R}^{\mathcal{P}(V)}$).

$$\begin{aligned}
 (4.22)(i) \quad & \min \mathbf{1}^T x \\
 (ii) \quad & \sum_{I \ni v} x_I \geq \lceil |U|/r(U) \rceil \text{ pour tout } U \subseteq V \\
 (iii) \quad & x \geq \mathbf{0}
 \end{aligned}$$

$$\begin{aligned}
 (4.23)(i) \quad & \max \sum_{U \subseteq V} \lceil |U|/r(U) \rceil \cdot y_U \\
 (ii) \quad & \sum_{U \mid I \cap U \neq \emptyset} y_U \leq 1 \text{ pour tout } I \in \mathcal{I} \\
 (iii) \quad & y_U \geq \mathbf{0}
 \end{aligned}$$

D'après l'Ingrédient 4.22, une solution entière de (4.22) est une couverture de V par des éléments de \mathcal{I} . On peut généraliser la borne donnée par cette relaxation linéaire en remplaçant la fonction de coût $\mathbf{1}$ par une fonction de coût f^{12} .

Proposition 4.23 *Pour tout hypergraphe héréditaire $H = (V, \mathcal{I})$ et toute fonction non-décroissante $f : \mathcal{I} \rightarrow \mathbb{R}_+$, le coût minimum d'une partition de V en indépendant est supérieur ou égal à :*

$$\begin{aligned}
 (4.24)(i) \quad & \max \sum_{U \subseteq V} \lceil |U|/r(U) \rceil \cdot y_U \\
 (ii) \quad & \sum_{U \mid I \cap U \neq \emptyset} y_U \leq f(I) \text{ pour tout } I \in \mathcal{I} \\
 (iii) \quad & y_U \geq \mathbf{0}
 \end{aligned}$$

La Proposition 4.24 montre que cette approche donne une formule min-max dans le cas des matroïdes uniformes si f est un coût max-batch. Ce problème de partition, équivalent à un problème d'ordonnancement par batchs parallèles avec contraintes de cardinalité :

$$B1 \mid p\text{-batch}, b \leq n \mid C_{\max}$$

peut être résolu par l'algorithme glouton suivant [33].

Algorithme glouton pour max-coloring : *Considérer les éléments de V dans un ordre décroissant des p . Mettre chaque sommet dans la couleur de plus petit indice possible (en préservant le fait que chaque batch est un indépendant).*

¹²De manière similaire à ce que nous avons fait dans le cas des hypergraphes conformes (voir (4.4) page 191).

Algorithme dual glouton \mathcal{U}_n^b : Considérer la chaîne d'ensembles $U_i = \{v_1, v_2, \dots, v_i\}$. Affecter la valeur $y_{U_n} := p(n)$ et $y_{U_i} := p(v_i) - p(v_{i+1})$ pour $i \in [1..n-1]$.

Proposition 4.24 Pour tout matroïde uniforme \mathcal{U}_n^b et tout $p : \{1, 2, \dots, n\} \rightarrow \mathbb{N}$, l'algorithme glouton pour max-coloring donne une partition de coût minimum en indépendants et l'algorithme dual glouton donne une solution de (4.24), de même valeur que la partition.

Preuve Pour tout hypergraphe héréditaire et toute fonction f non-décroissante la solution donnée par l'algorithme glouton est réalisable par construction. La solution donnée par l'algorithme dual glouton est elle aussi toujours une solution réalisable car, pour tout $I \in \mathcal{I}$, $y(I) = \max_{v \in I} p(v) \leq f(I)$. Puisque les deux solutions sont réalisables, le coût de la partition est supérieur ou égal à $\sum_{U \subseteq V} \lceil |U|/r(U) \rceil y_U$.

Nous montrons, que les deux solutions sont égales si $\mathcal{M} = \mathcal{U}_n^b$ et f est max-batch, par induction sur n . Considérons la partition induite en enlevant le sommet (n) de plus petit coût. Effectuons la mise-à-jour duale suivante : $y_{U_n} := 0$ et $y_{U_{n-1}} := y_{U_{n-1}} + p(n)$ et considérons deux cas :

Cas $n \not\equiv 1 \pmod{b}$: Le coût de la partition et la valeur de la solution duale restent inchangées.

Cas $n \equiv 1 \pmod{b}$: Le coût de la partition ainsi que la valeur de la solution duale diminuent de $p(n)$.

□

La borne inférieure donnée par (4.24) n'est cependant pas serrée pour les matroïdes et les coûts max-batch en général comme le montre la Figure 4.10.

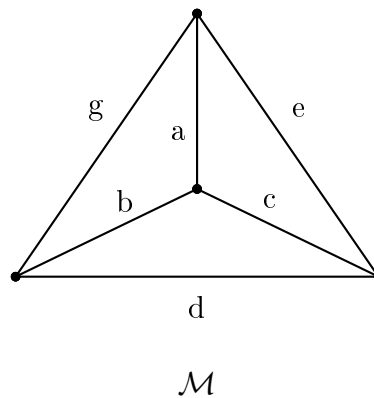


Figure 4.10 – Soit f le coût max-batch défini par $p(a) = p(b) = p(c) := 2$ et $p(d) = p(e) = p(g) := 1$. Le coût minimum d'une partition fractionnaire respectant les contraintes 4.22 ii) est $11/3$, obtenu par la solution $x_I = 1/3$ pour $I \in \{\{a, b, c\}, \{a, c, d\}, \{b, c, g\}, \{a, b, e\}, \{d, e\}, \{d, g\}, \{e, g\}\}$ et $x_I = 0$ sinon. Une solution duale (i.e. optimale pour 4.24) est obtenue avec $y_U = 1/3$ pour $U \in \{\{a\}, \{b\}, \{c\}, \{a, b, g\}, \{b, c, d\}, \{a, c, e\}, \{d, e, g\}\}$ et $y_U = 0$ sinon.

La question de la complexité de [MaxColor] dans les matroïdes reste toutefois ouverte :

Question 4.25 *Quelle est la complexité de trouver une partition en indépendants de coût minimum si l'on se restreint aux coûts max-batch ?*

Si ce problème était polynomial, on pourrait aussi essayer de l'étendre aux coûts valeur-polymatroïdaux. Cependant, pour les coûts sous-modulaires, le problème de partition en indépendants est NP-dur¹³.

Proposition 4.26 *Le problème de partition de coût minimum en indépendants est NP-complet si l'on se restreint aux matroïdes uniformes et aux coûts donnés par le TSP sur un arbre enraciné.*

Preuve On réduit [3-Partition] à ce problème.

[3-Partition]

Données : Un ensemble A ayant $3m$ éléments. Une taille $s : A \rightarrow \mathbb{N}$. Une capacité $B \in \mathbb{N}$.

Résultat : Existe-t-il une partition \mathcal{A} de A en m ensembles: $\mathcal{A} = \{A_1, A_2, \dots, A_m\}$, telle que $|A_i| = 3$ et $\sum_{a \in A_i} s(a) = B$ pour tout i .

Commentaires : fortement NPC, même si $\frac{B}{4} < s(a) < \frac{B}{2}$ pour tout $a \in A$ [80].

Pour une instance de [3-Partition], considérons le matroïde uniforme de rang B sur un ensemble de cardinalité mB . Créons ensuite un arbre enraciné dans une feuille r avec longueurs unitaires sur les arêtes, comme dans la Figure 4.11. Alors [3-Partition] a une solution si et seulement s'il existe une tournée de véhicules enracinée en r couvrant l'ensemble des feuilles de l'arbre de la Figure 4.11, telle que chaque tournée couvre au plus B sommets, et telle que le coût total de la tournée soit $2m(1 + 3 + B)$. \square

La question de l'approximabilité pour ce genre de problèmes élémentaires de tournées de véhicules semble malheureusement peu étudiée.

¹³Voir [68] pour des applications plus détaillées et des références.

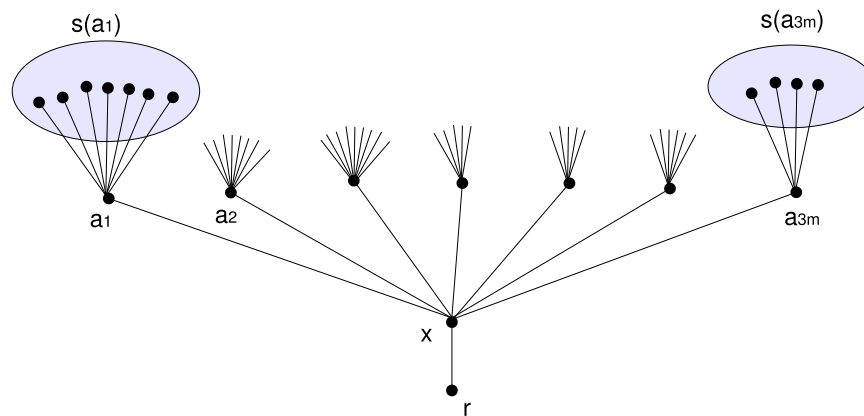


Figure 4.11 – Une réduction de [3-Partition] au problème de partition en indépendants dans un matroïde uniforme avec coûts donnés par le TSP enraciné sur un arbre.

Chapitre 5

Un objectif pour voir les rayons γ

Le problème de la coloration minimum de G se réduit au problème du stable de poids maximum dans un graphe auxiliaire : $G \times K_n$. L'expressivité de ce problème auxiliaire dépasse largement la coloration classique et permet de modéliser d'autres problèmes comme Min Color Sum ou la q -coloration partielle maximum, mais aussi l'ordonnancement avec dates de disponibilités et d'échéances, la coloration par liste et l'extension précolorée. Ces problèmes se réduisent tous à [GOCCP] (*generalized optimum cost chromatic partition*) : problème de coloration dans lequel on a seulement changé la fonction objectif (*dans tout ce chapitre, le temps d'opération d'un batch est unitaire*). Nous décrivons les réductions élémentaires entre ces problèmes.

La contrepartie de l'expressivité de cette réduction au problème du stable maximum dans $G \times K_n$ est sa complexité algorithmique. D'autres réductions moins expressives permettent de traiter plus efficacement certains sous-problèmes de [GOCCP] et de donner une relation min-max pour certaines classes de graphes. Nous discutons trois techniques de ce genre.

La formule de Greene-Kleitman met en relation la cardinalité maximum de l'union de k stables et le coût minimum d'une partition de G en cliques (au sens du Chapitre 4) lorsque la fonction de coût est le rang d'un matroïde uniforme.

L'ordonnancement chromatique avec dates d'échéances doit satisfaire une contrainte élémentaire par clique de G . La suffisance de ces contraintes caractérise les graphes sans- P_4 .

Étant donné une coloration partielle \mathcal{Q} d'un graphe G , l'extension précolorée consiste à trouver une coloration minimum de G qui étend la précoloration \mathcal{Q} . Cela se réduit à la coloration ordinaire dans un graphe auxiliaire G/\mathcal{Q} construit à partir de G en contractant chaque classe de précouleur en un sommet. Le résultat principal de ce chapitre est que le graphe G/\mathcal{Q} est parfait pour toute affectation préalable \mathcal{Q} de couleurs si et seulement si G est le complément d'un graphe de Meyniel.

Mots clés : préaffectation, extension précolorée, graphe de Meyniel, ordonnancement avec dates déchéances, graphe sans- P_4 , q -coloration partielle.

Autres objets utilisés : graphe parfait, graphe d'Artemis, min color sum, optimum cost chromatic partition (OCCP), liste-coloration, stable maximum.

Menu du jour

5.1	Objectifs de l'ordonnancement et coloration par liste .	216
5.1.1	GOCCP et produit de G avec K_n	217
5.1.2	Sous-problèmes et réductions	220
5.1.3	Quelques résultats de complexité	223
5.2	Formule de Greene-Kleitman et box-perfection	224
5.3	Dates déchéances et graphes sans-P_4	227
5.4	Extension précolorée sur les co-Meyniel	229
5.4.1	Philosophie : les contraintes de préaffectations	230
5.4.2	Contraction, PrExt-perfection et graphes de co-Meyniel	230
5.5	Pour aller plus loin	238
5.5.1	Approximabilité de [PrExt] et de [ListCol]	239
5.5.2	Extension de précoloration dans les hypergraphes	240

5.1 Objectifs de l'ordonnancement et coloration par liste

Dans cette section, nous présentons les problèmes d'ordonnancement chromatique que l'on peut formuler à partir du problème de coloration classique simplement en changeant la fonction objectif. Nous présentons le problème [GOCCP] et montrons l'équivalence avec le problème du stable de poids maximum dans $G \times K_n$ (Section 5.1.1). Les problèmes que nous étudions dans ce chapitre sont des sous-problèmes de [GOCCP]. Concevoir ces problèmes comme des restrictions de [GOCCP] permet d'exhiber facilement certaines réductions polynomiales (Section 5.1.2). Ces réductions permettent de s'orienter dans les nombreux résultats de complexité connus pour ces problèmes. Nous survolons ces résultats de complexité pour quelques classes de graphes parfaits (Section 5.1.3).

5.1.1 GOCCP et produit de G avec K_n

Dans cette section, nous décrivons une réduction du problème de la coloration minimum au problème du stable *de cardinalité* maximum dans un graphe auxiliaire : $G \times K_n$. Nous présentons ensuite le problème [GOCCP] et montrons qu'il est équivalent au problème du stable *de poids* maximum dans $G \times K_n$.

Soit $G_1 = (V_1, E_1)$ et $G_2 = (V_2, E_2)$ deux graphes simples. Le **produit cartésien** de G_1 par G_2 (illustré dans la Figure 5.1), est le graphe $G_1 \times G_2 = (V_1 \times V_2, E')$ tel que $((v_1, v_2), (w_1, w_2)) \in E'$ si et seulement si

$$(5.1) \quad (v_1 = w_1 \text{ et } (v_2, w_2) \in E_2) \text{ ou } ((v_1, w_1) \in E_1 \text{ et } v_2 = w_2)$$

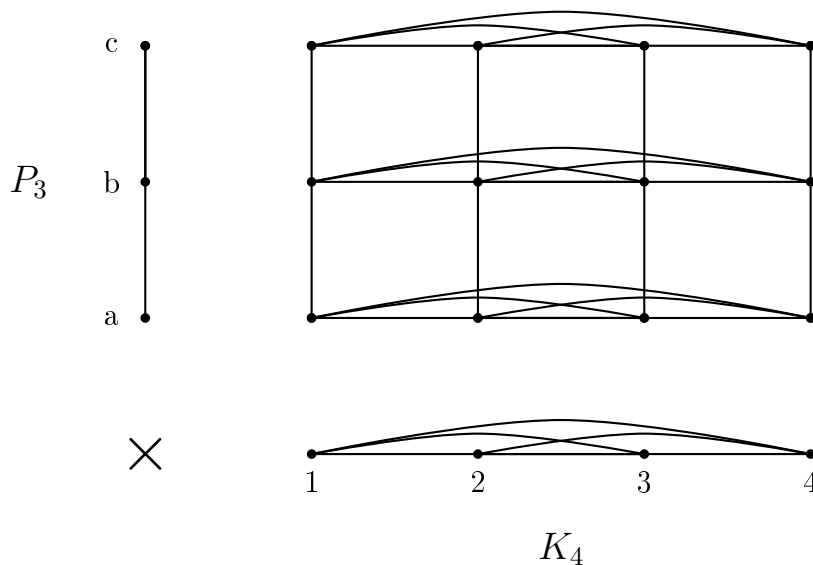


Figure 5.1 – Produit de P_3 avec K_4 .

La Proposition 5.1 met les colorations de G en correspondance avec les stables de $G \times K_n$. On parle ici de coloration au sens strict du terme, c'est-à-dire, une numérotation des sommets (et non une partition en stables, qui est une notion moins fine à cause des permutations sur les couleurs).

Proposition 5.1 [17, 18] *Soit $G = (V, E)$ un graphe et $k \in \mathbb{N}$. L'ensemble des k -colorations partielles de G est en bijection avec les stables de $G \times K_k$. En particulier, $\chi(G) = \min\{k \mid \alpha(G \times K_k) = |V(G)|\}$. De plus, la cardinalité maximum $\alpha_q(G)$ d'un sous-graphe q -colorable de G est égal à $\alpha(G \times K_q)$.*

Idée de preuve $S \subseteq V(G) \times V(K_k)$ est un stable de $G \times K_k$ si et seulement si

- i) S ne contient pas plus d'un sommet par ligne,
- ii) La restriction de S à chaque colonne est un stable.

La première condition équivaut au fait que chaque sommet est colorié au plus une fois. La deuxième condition traduit le fait que chaque colonne (qui correspond à une classe de couleur) induit un stable de G . \square

Nous observons maintenant que la Proposition 5.1 permet une grande expressivité pour modéliser les problèmes d'ordonnement chromatique, en pondérant les sommets de $G \times K_n$ et en cherchant un stable de poids maximum. La Proposition 5.2 montre de plus que cette réduction de la coloration au problème du stable maximum peut servir de définition au problème [GOCCP] ci-dessous.

[GOCCP] Generalized Optimum Cost Chromatic Partition

Données : Un graphe G , un entier $k \geq \chi(G)$ et une fonction de coût $f : V \times [1..k] \rightarrow \mathbb{N}$

Résultat : Une k -coloration $\{S_1, S_2, \dots, S_k\}$ de G .

Mesure : $\sum_{v \in V} f_v(C_v)$, à minimiser.

Dans [GOCCP], $f_v(t)$ est le coût de placer le sommet v dans la t -ème couleur et C_v est la notation en ordonnancement pour désigner la couleur du sommet v (voir Figure 5.2).

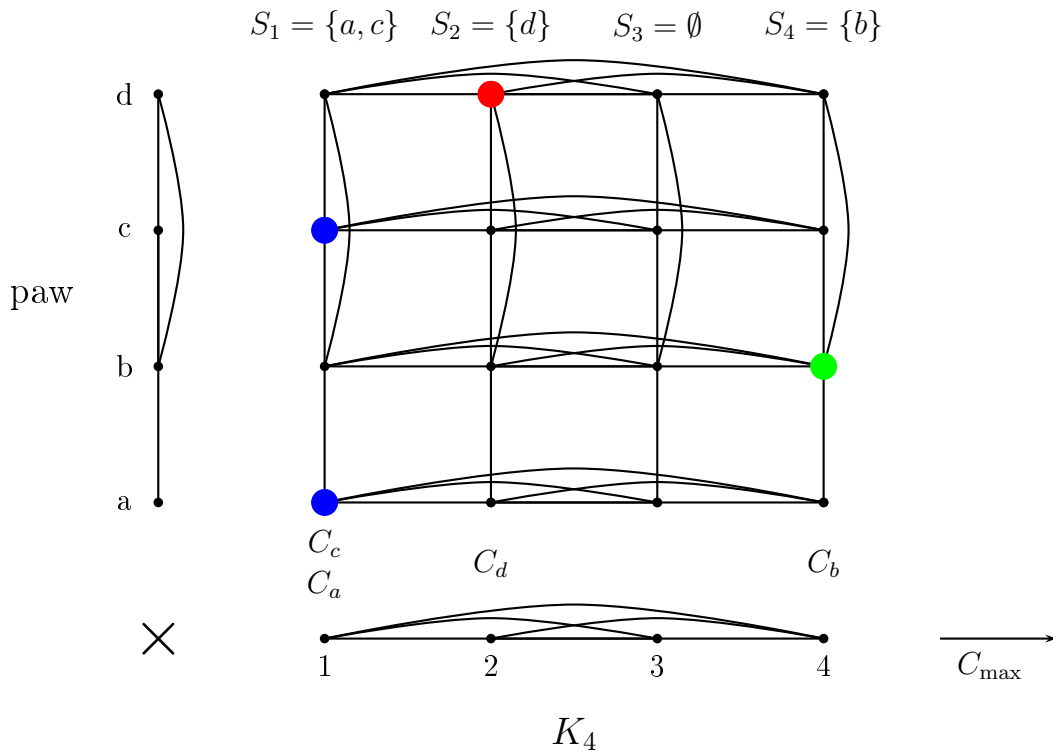


Figure 5.2 – Produit d'un paw avec K_4 . Les 4-colorations de G sont en bijection avec les stables de $G \times K_4$.

Nous montrons maintenant l'équivalence entre [GOCCP] et le problème du stable

de poids maximum dans $G \times K_n$.

Proposition 5.2 *Soit \mathcal{G} une classe de graphes. [GOCCP] sur \mathcal{G} est polynomialement équivalent au problème du stable de poids maximum sur la classe des graphes $\{G \times K_t \mid G \in \mathcal{G}, t \geq \chi(G)\}$.*

Preuve Soit G un graphe, $k \geq \chi(G)$ et $f : V \times [1..k] \rightarrow \mathbb{N}$ une instance de [GOCCP]. Soit [GOCCP](G, f) le coût minimum d'une k -coloration de G , B un majorant des valeurs de f et $\alpha(G, B - f)$ le poids maximum d'un stable dans le graphe $G \times K_t$ pondéré avec les valeurs de $B - f$. On a

$$B \times |V| = [\text{GOCCP}](G, f) + \alpha(G \times K_t, B - f)$$

Réciproquement, une pondération f_2 des sommets de $G \times K_k$ définit une instance de [GOCCP] en considérant le coût $B - f_2$ pour [GOCCP] où B est un majorant des valeurs de f_2 . \square

L'intérêt d'écrire les colorations (partielles) de G comme les stables de $G \times K_q$ apparaît lorsque l'on sait résoudre le problème du stable (de poids) maximum pour ces graphes dans $G \times K_q$. La perfection de $G \times K_q$ pour $q \geq 3$ caractérise les graphes triangulés sans-diamant (Théorème 2.25) et la perfection de $G \times K_2$ caractérise les graphes de parité (Théorème 2.24). S'il est futile de demander quand $G \times K_n$ est sans- $K_{1,3}$ (ce qui est équivalent au fait que G soit sans- P_3), l'appartenance à d'autres classes de graphes pourrait être intéressante et donner de nouveaux cas de polynomialité de [GOCCP].

Question 5.3 *Quelle est la classe des graphes G tels que $G \times K_q$ est h -parfait (pour $q \geq 2$) ?*

Une motivation derrière cette question est que si G est sans roue impaire (*i.e.* un trou impair avec un sommet universel) alors $G \times K_n$ l'est aussi. Il ne semble donc pas évident de générer des graphes $G \times K_q$ qui ne soient pas h -parfait si G est h -parfait. Le Théorème 5.10 indique cependant que $G \times K_q$ n'est pas h -parfait même si G est un graphe parfait très simple. Notre méconnaissance des obstructions à la h -perfection nous dissuade donc d'étudier directement la Question 5.3. Nous proposons toutefois :

Question 5.4 *Est-ce que $G \times K_q$ est h -parfait pour tout $q \in \mathbb{N}$ si G est un graphe séries-parallèles ?*

La Question 5.4 est motivée par les faits suivants :

- Les graphes séries-parallèles sont t -parfaits [129] (et donc h -parfaits).
- Les graphes séries-parallèles ont tree-width au plus 2 (la complexité de [GOCCP] sur les graphes ayant tree-width au plus k semble ouverte, mais les résultats de [101] indiquent que ce problème pourrait être polynomial).
- $G \times K_q$ est parfait pour tout $q \in \mathbb{N}$ si G est un arbre (Théorème 2.25).

5.1.2 Sous-problèmes et réductions

Dans cette section, nous décrivons les réductions de base entre les problèmes d'ordonnement chromatique pour différentes fonctions objectif et les sous-problèmes de la coloration par liste. Une définition plus précise de ces problèmes est disponible dans l'Annexe A.3. Cette section vise principalement à expliquer la Figure 5.3.

La Figure 5.3 donne le diagramme des réductions polynomiales pour les problèmes d'ordonnement par batch avec temps d'opération unitaires ($B1 \mid p_S = 1, \dots$). En vertu de la Remarque 1.65, ces problèmes sont équivalents aux problèmes d'ordonnement sur un nombre infini de machines parallèles avec temps d'opération unitaires ($P_\infty \mid p_j = 1, \dots$). Si les relations entre les problèmes exprimés dans le langage de l'ordonnement classique et de l'ordonnement chromatique ne sont pas immédiates, la seule raison en est que les objectifs de l'ordonnement classique sont formulés en termes de minimisation d'un coût (makespan, temps total d'opération...), alors que certains objectifs de l'ordonnement chromatique sont habituellement formulés en termes de maximisation d'un gain (stable maximum, coloration partielle maximum...). Cependant, les formulations par gain et par coût sont reliées par une dualité élémentaire qui revient à dire que maximiser f est équivalent à minimiser $-f$. Plus précisément, la somme des optimums des problèmes de minimisation de coût et de maximisation de gain est souvent une expression très simple, qui implique l'équivalence polynomiale des deux problèmes considérés. A titre d'exemple, nous exprimons certaines de ces relations dans la proposition suivante. Pour un graphe G et $q \in \mathbb{N}$, $\alpha_q(G)$ est la cardinalité maximum de l'union de q stables de G . Pour $r, d : V \rightarrow \mathbb{N}$, $\alpha_{(r,d)}(G)$ est la cardinalité maximum d'un sous-graphe admettant une coloration partielle ϕ telle que $r(v) + 1 \leq \phi(v) \leq d(v)$. Pour $w : V \rightarrow \mathbb{N}$, $\alpha_{(r,d)}(G, w)$ est le poids maximum d'une telle coloration (voir la Section 1.5 pour l'explication des autres notations).

Proposition 5.5 *Pour tout (hyper-)graphe G on a les relations suivantes (la Figure 5.3 résume ces formules).*

$$(5.2) \quad |V| = \alpha(G) + B1 \mid col(G), p_S = 1, d_j = 1 \mid \sum U_j$$

$$(5.3) \quad |V| = \alpha_q(G) + B1 \mid col(G), p_S = 1, d_j = q \mid \sum U_j$$

$$(5.4) \quad |V| = \alpha_{(r,d)}(G) + B1 \mid col(G), p_S = 1, r_j \mid \sum U_j$$

$$(5.5) \quad w(V) = \alpha_{(r,d)}(G, w) + B1 \mid col(G), p_S = 1, r_j \mid \sum w_j U_j$$

$$(5.6) \quad B \times |V| = \alpha(G \times K_k, B - f) + B1 \mid col(G), p_S = 1 \mid \sum f_j C_j$$

où B est un majorant des valeurs de la fonction $f : G \times K_k \rightarrow \mathbb{N}$.

Preuve La preuve de ces formules repose sur le fait que la somme de la fonction objectif à maximiser et la fonction objectif à minimiser est une constante (égale à

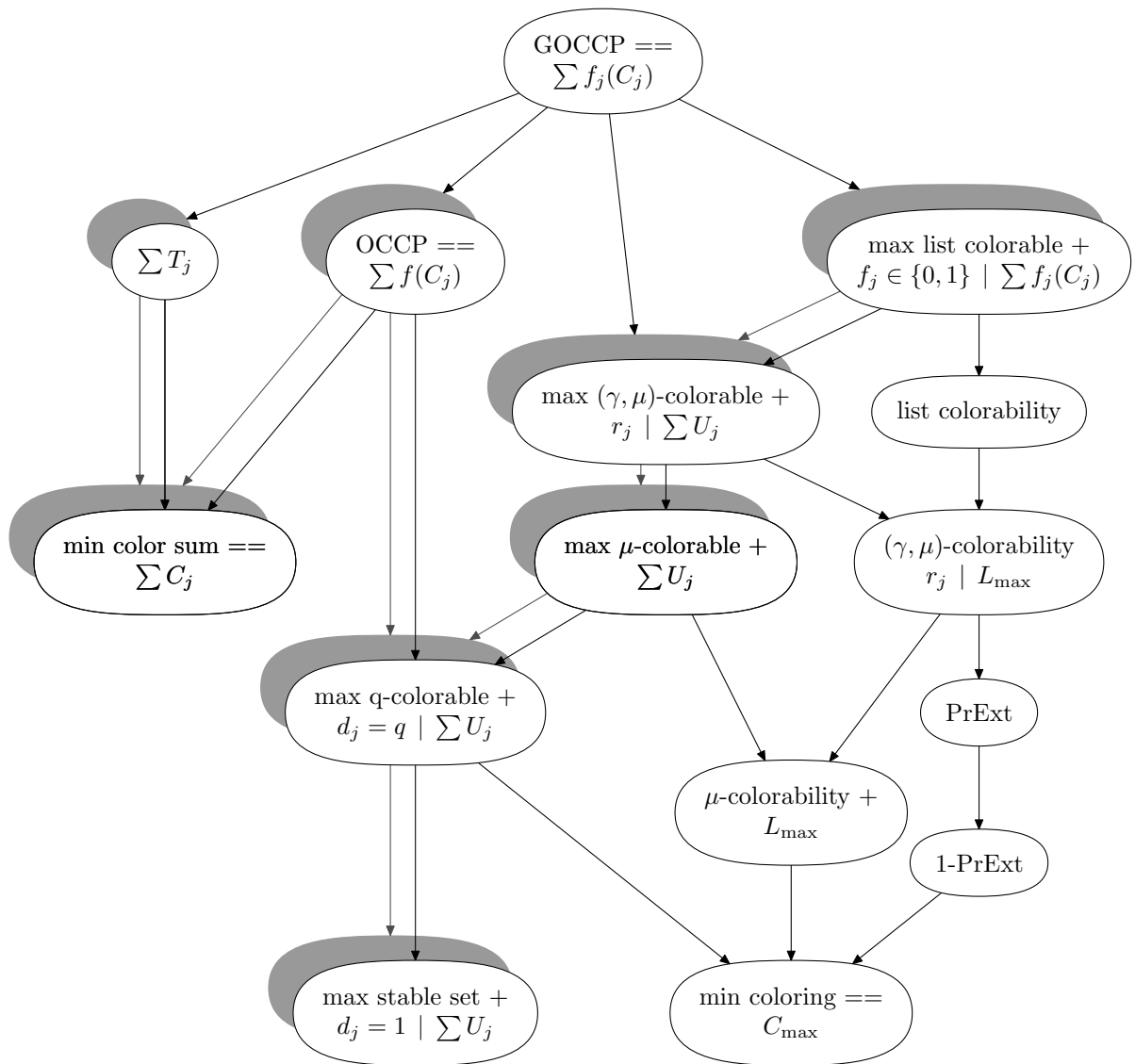


Figure 5.3 – Fonctions objectifs et réductions polynomiales élémentaires pour l’ordonnement chromatique dans le cas de temps d’opération unitaires. Les ombres correspondent aux versions pondérées de l’objectif (w_j). La dualité gain-coût est signalée par “+”, qui signifie que les deux objectifs sont complémentaires (*i.e.* leur somme admet une expression simple). “==” signale des définitions équivalentes pour la minimisation des coûts.

1 dans la plupart des cas, ou à $w(v)$ pour les cas pondérés). L'équation (5.6) a été prouvée dans la Proposition 5.2.

Considérons l'équation (5.4). Bien qu'elle soit, en un sens, un particulier de (5.6), nous en donnons une preuve car un des pas de la preuve consiste à intégrer des contraintes fortes (*i.e.* dates de disponibilité) à la fonction objectif.

Soit $r, d : V \rightarrow \mathbb{N}$ et ϕ une coloration de G . Chaque sommet v tel que $r(v) + 1 \leq \phi(v) \leq d(v)$ compte pour 1 dans la fonction objectif $\alpha_{(r,d)}(G)$ et les autres sommets comptent pour 0.

Inversement, pour le problème

$$B1 \mid col(G), p_S = 1, r_j \mid \sum U_j$$

les sommets en retard comptent pour 1 et les sommets tels que $r(v) + 1 \leq \phi(v) \leq d(v)$ comptent pour 0. Au sens strict de l'ordonnancement avec dates de disponibilités, si un sommet v est en avance (*i.e.* si $\phi(v) < r(v) + 1$), la coloration n'est pas réalisable. Cependant, étant donnée une coloration ϕ , on peut recolorer l'ensemble des sommets v tels que $\phi(v) < r(v) + 1$ de manière à ce que $\phi(v) > d(v)$. Autrement dit, on peut transformer tous les sommets en avance en sommets en retard. On peut donc supposer qu'il n'y a pas de sommets en avance, ce que l'on peut simuler en associant le coût 1 au fait que $\phi(v) < r(v) + 1$.

La somme de la fonction de gain (associée à $\alpha_{(r,d)}(G)$) et de la fonction de coût (associée à $B1 \mid col(G), p_S = 1, r_j \mid \sum U_j$) est donc égale à 1 indépendamment du sommet et de la couleur considérés. \square

Proposition 5.6 [*ConvexListCol*] et $B1 \mid col(G), p_S = 1, r_j \mid L_{\max}$ sont polynomialement équivalents (et l'équivalence est valide à l'intérieur d'une classe quelconque d'hypergraphes).

Preuve Soit G un (hyper-)graphe et $r, d : V \rightarrow \mathbb{N}$. [*ConvexListCol*] se réduit à $B1 \mid p_S = 1, r_j \mid L_{\max}$ car G admet une coloration ϕ telle que $r(v) + 1 \leq \phi(v) \leq d(v)$ si et seulement si l'optimum de L_{\max} est négatif ou nul. L'autre sens vient du fait que l'optimum de L_{\max} est égal au minimum des $t \in \mathbb{Z}$ tels que [*ConvexListCol*] soit réalisable pour la même instance dans laquelle on a remplacé tous les $d(v)$ par $d(v) + t$. \square

Proposition 5.7 [*PrExt*] se réduit polynomialement à [*ConvexListCol*]

Preuve Soit G un (hyper-)graphe et \mathcal{Q} une précoloration de G . Quitte à renommer les couleurs, supposons que la liste des précouleurs utilisées est $[1..|\mathcal{Q}|]$. Pour $k \geq |\mathcal{Q}|$, il existe une k -coloration qui étend \mathcal{Q} si et seulement si $L_{\max} \leq 0$ en choisissant $r_j = 0$ et $d_j = k$ pour les sommets qui ne sont pas précoloriés et en choisissant $r_j = i - 1$ et $d_j = i$ si le sommet j est précolorié avec la couleur i . Le nombre minimum de couleurs nécessaires pour colorier G en respectant \mathcal{Q} est donc le minimum des k tels que $L_{\max} = 0$. \square

5.1.3 Quelques résultats de complexité

Nous survolons quelques résultats de complexité et d'approximabilité pour les problèmes décrits dans la Figure 5.3.

En plus des réductions polynomiales, il existe d'autres réductions intéressantes entre des sous-problème de [GOCCP]. Par exemple, [MinColorSum] est approximable avec un facteur 1.796 dans toute classe de graphe pour laquelle [PartialqColor] est polynomial [78].

Théorème 5.8 *Le problème de [MinColSum] est NP-complet mais approximable avec ratio borné pour les graphes*

- parfaits ($3.591 \stackrel{[78]}{\geq} \rho$).
- bipartis ($27/26 \stackrel{[82]}{\geq} \rho \stackrel{[13]}{>} 1$).
- de comparabilité ($1.796 \stackrel{[78]}{\geq} \rho$).

Pour discuter de la complexité de [PrExt], on considère souvent que les précouleurs sont rangées par taille décroissante ($|Q_1| \geq |Q_2| \geq \dots$). [1-PrExt] est donc équivalent à [PrExt] avec la condition $|Q_1| \leq 1$.

Théorème 5.9 *Le problème de l'extension pré-colorée est NP-complet pour les graphes*

- parfaits (même si $|Q_1| \leq 2$ et $|Q_3| = 0$ [109]).
- bipartis (même si $|Q_1| \leq 1$ et $|Q_4| = 0$) [98, 24].
- LGBIP [72, 75]
- Line-graphes de bipartis complets [46]
- de permutation [101]
- d'intervalles (même si $|Q_1| \leq 2$) [20]
- d'intervalles propres [116]

Cependant, [1-PrExt] est polynomial dans les graphes triangulés [117] et [PrExt] est polynomial dans les graphes parfaits si $|Q_2| \leq 1$ et $|Q_3| = 0$ [109]. La Figure A.4 page 261 résume la complexité pour [PrExt] sur quelques classes de graphes parfaits.

Théorème 5.10 *[ListCol] est NP-complet même si l'on se restreint aux graphes*

- i) co-bipartis (Cas conforme de [3DM]).
- ii) bipartis complets [101]
- iii) split graphes complets.

Preuve iii) La réduction est quasi-identique à celle de [101] pour les bipartis complets. Soit C_1, C_2, \dots, C_m les m clauses d'une instance de [SAT] sur n variables. On construit le split graphe complet $G = ((S, K), E)$ tel que S et en bijection avec les clauses et K en bijection avec les variables. La liste de couleurs possibles pour le

sommet représentant la variable X_i est $\{X_i, \neg X_i\}^1$. Pour le sommet représentant la clause $X_1 \vee X_2 \vee \neg X_3$ par exemple, la liste est $\{X_1, X_2, \neg X_3\}$, ce que l'on généralise sans ambiguïté. On vérifie que le graphe est liste-colorable si et seulement si l'instance de [SAT] correspondante est satisfaisable. \square

Proposition 5.11 [101] [ListCol] est Polynomial si l'on se restreint aux forêts.

Proposition 5.12 [60] [GOCCP] est polynomial si l'on se restreint aux graphes triangulés sans-diamant.

Contrairement à [MinColorSum], [OCCP] n'est pas approximable correctement même pour des sous-classes de graphes parfaits [100]. En revanche, l'approximabilité de [ListCol] semble relativement ouverte de même que celle de [PartialqColor] ou encore de [PrExt].

5.2 Formule de Greene-Kleitman et box-perfection

Dans cette Section, nous présentons quelques résultats "classiques" concernant la formule de Greene-Kleitman [89, 90] pour la taille $\alpha_q(G)$ du sous-graphe q -chromatique de cardinalité maximum de G . La principale originalité de cette section consiste à pointer du doigt que la validité de cette formule peut souvent être prouvée aisément en observant que le système $Ax \leq \mathbf{1}$ est TDAU pour certaines classes de graphes (au lieu de seulement TDI pour les graphes parfaits).

Puisque le système

$$(5.7) \quad \begin{array}{ll} \text{(i)} & x_v \geq 0 \text{ pour tout } v \in V \\ \text{(ii)} & x(C) \leq 1 \text{ pour toute clique maximale } C \text{ de } G \end{array}$$

décrit l'enveloppe convexe des stables pour les graphes parfaits, on pourrait envisager que le système suivant donne l'enveloppe convexe de l'union de q stables.

$$(5.8) \quad \begin{array}{ll} \text{(i)} & 0 \leq x_v \leq 1 \text{ pour tout } v \in V \\ \text{(ii)} & x(C) \leq q \text{ pour toute clique maximale } C \text{ de } G \end{array}$$

Notons qu'en plus d'avoir changé les contraintes $x(C) \leq 1$ en $x(C) \leq q$ (transformation, qui seule, ne fait pas perdre la propriété TDI), on a ajouté les contraintes

¹La réduction de [101] est identique au détail près que dans [101], K induit un stable. Comme les listes associées aux sommets sont disjointes, l'adjacence sur K est sans importance.

$x_v \leq 1$. Un graphe est de **Greene-Kleitman**² si pour tout sous-graphe induit et pour tout q , la maximisation de $\mathbf{1}^T x$ sujet à (5.8) a la propriété de dualité entière. Autrement dit, G est de Greene-Kleitman si pour tout $U \subseteq V(G)$ et pour tout entier q , la cardinalité maximum de l'union de q stables est égale au minimum de

$$(5.9) \sum_{K \in \mathcal{Q}} \min\{q, |K|\}$$

où \mathcal{Q} parcourt l'ensemble des partitions en cliques de $G[U]$. Greene et Kleitman [89, 90] ont montré que les graphes de comparabilité et leurs compléments ont cette propriété³. En particulier, le système (5.8) décrit en fait l'enveloppe convexe de l'union de q -stables pour les graphes de (co)-comparabilité [69, 39]. En général, on cherche à résoudre le problème :

[PartialColor] Coloration partielle maximum (version décision)

Données : Un graphe G , deux entiers k et t

Résultat : Existe-t-il une k -coloration partielle de G qui couvre t sommets ?

Commentaires : Contient [Color] et stable maximum (pour $t = |V|$, resp. $k = 1$).

La première mauvaise nouvelle est que la formule de Greene-Kleitman ne marche pas pour les graphes parfaits en général, comme le montre la Figure 5.4. Pire, [ColPart] est NP-dur dans les split graphes [138]⁴.

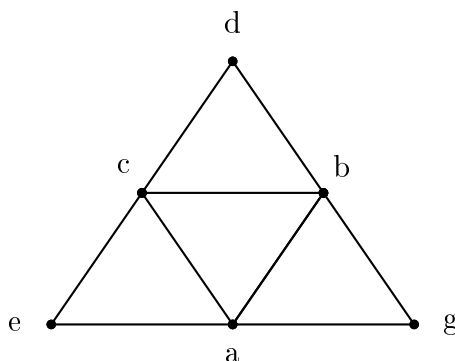


Figure 5.4 – Pour $q = 2$, $\max\{\mathbf{1}^T x \mid x \text{ sujet à (5.8)}\} = 9/2$ est atteint par $x_a = x_b = x_c := 1/2$ et $x_d = x_e = x_g := 1$. L'optimalité peut être prouvée à l'aide d'une solution duale, c'est-à-dire une partition fractionnaire en cliques : $y_C = 1/2$ si $C \in \{\{d\}, \{e\}, \{g\}, \{a, b, g\}, \{a, c, e\}, \{b, c, d\}\}$ et $y_C = 0$ sinon.

²Ceci est un néologisme.

³Ce résultat est sujet à de nombreuses généralisations, en dehors même du cadre des indépendants dans les graphes. La conjecture de Berge sur les partitions en chemin dans les graphes orientés est la plus célèbre [129].

⁴Bien que, dans les split graphes, le problème “dual”, de trouver une partition en clique qui minimise (5.9) soit trivial (Section 4.3.1.2).

On peut relier la non-intégralité du système (5.8), au fait que le système (5.7) n'est "que" TDI (et pas box-TDI) pour les graphes parfaits. Cela motive la définition des graphes *box-parfaits* comme les graphes pour lesquels le système (5.7) est box-TDI [39]. En effet, si (5.7) est box-TDI, alors (5.8) est TDI pour tout q . Cependant, les graphes box-parfaits forment une sous-classe stricte des graphes de Greene-Kleitman et aucune caractérisation n'est connue pour ces deux classes de graphes. Contrairement aux graphes parfaits, il n'est donc pas clair qu'il existe une classe naturelle et/ou aisément descriptible de graphes qui caractérise la validité des relations min-max données par le système (5.8). La Figure 5.5 montre les liens entre quelques classes de graphes reliées aux propriétés d'intégralité des systèmes (5.7) et (5.8). D'autres obstructions à la box-perfection sont mentionnées dans [39].

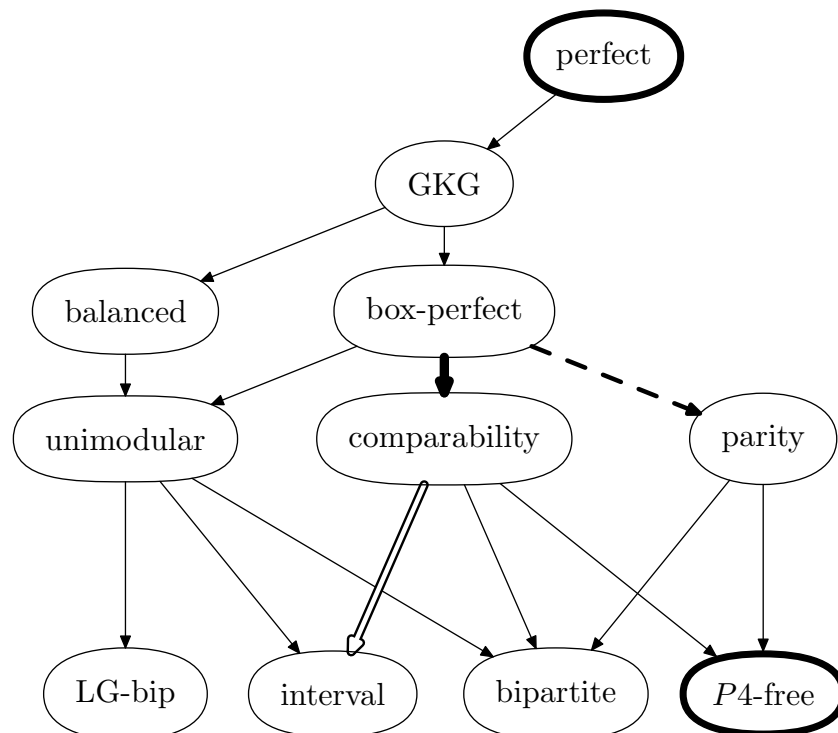


Figure 5.5 – Classes de graphes liées à l'intégralité des systèmes (5.7) et (5.8).

Proposition 5.13 *Les graphes équilibrés sont de Greene-Kleitman.*

Preuve Soit M la matrice d'incidence cliques-max-sommets d'un graphe équilibré. La matrice M' obtenue en collant la matrice identité de taille $|V(G)|$ au dessous de M est aussi équilibrée. \square

Conjecture 5.14 *Les graphes de parité sont box-parfaits.*

Pour prouver la conjecture 5.14, il suffirait de prouver que la box-perfection est préservée par collage par biparti.

Voyons maintenant des exemples qui montrent que les autres inclusions sont fausses.

Exemple 5.15 *A est une matrice équilibrée, minimalement non-TTU et M est la matrice d'incidence cliques-max-sommets d'un graphe G à 7 sommets. G est un graphe équilibré (et donc de Greene-Kleitman), mais G est minimalement non-unimodulaire et minimalement non-box-parfait.*

$$(5.10) \quad A := \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \quad M := \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Preuve M n'a pas la matrice d'incidence d'un cycle impair comme sous-matrice et est donc équilibrée. Vérifions que G n'est pas box-parfait (et donc pas unimodulaire non plus). Il suffit pour cela de choisir $q = 2$ et $w^T = (2, 1, 1, 1, 1, 1, 1)$. On a alors $\max w^T x$ sujet à (5.8) = $11/2$ atteint avec $x^T = 1/2(1, 1, 1, 1, 2, 2, 2)$. Une solution duale optimale est donnée par $y_i = 1/2$ pour chacune des lignes de M ainsi que pour les trois dernières lignes de la matrice identité de taille 7 placée sous M dans le système (5.8) et $y_i = 0$ pour les 4 premières lignes de cette matrice identité.

G est donc non-box-parfait (et donc non-unimodulaire). On peut vérifier que G est minimalement non-unimodulaire (et donc minimalement non-box-parfait) de la manière suivante. A symétrie près, G a trois type de sommets (correspondants respectivement à la première colonne de M , aux 3 suivantes et aux trois dernières). Notons a, b, c respectivement ces trois types de sommets. $G - a$ est un LGBIP. $G - b$ et $G - c$ sont des graphes d'intervalles. \square

Exemple 5.16 $\overline{3K_2}$ (le complément du graphe formé de 6 sommets et de trois arêtes disjointes) est un graphe sans- P_4 (et donc box-parfait) qui est minimalement non-équilibré.

5.3 Dates déchéances et graphes sans- P_4

Nous donnons ici une formulation polyédrale du problème d'ordonnancement avec dates d'échéances $B1 | col(G), p_S = 1 | L_{\max}$ et montrons que la validité de cette formulation caractérise les graphes sans- P_4 .

Proposition 5.17 *Soit $G = (V, E)$ un graphe et $d : V \rightarrow \mathbb{N}$. Si G admet une coloration $\phi : V \rightarrow \mathbb{N}$ telle que $\phi(v) \leq d(v)$ pour tout $v \in V$ alors $|K| \leq \max\{d(v) | v \in K\}$ pour toute clique K de G .*

Preuve Si ϕ respecte les dates d'échéances et $X \subseteq V$, on a $\phi(X) \subseteq [1.. \max\{d(v) | v \in X\}]$. Comme il faut $|K|$ couleurs distinctes pour colorier une clique K de G , on a

$|\phi(K)| = |K|$. On doit donc aussi avoir

$$|\phi(K)| = |K| \leq \lceil \max\{d(v) \mid v \in K\} \rceil$$

□

La condition de la Proposition 5.17 n'est pas suffisante en général, comme le montre la Figure 5.6. Cependant, elle est suffisante pour les graphes sans- P_4 .

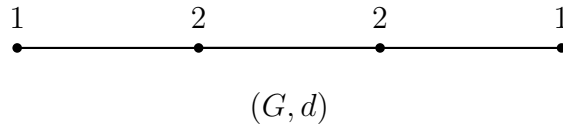


Figure 5.6 – Graphe pondéré (G, d) tel que $|K| \leq \max\{d(x) \mid x \in K\}$ pour toute clique K de G mais qui n'a pas de coloration $\phi : V \rightarrow [1..2]$ telle que $\phi(v) \leq d(v)$ pour tout v .

La caractérisation suivante a été observée indépendamment dans [26], sous la terminologie “ μ -coloration”.

Théorème 5.18 *Soit $G = (V, E)$ un graphe. Les propriétés suivantes sont équivalentes :*

- i) G est sans- P_4
- ii) Pour tout $d : V \rightarrow \mathbb{N}$, il existe une coloration $\phi : V \rightarrow \mathbb{N}$ telle que $\phi(v) \leq d(v)$ pour tout v si et seulement si pour toute clique K de G , on a $|K| \leq \max\{d(w) \mid w \in K\}$.

Preuve ii) \implies i) Si G a un P_4 induit, les dates d'échéances données dans la Figure 5.6 (étendues avec $d(v) = |V|$ sur les autres sommets) donnent une obstruction à l'existence d'une coloration sans que la condition de clique ne soit violée.

i) \implies ii) Soit G sans- P_4 et $d : V(G) \rightarrow \mathbb{N}$ tels que $|K| \leq \max\{d(v) \mid v \in K\}$ pour toute clique K de G . Soit (v_1, v_2, \dots, v_n) un ordre qui étend l'ordre non-décroissant des $d(v)$ sur V . Montrons que l'algorithme glouton appliqué à (v_1, v_2, \dots, v_n) donne une coloration ϕ telle que $\phi(v) \leq d(v)$ pour tout v . Sinon, il existe $t \in [1..n]$ tel que $\phi(v_t) > d(v_t)$. Comme l'algorithme glouton donne une coloration optimale de $G[\{v_1, v_2, \dots, v_t\}]$ et que G est parfait, il existe une clique K_t de taille $\phi(v_t)$ dans $G[\{v_1, v_2, \dots, v_t\}]$. Par construction de l'ordre (v_1, v_2, \dots, v_n) , on a $\max\{d(v_i) \mid i \leq t\} = d(v_t)$. On a $|K_t| = \phi(v_t) > d(v_t)$, contradiction. □

Corollaire 5.19 *Le problème $B1 \mid col(G), p_S = 1 \mid L_{\max}$ est polynomial si on se restreint aux graphes sans- P_4 .*

Preuve Soit G un graphe sans- P_4 et $d : V(G) \rightarrow \mathbb{N}$. La valeur optimale de L_{\max} est égale au minimum des $t \in \mathbb{Z}$ tels que le graphe admette une coloration respectant les dates échues $d_t(v) = d(v) + t$. Une telle coloration peut être trouvée de manière gloutonne si elle existe (voir la preuve du Théorème 5.18). \square

Plus généralement, il semble que pour les graphes sans- P_4 , on puisse traiter le problème $B1 \mid col(G), p_S = 1 \mid \sum w_j U_j$ avec la technique discutée dans cette section. Toutefois, l'algorithme glouton ne semble pas s'étendre directement. A la place nous proposons seulement une éventuelle formule min-max pour ce problème :

Conjecture 5.20 *Pour tout graphe sans- P_4 et tout $d : V \rightarrow \mathbb{N}$, le système suivant, à variables dans \mathbb{R}^V est box-TDI :*

$$(5.11) \quad x(K) \leq \max\{d(v) \mid v \in K\} \text{ pour toute clique } K \text{ de } G$$

Une des motivations pour la Conjecture 5.20 est que pour un graphe sans- P_4 , un vecteur $x \in \{0, 1\}^V$ qui satisfait (5.11) admet une coloration qui respecte les dates d'échéances (voir preuve du Théorème 5.18). La Conjecture 5.20 implique donc une formule min-max pour le maximum de sommets que l'on peut colorier avant leur date échue (cela serait étendrait donc aussi [26], en donnant une formule pour le sous-graphe μ -colorable de poids maximum). Comme cas particulier de la Conjecture 5.20, si (5.11) est TDI, on en dérive une formule min-max pour le problème [MaxColor] sur le complément du graphe G (voir Chapitre 4).

5.4 Extension précolorée sur les co-Meyniel

Dans cette section, nous décrivons le problème de l'extension précolorée. L'étude de ce problème peut être motivée par la problématique plus large dont il relève : les "contraintes de préaffectation" en optimisation. Nous montrons comment réduire l'extension précolorée à la coloration classique dans un graphe auxiliaire et caractérisons les compléments de graphes de Meyniel en terme de cette réduction.

Une **pré-coloration** d'un graphe $G = (V, E)$ est une coloration d'un sous-graphe induit de G . Autrement dit, une pré-coloration est une collection $\mathcal{Q} = \{C_1, \dots, C_m\}$ de stables deux-à-deux disjoints de G . On dit qu'une k -coloration $\{S_1, \dots, S_k\}$ de G **étend** \mathcal{Q} si pour chaque $j = 1, \dots, m$ on a $C_j \subseteq S_j$. Le problème [PrExt] est défini comme suit :

[PrExt] Extension pré-colorée (version décision)

Données : Un graphe G , un entier k et une pré-coloration \mathcal{Q} de G utilisant seulement des couleurs dans $\{1, \dots, k\}$.

Résultat : Existe-t-il une coloration de G qui étend \mathcal{Q} ?

Commentaires : NPC. Une k -coloration qui étend \mathcal{Q} est un certificat NP. [PrExt] généralise [Color] (qui revient à prendre $\mathcal{Q} = \emptyset$).

Nous étudions principalement la version sur le graphe complémentaire, avec le problème de partition en cliques.

5.4.1 Petite philosophie à l'usage des (non-)spécialistes : l'ubiquité des contraintes de préaffectation.

Dans des contextes appliqués, il arrive souvent qu'on ait besoin de contraintes exotiques pour modéliser un problème d'optimisation. En particulier, pour des raisons techniques, historiques ou sociales, l'organisation du système que nous désirons optimiser est parfois partiellement fixée à l'avance. Par exemple, lors de la réorganisation des chercheurs dans les bureaux d'un laboratoire de mathématique, il arrive que l'on réaffecte seulement les bureaux des non-permanents. En termes de programmation mathématique, on peut modéliser ces contraintes de préaffectations en fixant la valeur de certaines variables avant le processus d'optimisation. Bien que ces préaffectations réduisent la taille du problème considéré, elles peuvent altérer les propriétés structurelles de telle manière que la complexité algorithmique passe de polynomiale à NP-dure. Le problème de l'extension précolorée illustre bien ce phénomène d'augmentation de complexité, car s'il est trivial de colorier un graphe biparti optimalement, [PrExt] est NP-complet sur les graphes bipartis.

Le Sudoku peut être vu comme un cas particulier de [PrExt] dans lequel $V(G)$ est la grille 9×9 et deux sommets sont adjacents s'ils sont sur la même ligne, la même colonne ou le même sous-carré. Dans le Sudoku, on cherche à étendre une coloration partielle en une 9-coloration. Bien qu'il soit facile de trouver une 9-coloration de la grille vierge, il est parfois difficile de trouver une extension d'une précoloration donnée. Certaines généralisations (naturelles) du Sudoku sont d'ailleurs NP-dures.

5.4.2 Contraction, PrExt-perfection et graphes de co-Meyniel

Nous expliquons comment [PrExt] se réduit à [Color] dans un graphe auxiliaire. Puis nous utilisons cette technique pour donner une nouvelle caractérisation des graphes de Meyniel. La polynomialité de [PrExt] sur les graphes de co-Meyniel est une conséquence.

Etant donné une pré-coloration $\mathcal{Q} = \{C_1, \dots, C_m\}$ de G , on définit le graphe G/\mathcal{Q} dont les sommets sont obtenus à partir de V en contractant, pour chaque j , l'ensemble des sommets de C_j en un seul sommet c_j , les autres sommets restant inchangés. On met une arête sur toutes les paires de c_j . Les arêtes contenues dans $V \setminus (C_1 \cup \dots \cup C_m)$ sont inchangées. Il y a une arête dans G/\mathcal{Q} entre $v \in V \setminus (C_1 \cup \dots \cup C_m)$ et c_j si v est adjacent à au moins un sommet de C_j . La technique de contraction est illustrée dans la paire de Figures 5.7 et 5.8. La preuve du lemme suivant est directe.

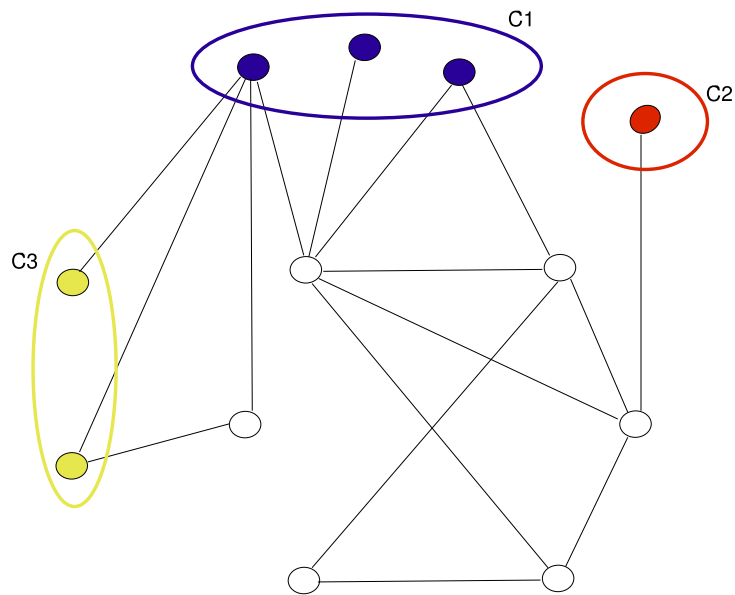


Figure 5.7 – Un graphe G et une précoloration \mathcal{Q} de G

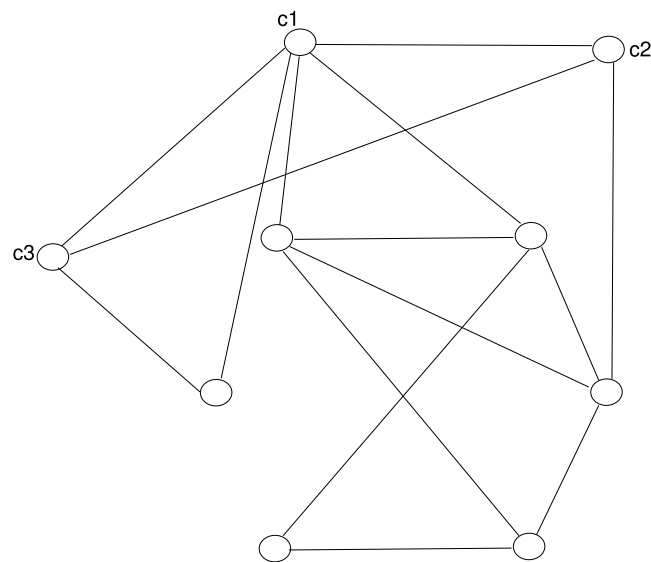


Figure 5.8 – Le graphe contracté G/\mathcal{Q} de la Figure 5.7

Lemme 5.21 [106] *Pour tout graphe G et toute pré-coloration \mathcal{Q} de G , l'ensemble des colorations qui étendent \mathcal{Q} est en bijection avec les colorations de G/\mathcal{Q} . En particulier, le nombre minimum de couleurs qu'il faut pour étendre \mathcal{Q} est égal à $\chi(G/\mathcal{Q})$. \square*

La portée du Lemme 5.21, en vue de transposer les résultats de complexité entre [Color] et [PrExt], semble très générale. En particulier, il semble naturel de demander dans quels cas le graphe contracté est un graphe parfait, puisqu'on peut alors résoudre [PrExt] en temps polynomial en coloriant le graphe contracté avec la méthode de Grötschel-Lovász-Schrijver [91].

Théorème 5.22 [106] *G/\mathcal{Q} est un graphe parfait pour toute pré-coloration \mathcal{Q} de G si et seulement si G est un graphe de co-Meyniel.*

La polynomialité découle immédiatement :

Corollaire 5.23 [106] *[PrExt] est polynomial pour les graphes de co-Meyniel.*

La nécessité d'être un graphe de co-Meyniel dans le Théorème 5.22 se vérifie aisément car les minimaux exclus pour les graphes de co-Meyniel sont les trous impairs et les chemins impairs de longueur supérieure ou égale à 5. Si un graphe G a un trou impair, la pré-coloration \mathcal{Q} vide donne G/\mathcal{Q} imparfait. Si G a un chemin impair $v_1 \cdots v_t$, la pré-coloration $\mathcal{Q} = \{\{v_1\}, \{v_t\}\}$ donne un trou impair dans le graphe G/\mathcal{Q} (la contraction a pour seul effet d'ajouter l'arête $v_1 v_t$).

Nous avons dérivé la suffisance d'être co-Meyniel par une minutieuse inspection [106], qui montre que lors de la contraction des graphes de co-Meyniel, on ne crée pas de trou impair, ni d'antitrou (pair ou impair). En fait, comme on ne crée pas non plus de prismes, on a un résultat plus précis :

Théorème 5.24 [106] *Soit G un graphe de co-Meyniel et \mathcal{Q} une pré-coloration de G , alors G/\mathcal{Q} est le complément d'un graphe d'Artemis.*

Il reste cependant ouvert de caractériser la classe des graphes que l'on peut obtenir par contraction d'un graphe de co-Meyniel. Un des intérêts du Théorème 5.24 est qu'on peut se passer de la preuve du Théorème fort des graphes parfaits pour terminer la preuve du Théorème 5.22 et utiliser seulement la preuve de la perfection des graphes d'Artemis.

Le reste de la Section 5.4 consiste en une traduction de la preuve du Théorème 5.24 donnée dans [106]. La discussion est menée dans le graphe complémentaire, pour le problème correspondant de partition en cliques, à l'aide de la co-contraction. Si \mathcal{Q} est un ensemble de cliques de G deux-à-deux disjointes, le graphe cocontracté est $G^{\mathcal{Q}} := \overline{(G/\mathcal{Q})}$.

Lemme 5.25 *Dans un graphe de Meyniel G , soit $P = p_0 \cdots p_n$ un chemin induit et x un sommet de $V(G) \setminus V(P)$ adjacent à p_0 et p_n . Alors soit x est adjacent à*

tous les sommets de P , soit n est pair et $N(x) \cap V(P) \subseteq \{p_{2i} \mid i = 0, \dots, n/2\}$.

Preuve Appelons *segment*, un sous-chemin S de longueur au moins 1 de P tel que $G[S + x]$ est un circuit. Le chemin P est partitionné en segments. Soit $p_h \dots p_j$ un segment tel que $j - h \geq 2$. Alors x, p_h, \dots, p_j induit un trou, donc $j - h$ est pair. Donc tout segment a une longueur paire ou égale à 1. Supposons qu'il y ait un segment de longueur 1 et un segment de longueur paire. Alors il y a aussi deux tels segments consécutifs. Avec x ces deux segments induisent une maison, ce qui contredit le fait que G est de Meyniel. En conclusion, soit tous les segments ont longueur 1 (i.e., x est adjacent à tous les sommets de P), soit tous les segments ont longueur paire. \square

Lemme 5.26 Dans un graphe de Meyniel G , soit H un trou pair et x un sommet de $V(G) \setminus V(H)$ adjacents à deux sommets consécutifs de H . Alors x est adjacent soit à tous les sommets de H soit à exactement trois sommets consécutifs de H .

Preuve Soient x_1, \dots, x_n les sommets de H ordonnés de manière circulaire. Supposons que x soit adjacent à x_1 et x_2 mais pas à tous les sommets de H et soit x_i un sommet de H qui n'est pas voisin de x . Supposons que x soit adjacent à un sommet x_j tel que $j \notin \{1, 2, 3, n\}$. Par symétrie, on peut supposer que $i < j$. Alors soit $x_1 \dots x_j$ soit $x_2 \dots x_j$ est un chemin induit impair. Dans les deux cas, x est adjacent aux deux extrémités de ce chemin sans être adjacent à tous ses sommets, ce qui contredit le Lemme 5.25. Donc $N(x) \cap V(H) \subseteq \{x_1, x_2, x_3, x_n\}$. Si x n'est adjacent ni à x_3 ni à x_n , alors $V(H) \cup \{x\}$ induit une maison. Si x est adjacent à la fois à x_3 et x_n alors x, x_3, \dots, x_n induit un trou impair. \square

Lemme 5.27 Dans un graphe de Meyniel G , soit Q une clique, $P = p_0 \dots p_n$ un chemin induit de $G \setminus Q$ et z un sommet extérieur à $Q \cup V(P)$. Supposons que z et p_0 soient adjacents à tous les sommets de Q , que z ne soit pas adjacent à p_1 , et qu'il existe $q \in Q$ adjacent à p_0 et p_n mais pas à p_1 . Alors p_n est adjacent à tous les sommets de Q .

Preuve Nous prouvons ce lemme par induction sur n . Supposons qu'un sommet $q' \in Q$ ne soit pas adjacent à p_n . D'après le Lemme 5.25 appliqué à P et q , puisque q est adjacent à p_0, p_n mais pas à p_1 , le chemin P a une longueur paire et $N(q) \cap V(P) \subseteq \{p_{2i} \mid i = 1, \dots, n/2\}$. Soit $j < n$ le plus grand entier tel que q soit adjacent à p_j . Puisque j est pair, $\{p_j, \dots, p_n, q\}$ induit un trou pair C . Supposons que $j = 0$. Alors q' est adjacent à q et p_0 mais pas à p_n . D'après le Lemme 5.26 appliqué à C et q' , ce dernier est adjacent à p_1 mais à aucun de $\{p_2, \dots, p_n\}$. Appelons C' le trou pair induit par $V(C) + q' - p_0$. Le sommet z est adjacent à q et q' dans C' mais pas à p_1 . D'après le Lemme 5.26, z est adjacent à p_n mais à aucun sommet de $\{p_1, \dots, p_{n-1}\}$. Mais alors $V(C) \cup \{z\}$ induit une maison, ce qui contredit que $j = 0$. Donc $j \geq 2$. Par l'hypothèse d'induction, p_j est adjacent à tous les sommets de Q . Puisque q' est adjacent à q et p_j sur C mais pas à p_n , le Lemme 5.26 implique que q' est adjacent à p_{j+1} mais à aucun sommet de $\{p_{j+2}, \dots, p_n\}$. Mais alors $\{p_0, q, q', p_{j+1}, \dots, p_n\}$ induit une maison, ce

qui contredit l'existence de q' . \square

Lemme 5.28 *Dans un graphe de Meyniel G , soit Q une clique, X un ensemble connexe de sommets de $G \setminus Q$ et z un sommet extérieur à $Q \cup X$. Supposons que z soit adjacent à tous les sommets de Q mais à aucun de X et que chaque sommet de X ait un non-voisin dans Q . Alors un des sommets de Q n'a aucun voisin dans X .*

Preuve Nous prouvons le lemme par induction sur la taille de X . Si $|X| = 1$, l'énoncé est creux. Supposons donc que $|X| \geq 2$. Soit x, x' deux sommets distincts de X tels que $X \setminus \{x\}$ et $X \setminus \{x'\}$ soient connexes (par exemple, prendre deux feuilles d'un arbre couvrant de X pour x, x'). Par l'hypothèse d'induction, il y a deux sommets q, q' de Q tels que q n'a pas de voisins dans $X \setminus \{x\}$ et q' n'a pas de voisins dans $X \setminus \{x'\}$. Si q n'est pas voisin de x ou si q' n'est pas voisin de x' , alors le lemme est prouvé. Supposons donc que q soit voisin de x et que q' soit voisin de x' . Soit P le plus court chemin de x à x' dans X . Alors soit $V(P) \cup \{q, q'\}$ induit un trou impair, soit $V(P) \cup \{q, q', z\}$ induit une maison, contradiction. \square

Lemme 5.29 *Soit G un graphe de Meyniel et \mathcal{Q} une famille de cliques deux-à-deux disjointes de G . Alors le graphe cocontracté $G^{\mathcal{Q}}$ ne contient pas d'antitrous de taille supérieure ou égale à 6.*

Preuve Supposons que $G^{\mathcal{Q}}$ contienne un antitrou A de taille au moins 6. Puisque l'ensemble des cliques de \mathcal{Q} est cocontracté en un stable, il y a au plus deux sommets de A qui proviennent de la cocontraction d'une clique et s'il y a deux tels sommets dans A , ils sont consécutifs dans l'ordre cyclique de \bar{A} . S'il y a 5 sommets consécutifs de A qui ne proviennent pas de la cocontraction d'une clique ils induisent une maison de G , contradiction. Donc il n'y a pas cinq tels sommets. A est donc de taille 6 et a exactement deux sommets provenant de la cocontraction d'une clique. Soit x_1, \dots, x_6 les sommets de A ordonnés de manière cyclique, tels que x_1 et x_2 soient des sommets cocontractés. Soit C_1 la clique dont la cocontraction ait donné x_1 . Puisque x_1 et x_6 ne sont pas adjacents, x_6 n'est pas adjacent à tous les sommets de C_1 , donc il y a un sommet q_1 de C_1 qui n'est pas adjacent à x_6 dans G . Donc $G[q_1, x_3, x_4, x_5, x_6]$ induit une maison de G , contradiction. \square

Lemme 5.30 *Soit G un graphe de Meyniel et \mathcal{Q} une famille disjointe de cliques de G . Alors le graphe cocontracté $G^{\mathcal{Q}}$ ne contient pas de trou impair.*

Preuve Nous prouvons le lemme par induction sur $m = |\mathcal{Q}|$. Si $m = 0$, alors $G^{\mathcal{Q}} = G$ et le lemme est prouvé. Supposons donc que $m > 0$ et soit $\mathcal{Q} = \{C_1, \dots, C_m\}$. Supposons que $G^{\mathcal{Q}}$ contienne un trou impair \mathcal{H} . Soit $\{x_1, \dots, x_n\}$ l'ensemble des sommets de \mathcal{H} ordonnés de manière cyclique. Pour chaque $j = 1, \dots, m$, on peut supposer que le sommet créé par la cocontraction de C_j appartient à \mathcal{H} (sinon \mathcal{H} est un trou impair de $G^{\mathcal{Q}} \setminus \{C_j\}$, ce qui contredit l'hypothèse d'induction. Soit x_{i_j} le sommet de \mathcal{H} créé par la cocontraction de C_j , et s.p.d.g, supposons que $1 < i_1 < i_2 < \dots < i_m \leq n$.

Supposons que $m = 1$. On peut supposer que $i_1 = 1$. Puisque x_3 et x_1 ne sont pas adjacents dans $G^\mathcal{Q}$, x_3 n'est pas voisin de tous les sommets de C_1 dans G . Il y a donc un sommet q_1 de C_1 qui n'est pas voisin de x_3 dans G . Le chemin $P = x_2 \cdots x_n$ est induit est impair, et q_1 est adjacent aux deux extrémités de P mais pas au sommet x_3 de P , ce qui contredit le Lemme 5.25. Donc $m \geq 2$.

Les sommets cocontractés x_{i_1}, \dots, x_{i_m} induisent un stable dans \mathcal{H} . Donc pour tout j , on a $i_{j+1} - i_j \geq 2$. Puisque n est impair et que $m \geq 2$, il existe j tel que $i_{j+1} - i_j$ est impair (et donc $i_{j+1} - i_j \geq 3$). S.p.d.g, nous pouvons supposer que $i_2 - i_1$ est impair et que $i_1 = 1$ (donc $i_2 \geq 4$). Soit R le chemin impair $x_2 \cdots x_{i_2-1}$.

Puisque x_3 et x_1 ne sont pas voisins dans $G^\mathcal{Q}$, il y a un sommet q_1 de C_1 qui n'est pas adjacent à x_3 dans G . De même, il y a un sommet q_2 de C_2 qui n'est pas adjacent à x_{i_2-2} dans G . De plus, si $m \geq 3$, on peut appliquer le Lemme 5.28 à la clique C_j , l'ensemble connexe R et x_{i_j-1} , ce qui implique que :

(5.12)

Pour $j = 3, \dots, m$, il y a un sommet $q_j \in C_j$ adjacent à aucun sommet de R .

Maintenant, choisissons des sommets y_1, \dots, y_n de G de la manière suivante. Pour $k = 1, \dots, n$, s'il existe j tel que $i_j = k$, posons $y_k := q_j$; sinon posons $y_k := x_k$. Les sommets choisis y_1, \dots, y_n induisent un cycle impair H de G . Notons que, dans H , le sommet y_2 est adjacent seulement à y_1, y_3 et éventuellement à $y_{i_2} = q_2$ (d'après (5.12)). Nous montrons maintenant que

(5.13) *Tout voisin de q_1 dans $V(H) \setminus \{y_2, y_n\}$ est dans $\{q_2, \dots, q_m\}$.*

Soit u un voisin de q_1 dans $V(H) \setminus \{y_2, y_n\}$. Puisque x_2 et x_{i_2} ne sont pas adjacents dans $G^\mathcal{Q}$, il y a un unique sommet q'_2 de C_2 qui n'est pas adjacent à y_2 dans G . Le sous-graphe de G induit par $V(H) \cup \{q'_2\} \setminus \{y_1, y_n, q_2\}$ est connexe, donc il contient un plus court chemin U de y_2 à u . Puisque y_3 est le seul voisin de y_2 dans ce sous-graphe, y_3 est un voisin de y_2 dans U . Le Lemme 5.27 peut être appliqué à C_1 , U et y_n , ce qui implique que u est adjacent à tout C_1 . Donc u doit être dans $\{q_2, \dots, q_m\}$ sinon q_1 et u seraient adjacents dans $G^\mathcal{Q}$. Cela prouve (5.13). De manière similaire, on a :

(5.14) *Tout voisin de q_2 dans $V(H) \setminus \{y_{i_2-1}, y_{i_2+1}\}$ est dans $\{q_1, q_3, \dots, q_m\}$.*

Maintenant, chacun de y_2, \dots, y_{i_2-1} a degré 2 dans H .

Colorons en bleu certains sommets du chemin $\mathcal{H} \setminus x_1 = x_2 \cdots x_n$ de $G^\mathcal{Q}$ de la manière suivante. Les sommets x_2 et x_n sont coloriés en bleu. Pour $j = 2, \dots, m$, le sommet x_{i_j} est colorié en bleu si et seulement si tous les sommets de la clique C_j correspondante sont voisins de q_1 . Tous les autres sommets de \mathcal{H} restent non-coloriés. Appelons segment bleu, tout sous-chemin de longueur au moins 1 dans $\mathcal{H} \setminus x_1$ dont les deux extrémités sont bleues et dont les autres sommets ne sont pas

coloriés. Puisque $\mathcal{H} \setminus x_1$ est de longueur impaire et ses extrémités sont bleues, il a un segment bleu impair. Soit $x_h \cdots x_i$ un segment bleu impair, tel que $2 \leq h < i \leq n$. Supposons que $i - h \geq 3$. Alors (5.13) implique que $q_1, x_h, x_{h+1}, \dots, x_{i-1}, x_i$ induit un trou impair dans $G^\mathcal{Q} \setminus \{C_1\}$, ce qui contredit l'hypothèse d'induction sur $|\mathcal{Q}|$. On doit donc avoir $i - h = 1$. Puisque $i_{j+1} - i_j \geq 2$ pour tout j et $i_2 \geq 4$, cela est possible seulement si $h = n - 1$. Cela implique que $x_{n-1}x_n$ est le seul segment bleu impair et que tout sommet bleu x_k différent de x_n a k pair.

De manière similaire, nous colorons en rouge certains sommets du chemin $\mathcal{H} \setminus x_{i_2}$ de $G^\mathcal{Q}$ de la manière suivante. Les sommets x_{i_2-1} et x_{i_2+1} sont coloriés en rouge. Pour $j = 1, 2, \dots, m$ et $j \neq 2$, le sommet x_{i_j} est colorié rouge si et seulement si tous les sommets de la clique C_j correspondante sont adjacents à q_2 . Appelons segment rouge, tout sous-chemin de longueur supérieure ou égale à 1 de $\mathcal{H} \setminus x_{i_2}$ dont les deux extrémités sont rouges et dont les sommets intérieurs ne sont pas rouges. Comme dans le cas bleu, nous obtenons que $x_{i_2+1}x_{i_2+2}$ est le seul segment rouge impair, et que tout sommet rouge x_l différent de x_{i_2-1} et de x_{i_2+1} a soit l pair soit $l = 1$.

Si $i_2 = n - 1$, alors $m = 2$ et $V(R) \cup \{q_1, q_2, x_n\}$ induit un trou impair (si q_1, q_2 ne sont pas adjacents) ou une maison (si q_1, q_2 sont adjacents) dans G , contradiction. Supposons donc que $i_2 \leq n - 3$. Puisque x_{i_2+2} est rouge et que x_{n-1} est bleu, il y a un sous-chemin $x_k \cdots x_l$ de $x_{i_2+2} \cdots x_{n-1}$ tel que x_k est rouge, x_l est bleu, et aucun des sommets intérieurs de $x_k \cdots x_l$ n'est colorié. D'après les deux paragraphes précédents, k, l sont tous deux pairs. Si $k = l$, alors (5.13) implique qu'il y a une clique C_j telle que $k = i_j$, et alors $V(R) \cup \{q_1, q_2, q_j\}$ induit un trou impair (si q_1 et q_2 ne sont pas adjacents) ou une maison (si q_1 et q_2 sont adjacents) de G , contradiction. Donc $k \neq l$. Si q_1 et q_2 sont adjacents, alors (5.13) et (5.14) impliquent que $\{q_1, x_k, \dots, x_l, q_2\}$ induit un trou impair de $G^\mathcal{Q} \setminus \{C_1, C_2\}$, ce qui contredit l'hypothèse d'induction sur $|\mathcal{Q}|$. Si q_1 et q_2 ne sont pas adjacents, alors $V(R) \cup \{q_1, x_k, \dots, x_l, q_2\}$ induit un trou impair dans $G^\mathcal{Q} \setminus \{C_1, C_2\}$, encore une contradiction. Cela termine la preuve du Lemme 5.30 \square

Lemme 5.31 *Soit G un graphe de Meyniel et \mathcal{Q} une famille de cliques disjointes de G . Alors le graphe cocontracté $G^\mathcal{Q}$ ne contient pas de prisme.*

Preuve Supposons que $G^\mathcal{Q}$ contient un prisme K formé par des chemins $P_1 = u_0 \cdots u_r$, $P_2 = v_0 \cdots v_s$, $P_3 = w_0 \cdots w_t$, avec $r, s, t \geq 1$ et des triangles $A = \{u_0, v_0, w_0\}$ et $B = \{u_r, v_s, w_t\}$. D'après le Lemme 5.29, K n'est pas un antitrou sur 6 sommets et on peut donc supposer qu'un de r, s, t n'est pas égal à 1. Soit $\mathcal{Q} = \{C_1, \dots, C_m\}$. Nous avons $m \geq 1$ puisqu'un graphe de Meyniel ne contient pas de prismes (car ceux-ci contiennent une maison). Pour chaque $j = 1, \dots, m$, appelons c_j le sommet de $G^\mathcal{Q}$ qui provient de la cocontraction de C_j , et soit $C = \{c_1, \dots, c_m\}$. Par le Lemme 5.30, $G^\mathcal{Q}$ ne contient pas de trou impair, donc r, s, t ont la même parité. Notons que $V(K) \setminus C \subset V(G)$ et que $N(c_j) \subset V(G)$ pour chaque $j = 1, \dots, m$ (puisque C est un stable). Montrons que

$$(5.15) \quad |A \cap C| = 1 \text{ et } |B \cap C| = 1.$$

Notons que $|A \cap C| \leq 1$ et $|B \cap C| \leq 1$ puisque A, B sont des cliques et C est un stable de $G^\mathcal{Q}$. Maintenant, supposons à symétrie près, que $A \cap C = \emptyset$. Pour chaque $j = 1, \dots, m$, nous pouvons appliquer le Lemme 5.28 dans G à la clique C_j , l'ensemble connexe $A \setminus N(c_j)$ et un voisin z de c_j dans K (plus précisément : si $c_j = u_i$ avec $i < r$, alors prenons $z = c_{i+1}$; si $c_j = u_r$ et si $s \geq 2$ ou $t \geq 2$, prenons $z = v_s$ ou $z = w_t$ respectivement ; si $c_j = u_r$ et $s = t = 1$, alors $r \geq 3$ et prenons $z = u_{r-1}$; un z similaire existe si $c_j \in V(P_2) \cup V(P_3)$). Le Lemme 5.28 implique qu'il y a un sommet $q_j \in C_j$ qui n'est voisin d'aucun sommet de $\mathcal{A} \setminus N(c_j)$. Soit P le sous-graphe de G induit par $(V(K) \setminus C) \cup \{q_1, \dots, q_m\}$. Soit u'_1 le voisin de u_0 dans $P \setminus \{v_0, w_0\}$ (donc u'_1 est soit u_1 soit un q_j), et de même soit v'_1 le voisin de v_0 dans $P \setminus \{u_0, w_0\}$. Soit R un plus court chemin de u'_1 à v'_1 dans $P \setminus \{u_0, v_0, w_0\}$. Alors $V(R) \cup \{u_0, v_0, w_0\}$ induit une maison dans G , contradiction. Cela prouve (5.15).

Par (5.15) et à symétrie près, on peut supposer que $u_0 = c_1$, et donc v_0, w_0 sont des sommets de G . Comme ci-dessus, d'après le Lemme 5.28, pour $j = 2, \dots, m$, nous pouvons choisir un sommet q_j dans C_j voisin d'aucun $\{v_0, w_0\} \setminus N(c_j)$.

Montrons maintenant que :

(5.16) *Les sommets v_1 et w_1 de $G^\mathcal{Q}$ sont dans C .*

Supposons, à symétrie près, que v_1 ne soit pas dans C . Nous pouvons alors choisir un sommet $q'_1 \in C_1$ qui n'est pas adjacent à v_1 . Soit P le sous-graphe de G induit par $(V(K) \setminus C) \cup \{q'_1, q_2, \dots, q_m\}$. Soit R le plus court chemin de q'_1 à v_1 dans $P \setminus \{v_0, w_0\}$. Alors R a longueur au moins 2 et $V(R) \cup \{v_0, w_0\}$ induit une maison dans G , contradiction. Cela prouve (5.16).

Par (5.16), nous pouvons supposer que $v_1 = c_2$ et $w_1 = c_3$. Rappelons que q_2 est un sommet de C_2 qui n'est pas adjacent à w_0 et que q_3 est un sommet de C_3 qui n'est pas adjacent à v_0 . Puisque v_1 et w_1 ne sont pas voisins, les longueurs s, t de P_2, P_3 ne peuvent pas être toutes deux égales à 1. Supposons donc que $s \geq 2$. Nous montrons que :

(5.17) *Le sommet q_2 est adjacent à tout C_1 .*

Supposons que q_2 n'est pas adjacent à un sommet $q''_1 \in C_1$. Soit P le sous-graphe de G induit par $(V(K) \setminus C) \cup \{q''_1, q_2, \dots, q_m\}$. Soit R le plus court chemin de q''_1 à q_2 dans $P \setminus \{v_0, w_0\}$. Alors R a longueur au moins 2 et $V(R) \cup \{v_0, w_0\}$ induit une maison de G , contradiction. Ce qui prouve (5.17).

Maintenant, soit q_1 un sommet de C_1 , et soit P le sous-graphe de G induit par $(V(K) \setminus C) \cup \{q_1, \dots, q_m\}$. Nous montrons que

(5.18) *Tout voisin de q_2 dans $V(P) \setminus \{v_0, v_2\}$ est dans $\{q_1, \dots, q_m\}$.*

Soit x un voisin de q_2 in $V(P) \setminus \{v_0, v_2\}$. On peut supposer que $x \neq q_1$. Le sous-graphe $P \setminus \{q_1, q_2, v_2\}$ est connexe, donc il contient un plus court chemin X de v_0 à

x . Puisque v_0 n'a pas de voisins dans $V(P) \setminus \{q_1, q_2, w_0\}$, w_0 appartient à X (comme voisin de v_0). Maintenant, le Lemme 5.27 peut être appliqué à C_2 , X et v_2 , ce qui implique que x est adjacent à tout C_2 . Cela signifie que x est dans $\{q_1, \dots, q_m\}$ car sinon, v_1 et x seraient adjacents dans $G^{\mathcal{Q}}$. Cela prouve (5.18).

Dans $G^{\mathcal{Q}}$, marquons certains sommets de $K \setminus v_1$ de la manière suivante. Les sommets v_0 et v_2 sont marqués. Pour $j = 1, \dots, m$ et $j \neq 2$, le sommet c_j est marqué si et seulement si, dans G , le sommet q_2 est voisin de tous les sommets de la clique C_j correspondante de G . Tous les autres sommets de K sont inchangés. Appelons segment un sous-chemin de longueur au moins 1 de $K \setminus v_1$ dont les deux extrémités sont marquées et les sommets intérieurs sont non-marqués. Supposons qu'il existe un segment impair X de longueur ≥ 3 . Alors $V(X) \cup \{q_2\}$ induit un trou pair dans $G^{\mathcal{Q} \setminus \{C_2\}}$, ce qui contredit le Lemme 5.30. Donc tout segment a une longueur paire ou égale à 1. Notons que $V(P_1) \cup V(P_2) \setminus \{v_0, v_1\}$ induit un chemin induit, de longueur impaire (car r, s ont la même parité), et ses deux extrémités sont marquées; donc ce chemin contient un segment impair, qui, comme noté ci-dessus à longueur 1. Appelons y le voisin de v_2 sur ce chemin. Notons que nous avons soit $s \geq 3$ et $y = v_3$ soit $s = 2$ et $y = u_r$. Par (5.18) et le fait que les sommets de C sont deux-à-deux non-adjacents, le seul segment possible de longueur 1 est v_2-y . Donc y est marqué, et (5.18) implique $y \in C$. Supposons que $s \geq 3$. Alors $V(P_3) \cup V(P_2) \setminus \{v_1, v_2\}$ induit un chemin induit impair, dont les extrémités sont marquées, et donc il contient un segment de longueur 1. Le seul tel segment possible est v_0-w_0 , donc w_0 est marqué, et (5.18) implique $w_0 \in C$, ce qui contredit (5.15). Donc $s = 2$ et $y = u_r$. Maintenant, puisque B contient u_r il ne peut pas contenir un autre sommet de C , donc w_t n'est pas dans C , ce qui, par (5.16) implique que $t \geq 2$. La symétrie entre s et t est donc rétablie, et comme ci-dessus, on peut prouver que $t = 2$ et que q_3 est voisin de u_r . Mais alors q_2, v_0, w_0, q_3, u_r induit un trou impair ou une maison de G , contradiction. \square

Les Lemmes 5.29, 5.30 et 5.31 impliquent que $G^{\mathcal{Q}}$ est un graphe d'Artemis, ce qui prouve le Théorème 5.24.

5.5 Pour aller plus loin

Nous mentionnons ici quelques extensions possibles du travail présenté dans ce chapitre. Une piste majeure qui semble largement inexploitée concerne l'approximabilité des sous-problèmes de [ListCol]. Nous remarquons ensuite que la technique de contraction pour l'extension précolorée s'étend aux hypergraphes. Cependant, c'est le théorème d'union des matroïdes qui permet de résoudre les problèmes de ce chapitre dans le cas des matroïdes.

5.5.1 Approximabilité de [PrExt] et de [ListCol]

Nous mentionnons ici le champ de problèmes ouverts que constitue l'approximabilité des sous-problèmes de la coloration par liste, et donnons une 2-approximation élémentaire pour [PrExt] sur les graphes parfaits.

Les deux sous-problèmes principaux de [GOCCP], définis dans la Section 5.1.2, qui ne sont pas aussi des sous-problèmes de [ListCol] sont relativement bien étudiés quant aux questions d'approximabilité (voir [92] pour [MinColorSum] et [100] pour [OCCP]). Une raison en est peut être que ces problèmes ont été définis originellement comme des problèmes d'optimisation. Or les sous-problèmes de la coloration par liste admettent aussi une version optimisation naturelle : quelle est la cardinalité maximum d'un sous-graphe liste-colorable ? En vertu de la traduction de ces problèmes dans le langage de l'ordonnancement, donnée dans la Figure 5.3, le seuil d'approximabilité des problèmes de minimisation correspondants peut se révéler plus pertinent pour certaines applications. Nous laissons là cette double piste de recherche.

Considérons maintenant la version suivante de la minimisation de [PrExt] : étant donné une précoloration \mathcal{Q} de G , quel est le nombre minimum de couleurs nécessaires pour étendre \mathcal{Q} ? Ce problème est bien approximable lorsque que [Color] l'est aussi, comme le montre l'algorithme suivant.

L'algorithme Union :

Données : Un graphe G et une précoloration \mathcal{Q} de G .

Résultat : Une k -coloration de G qui étend \mathcal{Q} .

Utiliser $|\mathcal{Q}|$ couleurs pour colorier les sommets dans \mathcal{Q} . Colorier $G - V(\mathcal{Q})$ avec des nouvelles couleurs. La coloration de G est donnée par l'union de \mathcal{Q} et des nouvelles couleurs.

Théorème 5.32 *Pour toute classe de graphe \mathcal{G} fermée par passage au sous-graphe induit, $\rho([PrExt(\mathcal{G})]) \leq 1 + \rho([Color(\mathcal{G})])$. Cela est atteint par l'algorithme Union et $1 + \rho$ est une borne serrée pour la garantie de performance de l'algorithme Union.*

Preuve Soit ρ tel que [Color] soit ρ approximable sur \mathcal{G} . Soit $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ une k -coloration d'un graphe G telle que $k \leq \rho\chi(G)$, et \mathcal{Q} une précoloration de G . Quitte à les renommer, supposons que les couleurs de \mathcal{C} soient différentes de celles de \mathcal{Q} . La coloration donnée par \mathcal{Q} si $v \in \mathcal{Q}$ et par \mathcal{C} sinon est valide et utilise au plus $k + |\mathcal{Q}|$ couleurs. On a :

$$(5.19) \quad \max\{\chi(G), |\mathcal{Q}|\} \leq \chi(G/\mathcal{Q}) \leq k + |\mathcal{Q}| \leq \rho\chi(G) + |\mathcal{Q}|$$

$$(5.20) \quad \frac{k + |\mathcal{Q}|}{\chi(G/\mathcal{Q})} \leq \frac{\rho\chi(G) + |\mathcal{Q}|}{\max\{\chi(G), |\mathcal{Q}|\}} \leq \frac{(1 + \rho) \times \max\{\chi(G), |\mathcal{Q}|\}}{\max\{\chi(G), |\mathcal{Q}|\}}$$

On a donc bien une $1 + \rho$ approximation de $\chi(G/\mathcal{Q})$.

La borne est serrée comme le montre la famille de graphes $\{G_n\}_{n \in \mathbb{N}}$, tel que $\overline{G_n}$

est l'union disjointe de n arêtes et \mathcal{Q} consiste n pré-couleurs de taille 1 (un sommet pour chacune des arêtes de $\overline{G_n}$). L'algorithme Union donne une $2n$ -coloration qui étend \mathcal{Q} alors que $\chi(G/\mathcal{Q}) = n$. \square

Corollaire 5.33 (PrExt) *est 2-approximable sur les graphes parfaits.*

Il semble que l'approximabilité à différence bornée soit ouverte pour [PrExt] dans les line-graphes et même pour les LGBIP.

5.5.2 Extension de pré-coloration dans les hypergraphes

Nous observons que la technique de contraction se généralise à l'extension de pré-coloration dans les hypergraphes. Bien que cette technique semble inutile dans le cas des matroïdes, le théorème d'union des matroïdes permet de résoudre [GOCCP] lorsque l'hypergraphe à colorier a pour arêtes les circuits d'un matroïde.

Etant donnée une pré-coloration $\mathcal{Q} = \{C_1, \dots, C_m\}$ d'un hypergraphe \mathcal{H} , on définit l'hypergraphe \mathcal{H}/\mathcal{Q} dont les sommets sont obtenus à partir de $V(\mathcal{H})$ en contractant, pour chaque j , l'ensemble des sommets de C_j en un seul sommet c_j , les autres sommets restant inchangés. On met une (hyper-)arête (de taille 2) sur chaque paire (c_i, c_j) avec $1 \leq i, j \leq m$. Les arêtes contenues dans $V(\mathcal{H}) \setminus (C_1 \cup \dots \cup C_m)$ sont inchangées. Les arêtes qui intersectent au moins deux pré-couleurs sont effacées. Si A forme une arête de \mathcal{H} qui intersecte exactement un pré-couleur C_j , on forme une nouvelle arête $c_j + (A \setminus (C_j))$ dans \mathcal{H}/\mathcal{Q} . La démonstration du lemme suivant est directe.

Lemme 5.34 *Pour tout hypergraphe \mathcal{H} et toute pré-coloration \mathcal{Q} de \mathcal{H} , l'ensemble des colorations qui étendent \mathcal{Q} est en bijection avec les colorations de \mathcal{H}/\mathcal{Q} . En particulier, le nombre minimum de couleurs qu'il faut pour étendre \mathcal{Q} est égal à $\chi(\mathcal{H}/\mathcal{Q})$. \square*

Cependant, nous ne connaissons aucune classe d'hypergraphes pour lesquels le Lemme 5.34 donne un résultat de complexité intéressant (en dehors des compléments de graphes de Meyniel). Par exemple, on perd la propriété d'être un matroïde en utilisant la contraction :

Exemple 5.35 *Soit $\mathcal{H} = \{1, 2, 3, 4\}, \mathcal{E}$ l'hypergraphe qui a comme arêtes, tous les ensembles de taille 3 (\mathcal{H} est l'ensemble des circuits du matroïde \mathcal{U}_4^2). Considérons la pré-coloration $\mathcal{Q} = \{\{1, 2\}\} = \{C_1\}$ de \mathcal{H} . Le clutter des minimaux de l'hypergraphe contracté*

$$(\mathcal{H}/\mathcal{Q})^{\min} = (\{c_1, 3, 4\}, \{\{c_1, 3\}, \{c_1, 4\}\})$$

ne définit pas les circuits d'un matroïde (car isomorphe au graphe P_3)

Cependant le théorème d'union des matroïdes s'adapte très bien à la pré-coloration et même à la coloration par liste (et en fait, encore plus généralement, à [GOCCP]).

Proposition 5.36 *Soit $\mathcal{M} = (V, \mathcal{I})$ un matroïde (donné par un oracle d'indépendance) et $L : V \rightarrow \mathcal{P}([1..k])$ une liste de couleurs admissibles pour chaque sommet. On peut liste-colorier un nombre maximum de sommets en temps polynomial.*

Preuve Pour chaque $i \in [1..k]$, $L^{-1}(i)$ est un matroïde \mathcal{M}_i (la restriction de \mathcal{M} à $L^{-1}(i)$). D'après le théorème d'union des matroïdes, l'union des \mathcal{M}_i est encore un matroïde, et on peut en trouver un indépendant maximum en temps polynomial. \square

Proposition 5.37 *Soit $\mathcal{M} = (V, \mathcal{I})$ un matroïde (donné par un oracle d'indépendance) et $f : V \times V \rightarrow \mathbb{N}$ une fonction de coûts pour [GOCCP]. On peut calculer en temps polynomial une partition de coût minimum en indépendants.*

Preuve Soit B un majorant des valeurs de f . Pour chaque couple $(v, t) \in V \times V$ on associe une nouvelle valeur $f'(v, t) := B - f(v, t)$. Grâce au Théorème d'union des matroïdes, une partition de coût minimum revient à trouver un indépendant de poids maximum dans l'union des V copies pondérées de \mathcal{M} . \square

Conclusion

Dans cette conclusion, je reviens d'abord sur le problème [Pintro] discuté en introduction. Pour cela, je présente un problème plus général [Pconclu] en faisant ressortir la complémentarité des résultats positifs et négatifs de complexité pour des sous-problèmes de [Pconclu]. Je présente ensuite le sentiment qui domine dans l'évaluation de mon travail et de la communauté à la vie de laquelle j'ai participé. Après 4 années de doctorat, je suis de plus en plus convaincu que la simplicité de l'accès aux résultats issus de la recherche en mathématiques discrètes peut et doit être améliorée.

En mettant en œuvre l'artillerie combinée de la combinatoire polyédrale et de la théorie de la complexité algorithmique, un troisième paradigme s'est révélé nécessaire : la classification. En effet, l'étude des nombreux problèmes discutés dans ce mémoire a fait apparaître un manque de cohérence et donc de visibilité des problématiques et résultats issus de l'ordonnement chromatique. D'une part, certains résultats sont redécouverts à plusieurs décennies d'intervalle. D'autre part, les mêmes problématiques théoriques apparaissent dans des contextes appliqués différents. Cela motive la mise en œuvre d'une terminologie commune. Dans ce sens, le principal apport de ce mémoire est l'identification de quelques bases conceptuelles utiles en vue d'une classification méthodique des résultats d'ordonnement chromatique. En particulier deux concepts généraux ont émergé :

- i) Les contraintes d'(in)compatibilité définies par des hypergraphes,
- ii) Les coûts d'opération définis par des fonctions d'ensemble.

La présentation des problèmes selon ces concepts a deux avantages. D'une part, cela augmente sensiblement la versatilité des modèles d'ordonnement par batchs. D'autre part, cela permet de concevoir de nombreux problèmes comme des cas particuliers d'un même problème général, et donc d'exhiber et de tirer parti des réductions entre ces problèmes. La contrepartie de la généralité de cette présentation est qu'elle nécessite d'être étudiés dans des modèles de calcul à oracles (pour pouvoir leur donner sens dans le cadre de la complexité algorithmique)⁵.

⁵Les questions liés aux matroïdes et aux fonctions sous-modulaires sont déjà discutées dans ce cadre depuis longtemps. Faut-il, par nos recherches, réaliser la prophétie de Yann Kieffer, qui

Pourquoi une telle soif de classification des résultats ? C'est parce que si l'on en croît les hypothèses classiques de la théorie de la complexité ($P \neq NP$), la profusion des résultats de complexité pour des cas spéciaux de problèmes classiques reflète la richesse intrinsèque du monde des problèmes. Or le théorème de Cook a 35 ans, et le nombre de résultats de complexité depuis lors nécessite des méthodes adaptées pour traiter efficacement et de manière cohérente cet ensemble de résultats.

Je tente d'illustrer cette richesse intrinsèque du monde des problèmes dans la Figure 5.9. Celle-ci donne, de plus, une vue schématique de la complémentarité des résultats obtenus grâce à l'approche polyédrale et la théorie de la complexité pour certains cas spéciaux du problème suivant :

[Pconclu] Partition de coût minimum sous-contraintes

Données : Un graphe $G = (V, E)$, un hypergraphe héréditaire (V, \mathcal{E}) donné par un oracle d'appartenance à \mathcal{E} , un temps d'opération $f : \mathcal{P}(V) \rightarrow \mathbb{R}_+$ donné par un oracle de valeurs.

Résultat : Une partition en cliques $\{C_1, C_2, \dots, C_k\}$ de G telle que $C_i \in \mathcal{E}$ pour tout i .

Mesure : $\sum_{i=1}^k (f(C_i))$ à minimiser.

Dans la première table, le coût f est unitaire (i.e. $f(U) = 1$ pour tout $U \subseteq V$) Dans la deuxième table, le système d'indépendance est libre. Dans la troisième table, le graphe de compatibilité est complet. **P** : il existe un algorithme polynomial. Un diamant signale l'existence d'une formule min-max. **NPC** : NP-complet. ρ : seuil d'approximabilité en temps polynomial.

[a] Théorème d'union des matroïdes

[b] Coloration des graphes parfaits

[c] Les inégalités de rang forment un système TDI

[d] Section 3.2 pour l'étude de la formule min-max et Section 3.3.3 pour l'algorithme et la preuve de la formule.

[e] et **[f]** Voir Section 3.4.2

[g] Voir Section 4.3.2

[h] Max-coloring est fortement NP-complet sur les split graphes [65, 27].

[i] Max-coloring est $4c$ -approximable quand le problème de la coloration est c -approximable [127].

[j] $\rho^* = 8/7$ est le seuil d'approximation de max-coloring sur les graphes bipartis [70, 127, 65].

annonce que les oracles sont le cadre de la théorie de la complexité de demain ? C'est le sujet d'un autre débat.

système d'indépendance	Matroïde Intersection	?	?	?	$\rho \geq 4/3$
	Matroïde	P [a]	?	?	$\rho \geq 4/3$
	Partition Généralisée	P	?	?	$\rho \geq 4/3$
	Uniforme ($b \leq n$)	P	P	P [d]	$2 \geq \rho \geq 4/3$ [e] [f]
	Libre ($b = +\infty$)	P	P	P	P [b]
		Complet	Intervalle	Triangulé	Parfait
Graphe de compatibilité					

Coût d'un batch	Polymatroïdal	P [c]	NPC	NPC	$\rho \geq 8/7$
	Steiner TSP sur un arbre	P	$2 \geq \rho \geq \text{NPC}$ [k] [l]	NPC	$\rho \geq 8/7$
	Max-batch	P	P [g]	NPC [h]	$4 \geq \rho \geq 8/7$ [i] [j]
	Unitaire	P	P	P	P [b]
		Complet	Intervalle	Triangulé	Parfait
Graphe de compatibilité					

Coût d'un batch	Polymatroïdal	P [c]	NPC	NPC	NPC
	Steiner TSP sur un arbre	P	NPC [n]	NPC	NPC
	Max-batch	P	P	P [m]	?
	Unitaire	P	P	P	P [a]
		Libre ($b = +\infty$)	Uniforme ($b \leq n$)	Partition Généralisée	Matroïde
Système d'indépendance					

Figure 5.9 – Complexité de trois sous-problèmes de [Pconclu].

[k] et [l] Voir [14].

[m]

[n] Proposition 4.26 page 213.

Apport philosophique. Dans ce mémoire, nous avons discuté de problèmes d'ordonnement chromatique. Nous avons illustré la complémentarité des angles de vue qu'offrent les disciplines suivantes :

- i) les classes de structures combinatoires [29, 28, 122],
- ii) l'ordonnement [32, 34],
- iii) la combinatoire polyédrale [129],
- iv) la théorie de la complexité algorithmique [80, 1],
- v) l'approximation polynomiale [8, 51].

Cette complémentarité laisse présager d'ouvrages de synthèses faisant ressortir les liens entre ces disciplines, car de plus en plus de publications font états de résultats issus du métissage de ces techniques. Aujourd'hui cependant, la maîtrise de chacun de ces paradigmes passe par une connaissance encyclopédique de nombreux résultats : s'orienter dans la littérature d'un de ces domaines est trop fastidieux pour que des non-spécialistes puissent profiter des résultats que l'on trouve dans les publications. D'un autre côté, les états de l'art permettent souvent un gain de temps substantiel pour le néophyte. Or une partie importante du travail de synthèse consiste à recenser et à classer les résultats. Il m'est apparu progressivement qu'une partie de ce travail de classement peut être fait de manière automatique. Cela permet non seulement de faire gagner du temps dans le travail de synthèse, mais aussi d'améliorer la qualité de ces synthèses. Je discute cela plus en détails dans l'Annexe A, en soulignant quelques idées simples et générales. Ces idées se résument principalement à utiliser "la méthode des posets pour les bases de données". La simplicité, la versatilité et la puissance de cette méthode me confèrent une motivation particulière pour participer à son développement et à sa mise en application dans les années à venir.

Annexe A

Classifications, SGBD et posets

Dans cette annexe, nous évoquons quelques utilisations des ensembles partiellement ordonnés (posets) dans les systèmes de gestion de base de données (SGBD). Cela permet une amélioration phénoménale de l'accès aux informations répertoriées dans cette base, lorsque ces informations sont naturellement munies de relations de comparaisons. Cette méthode est déjà implicitement et partiellement utilisée avec succès [34, 28, 51]. Elle se révèle d'une aide précieuse lors de la classification et la synthèse des résultats sur les structures combinatoires [28] et des résultats qui concernent l'optimisation [34, 28, 51]. Le but de cette annexe est de donner un aperçu de ces potentialités qu'offrent quelques concepts simples, mais qui ensemble, forment une boîte à outils cohérente et puissante. Je tenais à participer à la diffusion des idées contenues dans cette boîte à outils, car je souhaite participer à son développement dans les années à venir. Nous présentons d'abord les motivations et bases théoriques sur lesquelles s'appuient cette boîte à outils (Section A.1). Nous donnons ensuite un aperçu du type de synthèses qui peuvent être faites de manière graphique, automatisée et interactive grâce aux Figures des Sections A.2 et A.3 (cette fonctionnalité est déjà en partie mise en œuvre dans [28]).

A.1 De l'importance des posets dans les synthèses

Comme de nombreuses disciplines scientifiques "vivantes", les théories de l'ordonnement et de la coloration sont soumises à un phénomène de ramification des problématiques. Bien que certains résultats profonds et généraux simplifient parfois le corpus des résultats, l'ensemble des résultats pertinents (*i.e.* que l'on ne sait pas déduire d'autres résultats plus généraux) est en perpétuelle croissance.

Les synthèses de résultats font l'objet d'une forte demande de la part de la communauté scientifique. Cela est illustré par le succès des ouvrages de Garey et Johnson [80] pour la complexité des problèmes d'optimisation combinatoire, de Aaron-

son et Kuperberg [1] pour les classes de complexité, de T'Kindt et al. [137], Brucker et Knust [32, 34] pour l'ordonnancement, de Ausiello et al. [8, 51] pour l'approximabilité, de Schrijver [129, 128] pour la combinatoire polyédrale et la PLNE, de Brandstädt et al. [29, 28] pour les classes de graphes ou encore de Oxley pour les classes de matroïdes [122]. Les ouvrages mentionnés ci-dessus tentent d'apporter de la cohérence et de l'ordre au grand nombre de résultats dont ils traitent. Mais ils ne sont pas seulement utiles pour le domaine de spécialité dont ils traitent. Certains paradigmes utilisés pour effectuer la synthèse et la présentation des résultats se dégagent de ces ouvrages et peuvent être transposés à d'autres disciplines.

C'est au travers de cette vision transdisciplinaire de la découverte que nous avons tenté d'adapter les paradigmes de Brucker et Knust [32, 34]¹ pour formuler et appliquer la thèse suivante :

Thèse 1 : *L'utilisation d'un cadre général permettant d'exprimer les problèmes de coloration comme des sous-problèmes dans une notation symbolique unifiée (type $\alpha|\beta|\gamma$) permet d'effectuer la synthèse des résultats concernant les problèmes de coloration.*

La cohérence d'une telle synthèse favorise l'accessibilité et le développement des résultats (aussi bien de manière spécialisée que transdisciplinaire). Autrement dit, la mise en place d'une telle notation améliore l'applicabilité des résultats ainsi que leur potentiel à servir de socle pour des conjectures et théorèmes ultérieurs, comme nous le discutons dans la Section A.1.1. Alors que nous tirions les bénéfices de ce paradigme pour l'étude des problèmes d'ordonnancement chromatique, un autre paradigme, à la fois plus simple et plus général s'est naturellement dégagé : la méthode des posets pour les bases de données. Nous résumons cette méthode dans les deux thèses suivantes :

Thèse 2 : *Une myriade de théorèmes peut être exprimée efficacement comme les éléments d'un poset produit.*

Le but principal de la Section A.1.2 est d'expliquer en quoi les chercheurs expriment spontanément de nombreux théorèmes de cette manière. En pratique, l'intérêt de cette deuxième thèse est décuplé par la mise en œuvre de l'artillerie automatique des bases de données combinée avec les potentialités ergonomiques qu'offrent les algorithmes élémentaires pour manipuler les posets.

Thèse 3 : *L'utilisation combinée des posets et des bases de données permet des opérations simples et automatiques qui améliorent considérablement l'efficacité de l'accès aux informations (lorsque celles-ci sont "naturellement" munies de relations de comparaisons).*

Certaines opérations sur les posets particulièrement utiles sont discutées dans la Section A.1.3.

¹Basé sur des publications de Graham et al qui remontent à la fin des années 70.

A.1.1 Thèse 1 : La notation $\alpha|\beta|\gamma$ comme un pont entre la coloration et l'ordonnement

Cette première thèse est introduite plus précisément dans la Section 1.5 et est illustrée sobrement dans la deuxième partie de ce mémoire. Sa pertinence et sa réalisabilité s'appuient sur deux points :

Premièrement, de nombreuses ramifications sont étudiées parallèlement, voire de manière conjointe entre la coloration et l'ordonnement. Cela motive l'utilisation d'un langage et d'une notation commune.

Deuxièmement, la notation $\alpha|\beta|\gamma$ a fait ses preuves quant à sa capacité à structurer et à synthétiser une large palette de résultats de complexité. Or de nombreux résultats concernant les problèmes d'ordonnement chromatique sont redécouverts à plusieurs années d'intervalle. Il nous semble donc important d'améliorer l'accessibilité de ces résultats.

Dans la deuxième partie de ce mémoire, certains ponts sont construits entre ces deux disciplines, montrant que certains problèmes s'expriment indifféremment dans ces deux langages, pour peu que l'on dispose du bon dictionnaire. Nous tentons de donner les bases d'un tel dictionnaire dans la Section 1.5. A un niveau plus concret, la Section 5.1 (et particulièrement la Figure 5.3 page 221) donnent une bonne illustration de l'équivalence et de l'interfécondation des questionnements issus de la coloration et de l'ordonnement.

A.1.2 Thèse 2 : Exprimer l'information par les éléments d'un produit de posets

La plupart des théorèmes d'optimisation combinatoire sont exprimés dans une forme paramétrique, qui consiste à "croiser" les concepts spécifiques de certains paradigmes².

Nous donnons des exemples de posets qui nous intéressent particulièrement (Section A.1.2.1).

L'intérêt de cette deuxième thèse nous semble difficile à argumenter autrement que par l'exemple. A cette fin, nous présentons une synthèse de résultats concernant la coloration bornée, issus du Chapitre 3 (Section A.1.2.2).

²Les formalisme des problèmes, la complexité algorithmique et la combinatoire polyédrale sont trois exemples de paradigmes, présentés ci-dessous.

A.1.2.1 La paramétrisation standard des théorèmes

La décomposition en posets présentée ici est en partie arbitraire. Elle reflète toutefois la manière dont la plupart d'entre nous classons les théorèmes dans nos représentations personnelles, ainsi que la manière dont nous les communiquons. Un éventuel choix de posets plus "naturels", ou plus "élémentaires" nécessiterait probablement une longue réflexion et certainement une plus grande culture et une plus grande expérience que la mienne.

- Π_1 Classes de problèmes (Par exemple, si l'on souhaite se restreindre aux problèmes d'ordonnement, on choisit en général la décomposition en sous-posets selon $\alpha | \beta | \gamma$).
- Π_2 Classes de complexité "positive", c'est-à-dire existence d'un algorithme pour résoudre un problème donné (on peut séparer les propriétés d'un algorithme selon son efficacité (*i.e.* temps de calcul), la qualité de ses solutions (*i.e.* garantie d'approximation, probabilité de réussite), mais aussi le modèle de calcul dans lequel l'algorithme est formulé (RAM-séquentielle, parallèle, non-déterministe, quantique...)). La séparation entre ces critères semble à la fois naturelle, aisée en pratique et permet de représenter des dimensions complémentaires de l'analyse des algorithmes.
- Π_2' Classes de complétude ou de "complexité négative"³. L'étude des complexités positives et négatives ont l'étude des réductions en commun⁴.
- Π_3 Classes de graphes (que l'on peut étendre aux hypergraphes voire aux fonctions d'ensembles, ou encore restreindre aux classes contraintes de précédence (*i.e.* de posets)). L'étude des classes de graphes est liée à celle des posets des classes d'opérations qui préservent l'appartenance à certaines classes de graphes. Deux posets fondamentaux de classes d'opérations semblent se distinguer, selon que ces opérations interviennent dans des caractérisations par graphes interdits ou dans des théorèmes de (dé-)compositions (voir Sections 2.1.3 et 2.1.4).
- Π_4 Classes de résultats polyédraux. Etant donné un problème de Π_1 et une formulation pour ce problème (un ensemble de variables avec lequel on formule le problème), on exprime souvent un résultat de combinatoire polyédrale grâce à deux paramètres : l'un décrit l'ensemble des contraintes considérées⁵ et l'autre décrit les propriétés d'intégralité du programme ainsi formé⁶. Bien que la structuration des classes de formulations et des classes de contraintes ne semble pas chose aisée, le poset des propriétés d'intégralités est à la fois riche en relations et homogène dans sa terminologie.

³Bien que les résultats de complétude soient en général discutés de manière parallèle aux résultats positifs, ces deux types de résultats semblent indépendants.

⁴Les réductions servent indifféremment pour les résultats positifs et négatifs. De plus, on peut montrer une réduction entre deux problèmes sans connaître ni résultat positif ni résultat négatif pour aucun de ces deux problèmes.

⁵Pour la formulation avec des variables sur les sommets du stable de poids maximum, on considère les inégalités d'arêtes, de cliques, de cycle impair, de roues...

⁶matrice TTU/équilibrée, polyèdre entier/IDP/MIRUP/rang de Chvátal 1, système TDI...

Π_5 Classes de validation sociale⁷ (pour un papier ou un théorème, on peut considérer sa langue, ses dates associées (pour trouver le dernier état de l'art sur un sujet, ou encore créditer le premier auteur d'un résultat pour une découverte), son statut (validé par Pr.Dr. untel ou par un autre, publié ici ou là, accessibilité libre ou payante...)).

La pertinence et la portée de cette deuxième thèse semble donc relativement large. La mise en pratique d'une base de donnée utilisant les idées de la troisième thèse et l'utilisation de cette deuxième thèse pour d'autres types d'informations seraient nécessaire pour pouvoir les juger.

Notons que la présentation des théorèmes sous cette forme paramétrique pose un double problème : le choix des formulations générales et la décomposition en posets élémentaires. En effet, bien que certains aspects puissent être gérés de manière souple et dynamique (voir thèse 3), le format d'entrée dans la base de données semble nous contraindre à ces deux choix de manière statique. Le choix d'une formulation générale consiste à fixer, a priori, la forme des théorèmes que l'on souhaite exprimer. Le poset suggéré ci-dessus pour lequel cette question se pose de manière fondamentale est le poset Π_1 des problèmes (les autres jouissent d'une certaine uniformité culturelle qui rend les choix plus ou moins naturels). Le poset des problèmes est à la fois le plus grand⁸ et celui qui traduit la plus grande diversité culturelle⁹. L'enjeu à relever pour assurer la pérennité de cette deuxième thèse est illustré par le poset des problèmes : il s'agit de choisir une formulation générale qui jouisse de qualités a priori contradictoires.

- i) Simplicité (*i.e.* la définition du problème général doit être concise).
- ii) Large applicabilité et potentiel structurant. Le problème général permet d'exprimer de nombreux sous-problèmes simplement en restreignant les valeurs des paramètres qui définissent le problème général.
- iii) Traductibilité avec les cultures dominantes. Pouvoir traduire une information générale dans un cadre et un langage associé à un domaine de recherche particulier assure l'accessibilité de l'information et favorise la transdisciplinarité¹⁰.
- iv) Souplesse évolutive. C'est la grande inconnue liée au choix d'un formalisme : combien de temps sera-t-il adapté pour exprimer les résultats ultérieurs ?

⁷Ces exemples illustrent que les posets généralisent à la fois les ensembles et les données numériques. Autrement dit, les données que l'on manipule habituellement sont souvent munies d'une relation d'ordre vide ou complète.

⁸On peut considérer que le poset des classes de graphes est un sous-poset de celui des problèmes.

⁹De nombreux problèmes équivalents sont discutés avec des terminologies différentes selon qu'ils sont présentés dans le langage de l'ordonnancement, de la coloration de graphes, de la théorie des nombres...

¹⁰L'échec de l'Esperanto laisse de quoi méditer sur le coût à investir pour le succès d'un nouveau langage qui n'apporte pas de bénéfices nouveaux.

A.1.2.2 Une étude de cas illustrant la richesse des liens entre les différents théorèmes

Exemple A.1 *Le problème de la coloration bornée est un problème NP, qui généralise la coloration usuelle. A ce titre, il n'est pas approximable à rapport moins que $V^{1-\epsilon}$ pour les graphes généraux. Il admet cependant une 2-approximation dans les graphes parfaits. De plus, il est polynomial dans les graphes co-triangulés. Par ailleurs, c'est un problème polynomialement équivalent à la coloration usuelle dans toute sous-classe de graphes sans- $K_{1,3}$. D'un autre côté, c'est un cas spécial de la plupart des problèmes d'ordonnement sur machines parallèles avec contraintes d'incompatibilité...*

L'Exemple A.1 utilise implicitement la méthodologie des posets : on n'y parle pas de la polynomialité de la coloration bornée pour les compléments de graphes d'intervalles car cela découle du cas des graphes co-triangulés... pour peu que l'on connaisse la relation d'inclusion entre ces deux classes de graphes. Notons qu'en faisant ce choix de présentation, on perd le fait que l'algorithme pour les graphes d'intervalles est beaucoup plus efficace que l'algorithme connu pour les graphes triangulés. Des informations contenues dans l'Exemple A.1, on peut déduire automatiquement (avec la méthode des posets) que le problème de la coloration bornée est polynomial dans les line-graphes de bipartis¹¹. En effet, les line-graphes de bipartis sont des graphes à la fois parfaits et sans- $K_{1,3}$. Combiné avec la polynomialité de la coloration sur les graphes parfaits et l'équivalence entre coloration et coloration bornée sur les graphes sans- $K_{1,3}$, cela donne le résultat annoncé. De même, on a une approximation à différence au plus 1 dans les line-graphes, en utilisant les résultats de Vizing sur la coloration minimum des line-graphes.

Au travers de ce récit, nous espérons faire sentir la puissance de la méthode des posets. Nous invitons le lecteur sceptique à méditer sur le fait suivant. Frédéric Gardi a consacré sa thèse à l'étude de la coloration bornée [79]. Il y laisse la complexité de ce problème ouverte sur les graphes co-triangulés. Mais dans cette même thèse, il mentionne la polynomialité du couplage multidimensionnel parfait sur les graphes triangulés. Or, on montre aisément que le couplage b -dimensionnel parfait est polynomialement équivalent à la partition en b -cliques sur toute classe de graphe fermée par ajout de sommets universels (ce qui est le cas des graphes triangulés). Autrement dit, ce qui m'est apparu pendant quelques mois comme un problème ouvert intéressant pouvait être résolu en utilisant un résultat connu et une réduction élémentaire. De là à conclure que les mathématiques peuvent être faites par les ordinateurs, il y a un gouffre que je ne m'aviserai pas de franchir. En revanche, je suis persuadé que la découverte en mathématique n'est pas différente de la découverte dans les autres sciences : elle résulte en grande partie de la communication

¹¹Ce résultat a des applications importantes pour les problèmes d'emploi du temps.

et du nomadisme¹². On peut penser que Frédéric Gardi et moi-même sommes de mauvais étudiants puisqu'il a fallu si longtemps pour faire cete observation élémentaire. Mais on peut aussi interpréter les choses autrement : la "découverte" ne consiste-t-elle pas simplement en une formulation mentale des évidences cachées ? Bien sûr, on ne peut pas prédire la forme des évidences que vont formuler les génies. Mais pour la plupart des chercheurs, qui, comme moi, passent leur temps à raffiner des paradigmes établis, on peut prédire une bonne partie des résultats qui vont émerger. Et on peut donc utiliser l'outil informatique pour structurer cette connaissance émergente et en faciliter la diffusion.

A.1.3 Thèse 3 : La méthode des posets pour les bases de données

La méthode des posets dans une base de données consiste à permettre aux utilisateurs qui consultent la base d'utiliser les relations connues entre les données pour rendre l'accès à l'information plus efficace. Des illustrations de ce paradigme sont :

- Le calcul du diagramme de Hasse d'une relation d'ordre : lors d'un affichage graphique, enlever les relations qui sont conséquences des autres améliore la lisibilité.
- Le calcul des éléments extrêmes d'un poset : c'est ce qui est fait pour améliorer la lisibilité en réduisant le nombre de résultats accessibles dans [34] (maximaux polynomiaux, minimaux NP-durs) ainsi que dans [51] (bornes inférieures et supérieures sur le seuil d'approximabilité).
- La restriction des champs à l'intersection d'idéaux élémentaires¹³ et de compléments d'idéaux élémentaires : cela permet de réduire efficacement l'espace de recherche lors d'une consultation de la base, par un procédé de recherche dichotomique généralisé. C'est une méthode que l'on peut utiliser lors de la recherche de la "frontière de complexité d'un problème", ainsi que pour déterminer la classe des graphes qui caractérise une propriété donnée.
- La contraction d'"intervalles de valeurs"¹⁴ : cela permet de choisir le degré de détail avec lequel l'information est affichée. Par exemple, on contracte souvent les classes de complexité $\mathcal{O}(n \log(n))$, $\mathcal{O}(n^2)$, $\mathcal{O}(n^4)$ en polynomial et $\mathcal{O}(1.1^n)$, $\mathcal{O}(2^n)$ en exponentiel.

¹²Je réfère aux travaux de Michel Soutif pour une discussion historique du rôle, dans la découverte scientifique, de la communication au delà des frontières établies.

¹³Un idéal élémentaire inférieur (respectivement supérieur) d'un poset est l'ensemble des éléments plus petits (respectivement plus grands) qu'un élément donné.

¹⁴Un intervalle de valeurs est un sous-ensemble I d'un poset tel que si $a, c \in I$ et $a \leq b \leq c$, alors $b \in I$.

Le produit de posets étant encore un poset, les outils permettant de manipuler l'information d'un poset peuvent être définis pour un unique poset¹⁵. Par ailleurs, il semble que l'algèbre relationnelle soit une enveloppe théorique suffisante pour discuter les bases de données sur des posets (les relations dans un poset peuvent être codées dans une table). Autrement dit, une base de donnée utilisant la méthode des posets devrait pouvoir s'appuyer fortement sur les technologies existantes.

Dans la deuxième thèse, nous avons évoqué plusieurs qualités que devraient avoir les posets dans une base de donnée. Il semble au moins que l'utilisation de bases de données améliore grandement la *Traductibilité*. En effet, certaines traductions peuvent être mises dans la base rendant le processus de traduction automatique¹⁶. De plus, l'utilisation du langage permettant le plus grand niveau de détail ne semble pas nécessaire, ni pour l'entrée, ni pour l'extraction des données dans la base. Pour cela il convient d'identifier les concepts importants (comme "polynomial" pour la complexité), et de les étiqueter dans la base comme des contractions de concepts plus fins.

A.1.4 Conclusion sur la méthode des posets

De nombreux résultats peuvent être représentés comme les éléments d'un poset produit. Si ces résultats sont rentrés dans une base de données avec un grand nombre de détails, cela n'entrave en rien la simplicité de la consultation : dans la plupart des moteurs de recherche et dans la plupart des bases de données, c'est justement en restreignant les valeurs possibles de certains champs que l'on obtient une réponse concise à une question ciblée et cela indépendamment du nombre d'informations accessibles par ailleurs. L'intérêt de l'utilisation des bases de données en général et de la méthode des posets en particulier est évidente quand on lit la note descriptive de [51] ("paper describing how the compendium is used") dans laquelle les auteurs expliquent les choix préalables au travail d'archiviste¹⁷. En effet, le choix statique de la présentation oblige à passer sous silence la complexité exacte des algorithmes recensés (linéaire ou cubique), ce qui peut être un inconvénient majeur de son applicabilité. De plus, les notes en bas de pages dans [51] pour distinguer l'approximabilité des sous-problèmes (selon les classes de graphes par exemple), montrent clairement que le choix statique de la présentation n'est pas adapté pour rendre compte de telles ramifications.

¹⁵Bien que la place en mémoire de l'information et l'efficacité des algorithmes dépende de la décomposition en produit.

¹⁶Des exemples de traductions sont données dans la Figure 5.3 page 221.

¹⁷*"In 1993 the two authors of this paper started to collect approximation results. We soon noticed that the amount of approximability results was enormous, and therefore we had to limit ourselves to considering polynomial time approximation results for NP optimization problems that are not solvable optimally in polynomial time ... We of course collected the best upper and lower bounds of approximation, but we did not try to find the fastest approximation algorithm giving the best approximation upper bound - any polynomial time algorithm sufficed..."*

A.2 Les structures combinatoires utilisées

Dans cette annexe, nous présentons quelques inclusions pour :

- Certaines classes de fonctions d'ensemble, qui interviennent comme fonctions de coûts (ou temps d'opération) en ordonnancement par batchs (Figure A.1)
- Certaines classes de graphes parfaits (Figure A.2)

Pour améliorer la lisibilité nous ne présentons que le diagramme de Hasse des inclusions considérées.

Pour une classe de graphes \mathcal{G} , on note $\overline{\mathcal{G}} := \{G \mid \overline{G} \in \mathcal{G}\}$, la classe des graphes complémentaires de graphes de \mathcal{G} . Pour toute paire $(\mathcal{G}_1, \mathcal{G}_2)$ de classes de graphes, on a

- $\mathcal{G}_1 \supseteq \mathcal{G}_2 \iff \overline{\mathcal{G}_1} \supseteq \overline{\mathcal{G}_2}$ et ces inclusions sont indiquées par $\mathcal{G}_1 \rightarrow \mathcal{G}_2$.
- $\mathcal{G}_1 \supseteq \overline{\mathcal{G}_2} \iff \overline{\mathcal{G}_1} \supseteq \mathcal{G}_2$ et ces inclusions sont indiquées par $\mathcal{G}_1 \Rightarrow \mathcal{G}_2$.

Les classes de graphes auto-complémentaires (*i.e.* telles que $\mathcal{G} = \overline{\mathcal{G}}$) sont entourées d'un ovale épais. Lorsque $\mathcal{G}_1 \supseteq \mathcal{G}_2$ et $\mathcal{G}_1 \supseteq \overline{\mathcal{G}_2}$ on le note par une flèche épaisse, sauf dans le cas où l'une des deux classes est auto-complémentaire, car la double inclusion vient automatiquement avec l'une des inclusions. Finalement, les relations d'inclusion en traits discontinus sont des conjectures, ou pour le moins, des inclusions pour lesquelles nous ne connaissons ni preuve, ni contre-exemple.

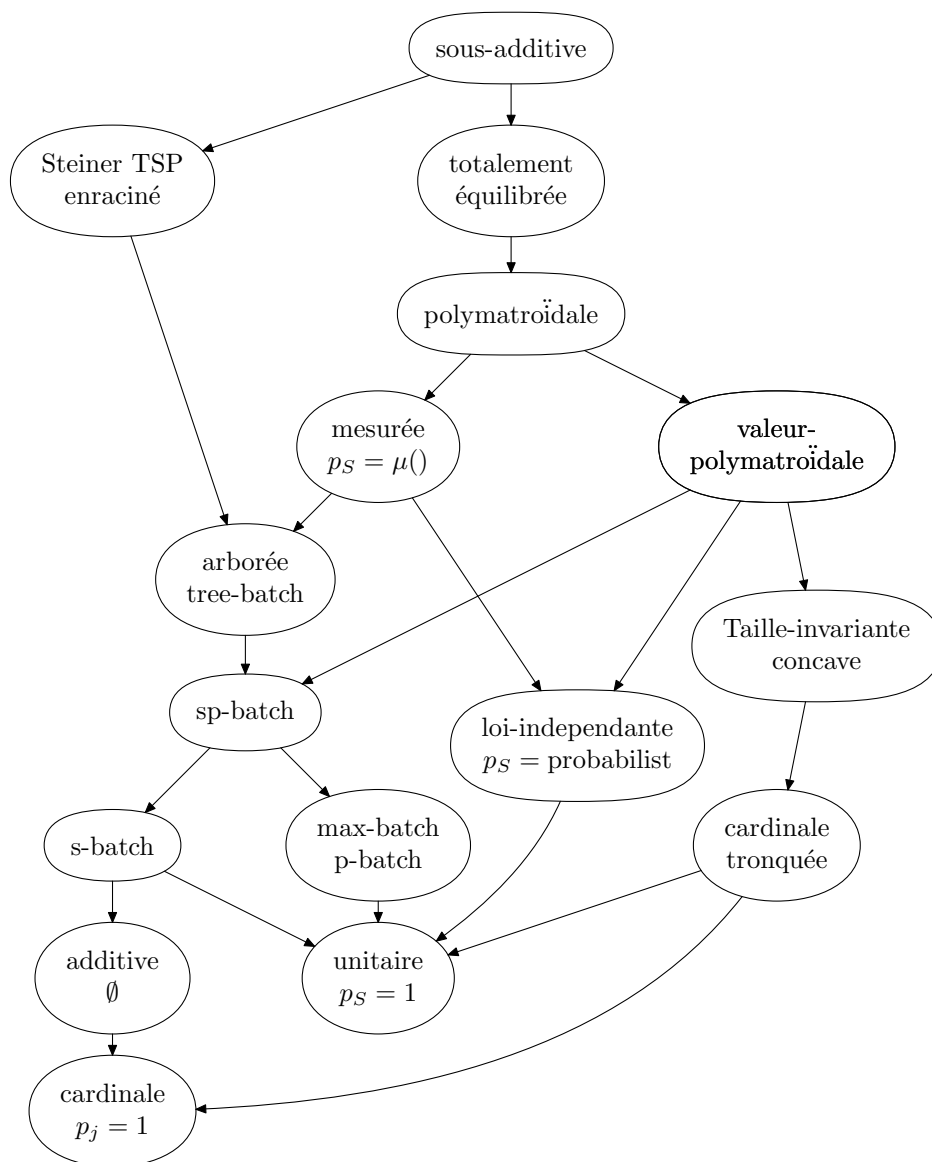


Figure A.1 – Quelques classes de fonctions d'ensembles

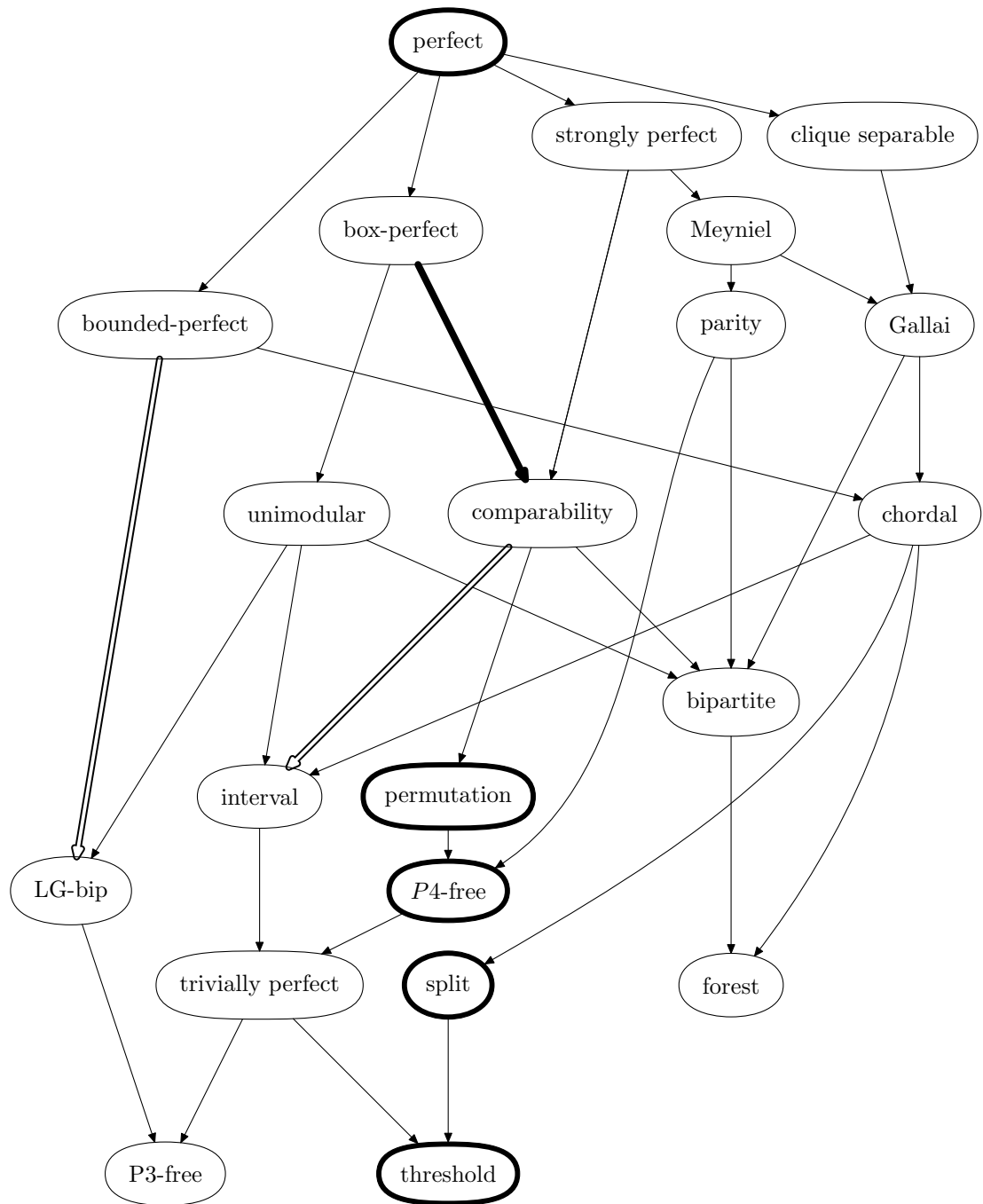


Figure A.2 – Quelques classes de graphes parfaits

A.3 Petite synthèse pour [ColB] et [PrExt]

Dans cette section, nous présentons quelques résultats de complexité pour les problèmes [ColB] et [PrExt]. Nous présentons des résultats pour [ColB] dans un format inspiré de la notation $\alpha | \beta | \gamma$, comme suggéré dans les Sections 1.5.2. Nous illustrons la simplicité de la mise en pratique de la Thèse développée dans la Section A.1.2, en y incluant quelques résultats d'approximation pour le problème [ColB].

Nous présentons ensuite quelques résultats de complexité pour [ColB] et [PrExt] selon un format graphique proche de celui utilisé dans [28]. Cela illustre la terrible puissance communicative des outils graphiques dont on dispose pour représenter l'information des posets et donne du crédit à la thèse développée dans la Section A.1.3.

[ColB] Coloration b -bornée (version décision)

Données : Un graphe G , deux entiers b et k .

Résultat : Existe-t-il une k -coloration de G telle que chaque couleur soit utilisée pour au plus b sommets?

Commentaires : [Color] est un cas particulier (pour $b = +\infty$). Partition en triangles est un cas particulier (sur le graphe \overline{G} pour $b = 3$).

[PrExt] Extension pré-colorée (version décision)

Données : Un graphe G , un entier k et une pré-coloration \mathcal{Q} de G utilisant seulement des couleurs dans $\{1, \dots, k\}$.

Résultat : Existe-t-il une coloration de G qui étend \mathcal{Q} ?

Commentaires : [Color] est un cas particulier (pour $\mathcal{Q} = \emptyset$).

Théorème A.2 *Les problèmes suivants sont (maximaux) polynomiaux*

- $B1 \mid b \leq n, \text{col}(\text{forest}) \mid C_{\max}$ [9, 102]
- $B1 \mid b \leq n, \text{col}(\text{claw-free-Berge}) \mid C_{\max}$ [55, 93, 79], Th. 3.6
- $B1 \mid b \leq n, \text{col}(\text{co-}K_4\text{-free-Berge}) \mid C_{\max}$ [93, 76], Th. 3.15
- $B1 \mid b \leq n, \text{col}(\text{co-chordal}) \mid C_{\max}$ Th. 3.35

- $B1 \mid b = \text{const}, \text{col}(P_4\text{-free}) \mid C_{\max}$ [22]
- $B1 \mid b = \text{const}, \text{col}(\text{bipartite}) \mid C_{\max}$ [22]

- $B1 \mid b \leq n, \text{col}(P_4\text{-free}) \mid C_{\max} = \text{const}$ [22]
- $B1 \mid b \leq n, \text{col}(\text{interval}) \mid C_{\max} = \text{const}$ [22]

Théorème A.3 *Les problèmes suivants sont (minimaux) NP-durs*

- $B1 \mid b \leq n, \text{col}(\{P_4, \text{paw}, C_4\}\text{-free}) \mid C_{\max}$ [22], Th. 3.8
- $B1 \mid b \leq n, \text{col}(\{P_4, \text{paw}, 2K_2 + v\}\text{-free}) \mid C_{\max}$ Th. 3.8

- $B1 \mid b = 4, \text{col}(\text{interval}) \mid C_{\max}$ [22]

- $B1 \mid b \leq n, \text{col}(\text{bipartite}) \mid C_{\max} = 3$ [22]

Théorème A.4 *Les approximations suivantes sont (maximales) polynomiales*

- $B1 \mid b \leq n, \text{col}(\text{perfect}) \mid C_{\max} \mid \rho = 2$ Th. 3.59
- $B1 \mid b \leq n, \text{col}(\text{line-graph}) \mid C_{\max} \mid \rho^* = \frac{4}{3}$ Prop. 3.60
- $B1 \mid b \leq n, \text{col}(\text{bipartite}) \mid C_{\max} \mid \rho^* = \frac{4}{3}$ Prop. 3.60

Théorème A.5 *Les approximations suivantes sont (minimales) NP-dures*

- $B1 \mid b \leq n, \text{col}(\text{line-graph}) \mid C_{\max} \mid \rho = \frac{4}{3} - \epsilon$ Prop. 3.60
- $B1 \mid b \leq n, \text{col}(\text{bipartite}) \mid C_{\max} \mid \rho = \frac{4}{3} - \epsilon$ Prop. 3.60

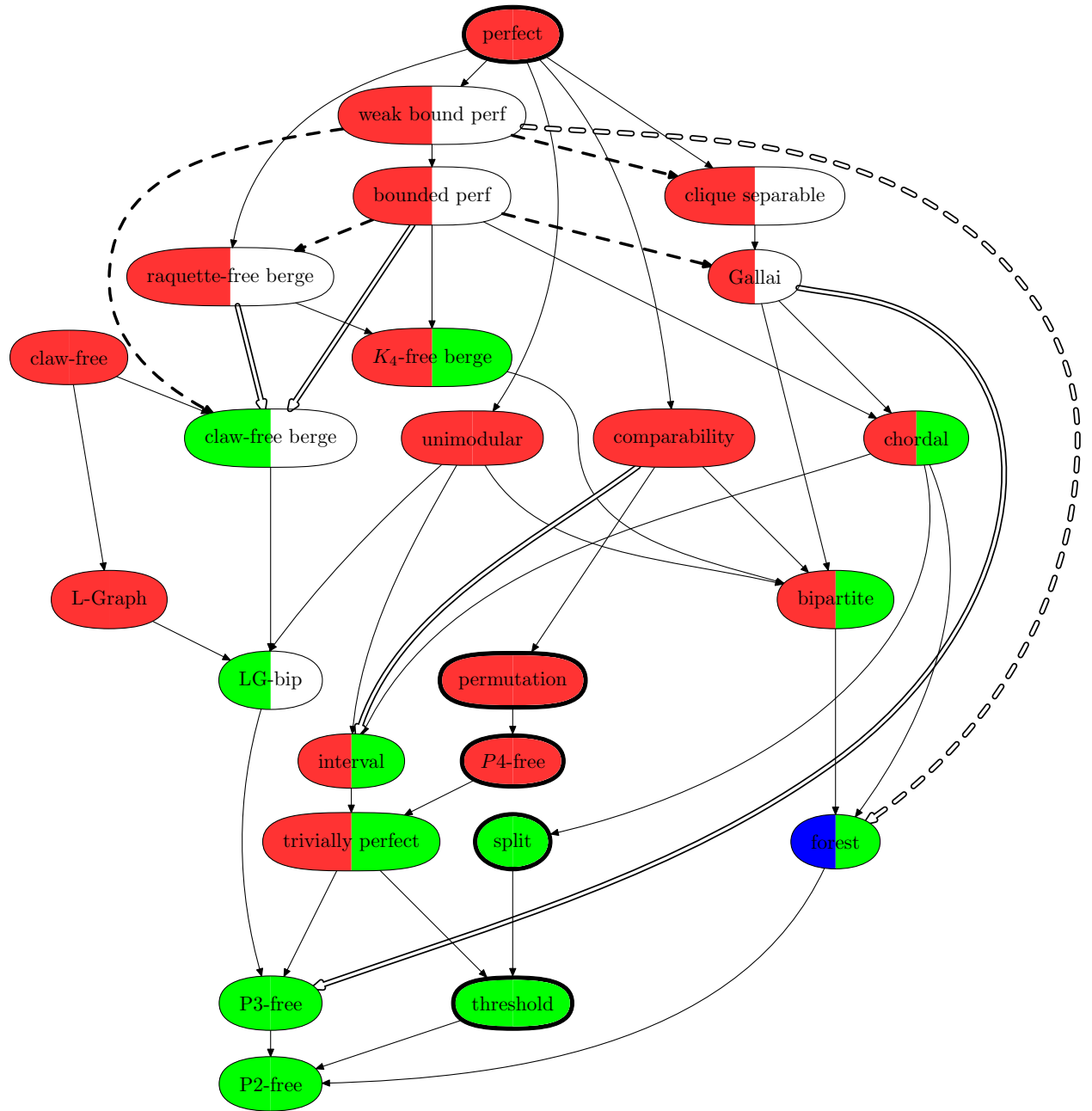


Figure A.3 – Complexité de [ColB] sur quelques classes de graphes et sur les classes complémentaires ($\mathcal{G} \mid \overline{\mathcal{G}}$). Rouge : NP-complet. Vert : borné-parfait + polynomial. Bleu : polynomial. Blanc : ouvert.

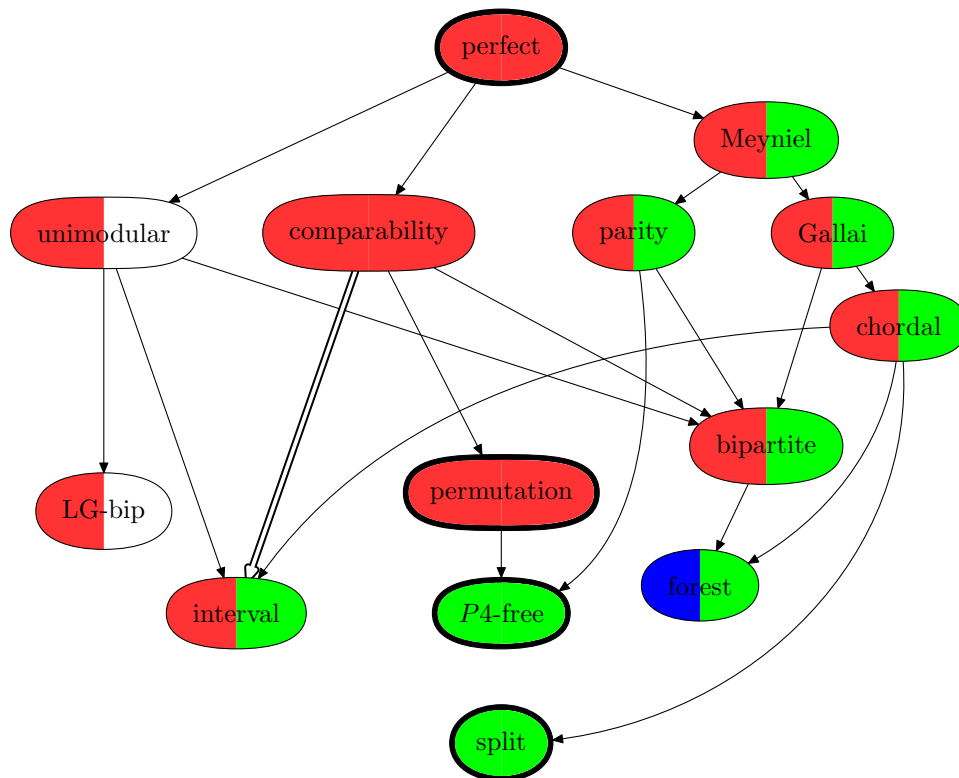


Figure A.4 – Complexité de [PrExt] sur quelques classes de graphes et sur les classes complémentaires ($\mathcal{G} \mid \overline{\mathcal{G}}$). Rouge : NP-complet. Vert : PrExt-parfait + polynomial. Bleu : polynomial. Blanc : ouvert.

Annexe B

Mots équilibrés et ordonnancement juste-à-temps

- Dans cette annexe, nous présentons brièvement les résultats discutés dans
- [31] Small deviations, JIT sequencing and symmetric case of Frankel's conjecture –2005– (accepté à *Discrete Mathematics* avec N. Brauner).
 - [111] Simultaneous optimization of classical objectives in JIT scheduling –2005– (accepté à *European Journal of O.R.* avec V. Lebacque et N. Brauner).

Comme le niveau de discussion de ces papiers est très technique, nous mettons ici l'accent sur les aspects philosophiques sous-jacents et les perspectives mathématiques plus générales dans lesquelles nous pensons qu'ils s'inscrivent. En particulier, nous discutons le principe de proportionnalité dans le cadre plus général de la cohérence. Nous essayons aussi de faire ressortir quelques dangers sournois créés par le choix des fonctions objectifs en juste-à-temps. Nous suggérons que la complexité algorithmique offre un angle d'attaque intéressant pour analyser et généraliser la conjecture de Fraenkel.

Mots clés : Partage et représentation proportionnelle en nombres entiers, mots équilibrés et séquences de Beatty, ordonnancement juste-à-temps, optimisation multi-objectifs simultanée.

Autres objets utilisés : Principe de proportionnalité, partages cohérents, transdisciplinarité, holisme et modélisation.

B.1 Petite philosophie de l'interfécondation éthique-économie-science

Selon Victor Hugo,

“Le monde matériel repose sur l’équilibre, le monde moral sur l’équité”.

L’équité dans le partage des choses est une préoccupation sociale omniprésente, qu’il s’agisse d’impôt, de démocratie représentative, ou encore d’accès aux soins [11]. Si l’on en croit la loi du moindre effort dans la philosophie Védique, ou encore la description des lois physiques selon Lagrange, l’optimalité et l’équilibre ne sont que deux manières de décrire les phénomènes naturels. L’équilibre est donc un souci économique primordial¹, qu’il s’agisse de fiabilité des procédés de fabrication, de compétitivité dans l’organisation des transports, ou encore de stabilité dans les prises de décisions collectives². A plus long terme, l’équilibre se traduit par la nécessité de faire évoluer notre mode de vie selon des critères de développement durables, sous peine de nous voir contraints au silence comme de vulgaires criquets³.

Or il semble que les conclusions des points de vue basées sur l’équité et sur l’équilibre convergent dans l’analyse mathématique de nombreuses situations. Par exemple, la méthode de Sainte-Laguë [12, 10, 11] pour les partages proportionnels en nombres entiers remporte autant de palmes selon des critères moraux que selon des critères d’optimisation. Le monde matériel et le monde moral ne seraient-ils pas deux illusions, ou plutôt deux descriptions complémentaires d’une même réalité plus vaste, dont nous tentons péniblement de rendre compte au travers de nos perceptions et concepts morcelés ?

Les diversités culturelles constituent une richesse particulière dans cette quête de connaissance : voir le voisin faire différemment nous offre la possibilité de remettre en question des a priori culturels mal fondés dont nous n’avons même pas conscience [11]. Mieux, ces diversités sont souvent une manière privilégiée d’envisager de nouvelles manières de faire, en tirant parti de l’expérience des autres [53] et en cherchant ce qui pourrait nous unir au delà de ce qui semble devoir nous séparer (Section B.4). En ce sens, la communauté des chercheurs se doit de participer à l’amélioration des outils de communications⁴ et de vulgarisation, tout autant que de développer des connaissances nouvelles par la recherche spécialisée.

¹La prolifération des perches du Nil et l’activité humaine collatérale dans le film “Le cauchemar de Darwin” peut laisser penser que l’équilibre n’est pas une nécessité de la nature : la loi du plus fort fait encore mieux l’affaire. Mais si l’on envisage les choses à l’échelle des civilisations et non des individus, il semble évident que la croissance illimitée est une illusion immature et l’exploitation intense, une folie suicidaire. Après l’apogée, le déclin : lorsque les criquets ont dévoré toutes les ressources, ils sont condamnés à se faire oublier et à espérer que la flore se régénère. . .

²Un lien entre la stabilité dans les jeux coopératifs et l’optimisation combinatoire est discuté dans les Sections 1.4 et 4.1.1 de ce mémoire.

³Est-il nécessaire d’évoquer la liste des déséquilibres planétaires auxquels participe l’humanité ? La “surpêche” (des océans) illustre de manière particulièrement imagée que les déséquilibres que nous sommes capables de percevoir en tant qu’individus ne sont que la partie émergée de l’iceberg des évolutions auxquelles nous participons.

⁴Nous avons tenté d’y participer avec l’analyse proposée dans l’Annexe A.

B.2 Problèmes de partages, solutions par des mots

Un *mot* infini (respectivement, fini) sur un alphabet Σ est une fonction de \mathbb{N} dans Σ (respectivement, de $[1..t]$ dans Σ). Par exemple $abacaba\dots$ est le début d'un mot sur $\{a, b, c\}$.

Exemple B.1 *A la mort d'un vieux chamelier, un notaire est en charge de partager les chameaux de celui-ci entre ses trois fils : Arthas, Balnazarr et César. Lors de l'écriture de son testament, le vieux chamelier avait 18 chameaux et avait demandé que 9 soient donnés à Arthas, 6 à Balnazarr et 3 à César. Malheureusement, depuis l'écriture du testament, certains chameaux sont morts, d'autres ont été volés et d'autres encore ont été prêtés à des cousins nomades. Finalement, le notaire ne sait pas combien de chameaux doivent être partagés entre les trois fils (car certains chameaux qui ont été prêtés sont susceptibles de revenir dans les prochains mois). Pour contourner le problème, le notaire décide que les chameaux seront donnés un par un et au fur et à mesure qu'on les retrouvera, selon un ordre préétabli, codé par un mot sur $\{a, b, c\}$. Par exemple le mot $aba\dots$ signifie que le premier chameau doit être donné à Arthas, le deuxième à Balnazarr, le troisième à Arthas...*

Les mots donnent un cadre simple et unifié pour traiter certaines questions de l'ordonnancement juste-à-temps (JIT)⁵, du partage des sièges de manière proportionnelle en démocratie représentative ou encore de la théorie des files d'attente. Pour $r : \{1, 2, \dots, n\} \rightarrow [0, 1]$ tel que $\sum_{i=1}^n r_i = 1$, on souhaite trouver un mot tel que la lettre i arrive avec la **densité**⁶ r_i . Autrement dit, si l'on note $x_{i,k}$ le nombre de fois où la lettre i est présente sur l'intervalle $[1..k]$, on souhaite trouver un mot tel que $x_{i,k} \simeq kr_i$. Souvent les r_i sont tous rationnels et on peut se ramener à l'étude de mots périodiques.

B.2.1 Les partages équitables sont-ils proportionnels ?

L'idéal $x_{i,k} = kr_i$ est lié à la philosophie Aristotélicienne qui identifie le concept de partage "juste" à la méthode du partage proportionnel [11]. Michel Balinski [11] donne un coup de pied monumental à notre croyance qui identifie inconsciemment justice et proportionalité⁷ : la proportionnalité est ancrée dans les racines de notre

⁵*just-in-time.*

⁶ r pour *rate*, ou taux de production.

⁷Cette fusion entre un concept philosophique difficilement définissable et une règle mathématique simple est à mon avis un phénomène courant. Blaise Pascal va encore plus loin dans cette critique : "La coutume fait toute l'équité, par cette seule raison qu'elle est reçue; c'est le fondement mystique de son autorité". Par exemple, peu de Français envisagent une remise en cause du système électoral pour les élections présidentielles alors que le scrutin majoritaire à deux tours a montré qu'il était la cause de larges biais quant à sa vocation à faire émerger le candidat représentant le mieux l'ensemble des français (En 1988, les sondages donnaient Raymond Barre

culture, mais l'étude la règle du Ketoubot dans le Talmud [11, 7] montre qu'il existe d'autres manières "cohérentes" que la proportionalité pour définir la justice.

B.2.2 De l'origine des partages représentables par des mots.

Une fois que l'on a choisi une règle de "partage juste" (par exemple entre le Ketoubot et la loi de proportionalité (qui donne le partage kr_i)), il arrive souvent que la répartition prescrite ne soit pas un nombre entier (alors que le partage $x_{i,k}$ d'un groupe de k chameaux est entier par définition). Un moyen de contourner le problème consiste à vendre les chameaux et à répartir le bénéfice des ventes. Mais certains problèmes de partages, comme les sièges en démocratie représentative ou l'ordonnancement juste-à-temps, sont intrinsèquement des problèmes de partages en nombres entiers. Plusieurs méthodes ont été discutées pour transformer un partage proportionnel fractionnaire en un partage en nombre entier [10] (cette problématique est le sujet du livre [12]). Une méthode employée historiquement, la méthode de Hamilton, consiste à choisir $x_{i,k} := \lfloor kr_i \rfloor$, puis à affecter les entiers restant par ordre décroissant des parties fractionnaires $kr_i - \lfloor kr_i \rfloor$. Cette méthode optimise de nombreux critères [104, 12] mais mène à un problème majeur : le paradoxe de l'Alabama, du nom de l'état qui a vu son nombre de siège $x_{i,k}$ diminuer lors d'une augmentation du nombre total k de sièges au congrès américain. Le *principe de monotonie de la chambre* est qu'il faut éviter le Paradoxe de l'Alabama. Dans les questions de partage en nombres entiers, se restreindre à des partages descriptibles par des mots est équivalent à se conformer au principe de monotonie de la chambre. Si les partages en nombres entiers sont bien étudiés dans le cadre de l'idéal proportionnel⁸, ça ne semble pas être le cas pour d'autres types de partages cohérents, comme celui prescrit par le Ketoubot.

comme "candidat Condorcet", c'est-à-dire qui aurait battu tous les autres au second tour. En 2002, le débat du second tour a clairement montré l'incapacité de ce type de scrutin à recentrer le débat autour des candidats les plus représentatifs de l'ensemble de la population.). Bien qu'il n'existe pas de systèmes électoral sans biais lorsqu'il s'agit d'élire un unique représentant [10], d'autres modes de scrutins souffrent de moins de défauts [10] (et permettent de recentrer le débat sur les candidats plus proches de l'opinion générale). A l'instar de Pascal, on peut donc critiquer les justifications culturelles qui servent de base à l'identification que nous faisons entre certains principes philosophiques et les procédés techniques utilisés pour *tenter* de mettre en pratique ces principes. Comme le montre implicitement Balinski [10, 11], ces identifications abusives inconscientes peuvent être repérées *grâce* aux différences culturelles (avant d'être analysées en vue d'être améliorées). Faire la distinction de nature entre les principes philosophiques énoncés dans les constitutions et les tentatives de mise en application est un préalable à la cohérence des systèmes politiques, économiques et judiciaires de demain.

⁸voir [12] pour l'approche axiomatique et le chapitre de Kubiak dans [112] pour l'optimisation de certaines fonctions objectifs.

B.2.3 Sciences physiques et RO : l'homogénéité des formules

Une fois l'idéal de proportionalité choisi et la monotonie de la chambre requise, un autre postulat consiste à chercher des mots tels que $\lfloor kr_i \rfloor \leq x_{i,k} \leq \lceil kr_i \rceil$, ou plus généralement, tels que $x_{i,k} - kr_i$ soit petit pour tous i et k . Ce faisant, on postule implicitement que c'est la *différence* $x_{i,k} - kr_i$ qui mesure l'injustice, l'irrégularité ou encore le coût économique de cette irrégularité. Une manière alternative consiste à chercher à ce que le *rapport* $x_{i,k}/kr_i$ soit proche de 1. A priori, ces objectifs ne sont pas contradictoires. Mais la différence et le rapport peuvent donner des résultats sensiblement différents lorsque l'on utilise une "fonction d'agglomération" (comme $\sum_{i,k} ((x_{i,k}/kr_i) - 1)^2$ ou $\max_{i,k} |x_{i,k} - kr_i|$) pour définir l'optimalité d'un mot. Le choix de la fonction d'agglomération est particulièrement important dans les modèles d'ordonnancement juste-à-temps qui étudient la régularité sur plusieurs niveaux⁹. Dans ces modèles, on associe un vecteur caractéristique à chaque type de lettre (si a est un type de voiture, le vecteur précise si ce type de voiture est équipée d'un ou de deux airbags, de l'ABS, voire du nombre de boulons que la voiture comporte...). Dans ce type de contexte, il convient d'être vigilant sur les absurdités auxquelles peuvent conduire les fonctions d'agglomération :

Exemple B.2 *Il est absurde de faire la somme d'un nombre de petits pois et d'un nombre de pastèques pour évaluer une quantité de nourriture. Il est plus pertinent de pondérer les deux termes de cette somme par la masse ou la quantité d'énergie d'un pois et d'une pastèque. On obtient alors un résultat homogène, c'est-à-dire un nombre de kilos ou de calories.*

Il en va de même pour les fonctions objectifs que nous développons en RO : le minimum syndical est que les formules soient "homogènes". Le verdict semble assez sévère pour les objectifs classiques du JIT (basés sur $x_{i,k} - kr_i$) : ils ne sont pas homogènes¹⁰. Dans le cas à un seul niveau, cela ne semble pas mener à des solutions aberrantes. Mais imaginons la situation suivante :

⁹*multi-level JIT scheduling.*

¹⁰La "dimension" de $x_{i,k} - kr_i$ est un nombre de pièce de type i . Prendre le max ou la somme de telles quantités n'est donc pas homogène. D'un autre côté, les objectifs issus de l'agglomération du rapport $(x_{i,k}/kr_i)$ sont homogènes. Par exemple, la minimisation de $\sum_{i,k} ((x_{i,k}/kr_i) - 1)^2$ conduit à l'optimalité d'une "méthode de diviseur", la méthode de Sainte-Laguë-Webster, promue en théorie des votes pour ses propriétés axiomatiques [12, 10, 11].

Exemple B.3 *Un responsable du planning sur une chaîne de production trouver, à l'aide de son ordinateur, le minimum de $\sum_{i,k} (x_{i,k} - kr_i)^2$, modélisation classique dans la littérature pour le cas multi-niveaux (le i variant sur les types de composants des voitures). Vous êtes particulièrement fier de votre solution car le problème est connu pour être NP-dur. Lorsque vous annoncez le planning aux employés, impressionnés par vos connaissances théoriques en optimisation, ils n'osent pas vous faire part de leurs doutes quant à la qualité de votre ordonnancement. Toutefois, au bout de deux heures, puis périodiquement, vous êtes contraints d'arrêter la ligne de production pour cause de surcharge de travail au stand où sont fixés les airbags. Perplexe quant à la raison de cette cacophonie, vous finissez difficilement la journée. Arrive alors l'heure de rendre des comptes à votre chef, le soir même en réunion. Après un quart d'heure de discussion, celui-ci vous traite d'abruti car il vient de réaliser que dans votre fonction objectif, le terme correspondant aux boulons¹¹ dans la fonction objectif a rendu négligeable celui des pièces dont la gestion est critique (et qui surviennent généralement en faibles quantités).*

Moralité, si l'on souhaite avoir des modèles pertinents, il convient d'interpréter leur sens physique et/ou économique. Mais cela semble souvent particulièrement difficile. Une manière courante consiste à tout exprimer en termes monétaires, ou ce qui revient au même, par une fonction d'agrégation à valeurs réelles. Même si la valeur exacte des coefficients reste inconnue pour les calculs appliqués, l'homogénéité des formules est une condition facilement vérifiable qui permet de détecter que certains modèles sont susceptibles de nous guider vers des solutions peu intéressantes. La difficulté de construire des formules pertinentes est un indice du fait que l'on ne comprend pas ce que l'on cherche à optimiser. Cela incite à réétudier le problème avec un point de vue économique. Une approche alternative, la modélisation axiomatique [12], est évoquée dans la Section B.4¹².

B.2.4 L'équivalence des fonctions objectifs.

Toujours en cherchant des mots proches de l'idéal proportionnel, une autre approche, qui permet d'éviter d'avoir à choisir entre plusieurs fonctions objectifs consiste à prouver l'équivalence de ces objectifs. Le cas des objectifs basés sur $x_{i,k} - kr_i$ est étudié en détail dans [111]. Malheureusement, pour la plupart des

¹¹Cette histoire est issue d'une critique que j'ai formulée à l'encontre d'une fonction objectif proposée dans le cadre d'un projet universitaire qui avait pour espoir d'améliorer un planning de production dans une usine de Toyota...

¹²La méthode (ou modélisation) axiomatique [12] consiste à chercher des solutions qui respectent des principes (comme la monotonie de la chambre). Cette approche permet d'exprimer les modèles dans un autre souvent plus adapté pour transcrire les idées informelles. La modélisation axiomatique permet d'éviter d'avoir à choisir un objectif particulier, souvent difficile à justifier. En contrepartie, la difficulté de la méthode axiomatique survient lorsque les principes requis ne sont pas compatibles : choisir entre plusieurs principes est en quelque sorte équivalent à choisir une fonction d'agglomération.

paires de fonctions objectifs, nous avons exhibé des familles de densités (dont la somme vaut 1) telles qu'il n'existe pas de mots (avec les densités prescrites) optimisant les deux objectifs à la fois. Cette impossibilité d'optimisation simultanée a en particulier lieu dans le cas de chacune des trois paires d'objectifs pris dans l'ensemble

$$\left\{ \max_{i,k} |x_{i,k} - kr_i|, \quad \sum_{i,k} |x_{i,k} - kr_i|, \quad \sum_{i,k} (x_{i,k} - kr_i)^2 \right\}$$

B.3 Le problème MDJIT et les mots équilibrés.

En vue de minimiser un objectif basé sur $x_{i,k} - kr_i$, on peut commencer par observer que :

Proposition B.4 [132, 30, 104] *Pour tout $n \in \mathbb{N}$ et tout $r_1 \geq r_2, \geq \dots \geq r_n \geq 0$ tels que $\sum r_i = 1$, il existe un mot tel que*

$$\lfloor kr_i \rfloor \leq x_{i,k} \leq \lceil kr_i \rceil$$

pour tout $1 \leq i \leq n$ et tout $k \in \mathbb{N}$.

Pour des problèmes comme celui de l'Exemple B.1 ou pour le partage de sièges en représentation proportionnelle, il semble suffisant de chercher à ce que $x_{i,k}$ soit proche de kr_i (en termes de $x_{i,k} - kr_i$ ou de $x_{i,k}/kr_i$). Cependant, pour le JIT, une des motivations est de réguler la charge de travail sur la ligne de production. Dans ce cas, l'origine du temps n'a que peu d'importance et c'est plutôt "l'équilibre local" qui est déterminant. En terme d'objectifs juste-à-temps, les ingénieurs de Renault ont d'ailleurs retenu l'intérêt de ce type de contraintes (par rapport aux hypothèses plus classiques du JIT) lors du Challenge ROADEF'05 avec des formules du type : " $x_{i,k+10} - x_{i,k}$ doit être proche de $10 \times r_i$ pour tout k ". Cela se rapproche de la notion d'équilibre étudiée en théorie des nombres. La lettre i est *C-équilibrée* dans un mot si elle apparaît autant de fois sur toutes paires de sous-mots de même taille, en tolérant une différence d'au plus C . Autrement dit, i est équilibrée si pour tout (k_1, l_1) et (k_2, l_2) tels que $l_1 - k_1 = l_2 - k_2$ on a $|(x_{i,l_1} - x_{i,k_1}) - (x_{i,l_2} - x_{i,k_2})| \leq C$. Un mot est *C-équilibré* si toutes ses lettres sont *C-équilibrées* (on dit plutôt "équilibré" pour 1-équilibré).

Nous avons observé que le *C-équilibre* de la lettre i est impliqué par le fait que $|x_{i,k} - kr_i|$ soit petit pour tout k . Cette observation provient de l'étude du **problème MDJIT**¹³ qui consiste, étant donné $B \in \mathbb{Q}_+$, à chercher un mot tel que $|x_{i,k} - kr_i| \leq B$ pour tout $1 \leq i \leq n$ et tout k .

¹³ *Maximum deviation just-in-time.*

Proposition B.5 [104] *Supposons qu'un mot satisfasse $|x_{i,k} - kr_i| \leq B$ pour tout $1 \leq i \leq n$ et tout $k \in \mathbb{N}$.*

- *Si $B < 1/2$, alors le mot est équilibré [30],*
- *Si $B < 3/4$, alors le mot est 2-équilibré,*
- *Si $B < 1$, alors le mot est 3-équilibré.*

Bien que l'on ne puisse pas toujours construire un mot équilibré avec des densités r_i données (même en supposant $\sum r_i = 1$), les Propositions B.4 et B.5 impliquent que l'on peut toujours construire un mot 3-équilibré. La possibilité de toujours pouvoir construire un mot 2-équilibré semble ouverte.

Malgré ce lien entre le MDJIT et les mots C -équilibrés, la complexité algorithmique nous amène à penser que ces deux problèmes sont relativement différents. Soit $n \in \mathbb{N}$, $B \in \mathbb{Q}_+$, $d_1 \geq d_2 \geq \dots \geq d_n \geq 0$ une liste d'entiers et $D := \sum d_i$. On souhaite construire un mot de D lettres sur l'alphabet $\{1, 2 \dots n\}$ avec les densités d_i/D et des "propriétés de régularité" (*i.e.* un petit B pour MDJIT ou un petit C pour les mots équilibrés).

Pseudo-polynomial, est-ce efficace ?

Avant de discuter la complexité de ces problèmes, il nous faut noter le point philosophique suivant. *Du point de vue de la complexité algorithmique, l'étude des mots doit accorder une place d'honneur au cas du codage des d_i en unaire.* Autrement dit, les algorithmes pseudo-polynomiaux pour construire des mots peuvent souvent être considérés comme efficaces en pratique. Nous tentons de donner un argument pour cette affirmation. Dans l'Exemple B.1, une fois le procédé de partage décidé, il faut le mettre en place. Or, dans les mondes matériels et économiques, le coût logistique de n chameaux (transport, nourriture, soins...) est plutôt de l'ordre de n que de $\log(n)$. Autrement dit, il est acceptable que le temps de calcul de la procédure d'affectation des chameaux soit multiplié par 2 lorsque le nombre de chameaux à distribuer double, car en ce cas, le temps de calcul de la procédure de décision reste raisonnable par rapport au coût de sa mise en place.

Théorème B.6 [132] *MDJIT est pseudo-polynomial (et se réduit à trouver un couplage parfait dans un graphe biparti convexe ayant $2D$ sommets).*

Dans le codage unaire des d_i , MDJIT est donc un problème facile. Pour ce qui est du codage binaire, l'appartenance de MDJIT à NP est ouverte, cependant :

Théorème B.7 [30] *MDJIT est dans co-NP : on peut certifier en temps polynomial qu'il n'existe pas de mot ayant les densités $r_i := d_i/D$ avec une déviation maximum ne dépassant pas B .*

Le Théorème B.7 se déduit des conditions de Hall (pour l'existence d'un couplage parfait) appliquées au graphe construit dans [132] : ces conditions se traduisent par une condition arithmétique compacte.

Pour l'existence d'un mot équilibré, la situation est inverse : on peut certifier en temps polynomial (pour un codage binaire) qu'un mot équilibré existe. Cela provient en grande partie du fait que les séquences 1-équilibrées sont des objets mathématiques bien plus structurés que leur définition ne le laisse supposer. Une *séquence de Beatty* est un sous-ensemble de \mathbb{Z} de la forme : $\{[r_i^{-1}t + \beta_i]\}_{t \in \mathbb{Z}}$. Toute séquence de Beatty est équilibrée et réciproquement, toute séquence équilibrée est une séquence de Beatty à partir d'un certain entier [119, 120, 135]. Cette quasi-équivalence est à la base de la preuve du résultat suivant.

Théorème B.8 [Non publié] Soit $n \in \mathbb{N}$, $d_1 \geq d_2 \geq \dots \geq d_n \geq 0$ et $D := \sum d_i$. Décider s'il existe un mot équilibré sur n lettres tel que la i -ème lettre a la densité d_i/D est un problème NP.

Idée de preuve Pour toute séquence équilibrée $S \subseteq \mathbb{Z}$ de densité r_i , il existe un nombre entier t_i et un nombre β_i tels que

$$S \cap [t_i.. \infty] = \{[r_i^{-1}t + \beta_i]\}_{t \in \mathbb{Z}} \cap [t_i.. \infty]$$

En particulier, si r_i est rationnel, la séquence S est périodique à partir de t_i . Un mot équilibré sur n lettres est donc périodique à partir de $\max\{t_i \mid i \in [1..n]\}$. Si l'on considère une période de ce mot et qu'on la répète, on obtient une partition de \mathbb{Z} en séquences de Beatty. Il existe donc un mot équilibré avec les densités r_i si et seulement s'il existe une liste $\{\beta\}_{i=1}^n$ tels que pour tout $0 \leq i < j \leq n$ on a :

$$(B.1) \quad \{[r_i^{-1}t + \beta_i]\}_{t \in \mathbb{Z}} \cap \{[r_j^{-1}s + \beta_j]\}_{s \in \mathbb{Z}} = \emptyset$$

Chacune des $\mathcal{O}(n^2)$ équations (B.1) peut être réécrite comme un PLNE avec 3 variables. Les équations (B.1) peuvent donc être vérifiées en temps polynomial [128]¹⁴. La liste des β est donc un certificat NP de l'existence d'un mot équilibré car l'on peut supposer que $\beta_i = b_i/d_i$ avec $0 \leq b_i < D$. \square

Remarque B.9 Il serait intéressant d'avoir une caractérisation arithmétique simple pour résoudre les équations B.1, plutôt que d'invoquer la machinerie de Lenstra ou de Barvinok pour la PLNE en dimension fixe.

La question suivante ne semble pas étudiée explicitement :

Question B.10 Existe-t-il un algorithme (pseudo-)polynomial pour décider s'il existe un mot équilibré avec des densités données ? Ce problème est-il (fortement-)NP-complet ? Est-il dans co-NP ?

A la place, une question d'apparence anecdotique concernant le cas où toutes les densités sont différentes tient le devant de la scène des publications concernant les mots équilibrés :

¹⁴La PLNE en dimension fixe est polynomiale.

Conjecture B.11 (Fraenkel) *Soit $n \geq 3$ et $r_1 > r_2 > \dots > r_n > 0$ tels que $\sum r_i = 1$. Il existe un mot équilibré sur n lettres tel que la lettre i a la densité r_i si et seulement si pour tout i , on a*

$$r_i = \frac{2^{n-i}}{2^n - 1}$$

Pour le cas $n = 2$, toutes les paires de nombres $(r_1, 1 - r_1)$ sont valides car le complément d'une séquence de Beatty est encore une séquence de Beatty. Pour $n = 3$ et 4, une caractérisation des n -uplets de densités avec lesquelles on peut construire un mot équilibré est donnée dans [134] et [5]. La Conjecture de Fraenkel est prouvée pour $n = 3$ [134], $n = 4$ [5], $n = 5$ [135] et $n = 6$ [136]. C'est la nécessité qui résiste dans cette conjecture. La suffisance est facile car on peut récursivement construire des mots équilibrés avec les densités prescrites à partir de *abacaba* : si M est la période pour le mot sur n lettres et z est une nouvelle lettre, MzM donne la période pour le mot sur $n + 1$ lettres. Notons que cette période est toujours "symétrique", mais cela est seulement un fait empirique. En fait, on peut prouver la Conjecture de Fraenkel si l'on se restreint aux mots symétriques (la symétrie d'un mot implique en particulier que ses densités soient différentes).

Théorème B.12 [31] *Soit $n \geq 3$ et $r_1 \geq r_2 \geq \dots \geq r_n > 0$ tels que $\sum r_i = 1$. Alors il existe un mot équilibré symétrique sur n lettres tel que la lettre i a r_i pour densité si et seulement si pour tout i , on a*

$$r_i = \frac{2^{n-i}}{2^n - 1}$$

B.4 Epilogue - Théorie des jeux et RO : la cohérence.

Comme le souligne Michel Balinski [11], il existe un principe mathématique qui justifie aussi bien les partages proportionnels que les partages promus par le Talmud : la cohérence. La cohérence est un critère mathématique issu de la théorie des jeux, dont la simplicité et la généralité en font quasiment un critère nécessaire d'équité [11]. Sa force tient en partie dans la généralité de sa définition qui lui permet d'éclairer les questions d'équité dans des contextes très généraux et en particulier pour le contexte des partages en nombres entiers. La définition formelle peut paraître abstraite, mais elle traduit simplement le bon sens : si l'on considère un partage entre les trois fils A, B et C, comme équitable, alors on devrait en particulier considérer ce partage comme équitable si l'on ne compare que les attributions de A et B¹⁵. Nous définissons maintenant la cohérence dans un cadre général avant de l'illustrer par un contre-exemple dans le cas du problème des chameaux.

¹⁵Quelques absurdités matérielles auxquelles mènent les partages non-cohérents sont soulignés dans [11].

Soit M une méthode de partage sur un ensemble Σ de bénéficiaires. On suppose que M permet aussi d'effectuer des partages sur les sous-ensembles de Σ . Alors M est *cohérente* si pour toute répartition x produite par M et tout $U \subseteq \Sigma$, la restriction de x à U est aussi une répartition produite par M . Considérer la cohérence comme un principe nécessaire a conséquence est sévère pour l'analyse des fonctions classiquement utilisées en JIT. . . Par exemple, les solutions données par le MDJIT ne sont pas cohérentes, comme nous l'expliquons maintenant. Considérons le jeu de demandes $(1, 1, 4, 4)$. A quelques symétries près¹⁶, l'unique solution optimale pour cette instance est *cdacdcbcd* [111]. Or, si l'on se restreint aux lettres a et c (*i.e.* aux demandes $(1, 4)$), MDJIT donne l'unique solution *ccacc*. Les solutions optimales de MDJIT ne sont donc pas cohérentes car pour respecter les solutions données sur les paires de lettres, une solution optimale pour les demandes $(1, 1, 4, 4)$ devrait permettre de placer a et b au milieu du mot.

La réponse classique, donnée par la communauté RO pour défendre les objectifs comme $\max_{i,k} |x_{i,k} - kr_i|$ ou $\sum_{i,k} |x_{i,k} - kr_i|$ est que, ce qui compte du point de vue opérationnel, c'est la minimisation des coûts et non la justice envers les types de pièces. Nous sommes donc renvoyés à la mise en garde proposée dans les Sections B.1 et B.2.3 :

Lorsque l'on souhaite appréhender les coûts de manière globale, voire dans une optique de développement durable, peut-on espérer modéliser correctement "la minimisation des coûts" ?

J'aimerais revenir sur un élément de réponse, pour moi très constructif, car il donne du sens et guide le plaisir dans l'organisation des connaissances. Cet élément de réponse consiste en un acte de foi : la démarche transdisciplinaire est un prérequis à la pertinence et la simplicité¹⁷ de ce que nous entreprenons. Plus précisément, la quête holistique est nécessaire à la profondeur, la beauté et la pérennité de nos constructions mentales. Autrement dit,

Nous construisons sans cesse des modèles mathématiques pour tenter de quantifier les principes de bases de la physique, de l'éthique et de l'économie. Le destin de nos modèles n'est-il pas de décrire une réalité unique, ou du moins cohérente, dont la simplicité transcende les contradictions inhérentes au cloisonnement culturel du monde dans lequel nous évoluons ?

¹⁶Ce qui nous intéresse ici c'est que l'on ne peut échanger la place de a qu'avec b [111].

¹⁷Dans le sens de l'épuration du superflu.

Bibliographie

- [1] S. Aaronson and G. Kuperberg. Complexity zoo. <http://qwiki.caltech.edu/wiki/ComplexityZoo>.
- [2] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. 2004.
- [3] M. Aigner and G. Ziegler. *Proofs from THE BOOK*. Springer-Verlag, Berlin - Heidelberg - New-York, 1999.
- [4] N. Alon and A. Orlitsky. Source coding and graph entropies. *IEEE Trans. Inf. Theory*, 42(5) :1329–1339, 1996.
- [5] E. Altman, B. Gaujal, and A. Hordijk. Balanced sequences and optimal routing. Technical report 3180, INRIA, 1997.
- [6] R. J. Aumann and S. Hart. *Handbook of game theory with economic applications*. Handbooks in Economics. 11. Amsterdam : Elsevier, 1992-1994-2002.
- [7] R. J. Aumann and M. Maschler. Game theoretic analysis of a bankruptcy problem from the talmud. *Journal of Economic Theory*, 36 :185–213, 1985.
- [8] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and approximation. Combinatorial optimization problems and their approximability properties*. Springer Berlin, 1999.
- [9] B. S. Baker and E. G. Coffman. Mutual exclusion scheduling. *Theor. Comput. Sci.*, 162(2) :225–243, 1996.
- [10] M. Balinski. Dossier : Théorie des votes. *Pour la science*, 294 :46–69, 2002.
- [11] M. Balinski. Quelle équité? *Pour la science*, 311 :82–87, 2003.
- [12] M. Balinski and H. Young. *Fair Representation : meeting the ideal of one man one vote*. Yale University Press, New Haven and London, 1982.
- [13] A. Bar-Noy and G. Kortsarz. Minimum color sum of bipartite graphs. *J. Algorithms*, 28(2) :339–365, 1998.
- [14] L. Becchetti, P. Korteweg, A. Marchetti-Spaccamela, M. Skutella, L. Stougie, and A. Vitaletti. Latency constrained aggregation in sensor networks. Workshop of Combinatorial Optimization, Aussois, 2006.
- [15] M. Bellare and S. Goldwasser. The complexity of decision versus search. *SIAM J. Comput.*, 23(1) :97–119, 1994.
- [16] C. Bentz and C. Picouleau. Feasible node colorings of trees with cardinality constraints, 2006. soumis.

- [17] C. Berge. *Graphes 3rd ed.* Gautier-Villars, Paris, France, 1983.
- [18] C. Berge. Minimax relations for the partial q -colorings of a graph. *Discrete Math.*, 74(3-14) :15–27, 1989.
- [19] J. Bilbao. *Cooperative games on combinatorial structures.* Kluwer Academic Publishers, Dordrecht, 2000.
- [20] M. Biró, M. Hujter, and Z. Tuza. Precoloring extension. I : Interval graphs. *Discrete Math.*, 100(1-3) :267–279, 1992.
- [21] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, and J. Weglarz. *Scheduling computer and manufacturing processes. 2nd ed.* Springer, 2001.
- [22] H. Bodlaender and K. Jansen. Restrictions of graph partition problems. part I. *Theor. Comput. Sci.*, 148(1) :93–109, 1995.
- [23] H. L. Bodlaender and F. V. Fomin. Equitable colorings of bounded treewidth graphs. In Springer-Verlag, editor, *MFCSS : 29th Symposium on Mathematical Foundations of Computer Science*, pages 180–190, 2004. LNCS 3153.
- [24] H. L. Bodlaender, K. Jansen, and G. J. Woeginger. Scheduling with incompatible jobs. *Discrete Appl. Math.*, 55(3) :219–232, 1994.
- [25] O. Bondareva. Some applications of the methods of linear programming to the theory of cooperative games. *Probl. Kibern.*, 10 :119–139, 1963.
- [26] F. Bonomo and M. Cecowski. Between coloring and list-coloring : μ -coloring. *Elect. Notes. Discr. Math.*, 19 :117–123, 2005.
- [27] M. Boudhar. Dynamic scheduling on a single batch processing machine with split compatibility graphs. *J. of Mathematical Modelling and Algorithms*, 2 :17–35, 2003.
- [28] A. Brandstädt and L. Van Bang. Information system on graph class inclusions. <http://www.teo.informatik.uni-rostock.de/isgci/>.
- [29] A. Brandstädt, L. Van Bang, and J. P. Spinrad. *Graph classes : a survey.* SIAM Monographs on Discrete Mathematics. and Applications, 3. Philadelphia, 1999.
- [30] N. Brauner and Y. Crama. Facts and questions about the maximum deviation just-in-time scheduling problem. *Discrete Applied Math.*, 134 :25–50, 2004.
- [31] N. Brauner and V. Jost. *Small deviations, JIT sequencing and symmetric case of Frankel's conjecture.* Laboratoire Leibniz-IMAG, 2005. accepté à *Discrete Math.*
- [32] P. Brucker. *Scheduling algorithms. 4th edition.* Springer, 2004.
- [33] P. Brucker, A. Gladky, H. Hoogeveen, M. Kovalyov, C. Potts, T. Tautenhahn, and S. van de Velde. Scheduling a batching machine. *J. of Scheduling*, 1 :31–54, 1998.
- [34] P. Brucker and S. Knust. Complexity results of scheduling problems. www.mathematik.uni-osnabrueck.de/research/OR/class/.

- [35] M. Buriel and J. Fonlupt. Polynomial algorithm to recognize a Meyniel graph. *Ann. Discrete Math.*, 21 :225–252, 1984.
- [36] M. Buriel and J. Uhry. Parity graphs. *Ann. Discrete Math.*, 16 :1–26, 1982.
- [37] M. Buriel and J. Uhry. Parity graphs. *Ann. Discrete Math.*, 21 :253–277, 1984.
- [38] J. S. C. Bornstein. Personal communication. Rio de Janeiro, November 2003.
- [39] K. Cameron. A min-max relation for the partial q -colourings of a graph. II : Box perfection. *Discrete Math.*, 74(1-2) :15–27, 1989.
- [40] K. Cameron and J. Edmonds. Existentially polytime theorems. In *Polyhedral combinatorics, Proc. Workshop, Morristown/NJ (USA) 1989, DIMACS, Ser. Discret. Math. Theor. Comput. Sci. 1, 83-100*. 1990.
- [41] J. Cardinal, S. Fiorini, and G. Joret. Minimum entropy coloring. In *Proc. Int. Symposium on Algorithms and Computation (ISAAC), LNCS 3827*, pages 819–828, 2005.
- [42] B.-L. Chen, M.-T. Ko, and K.-W. Lih. Equitable and m -bounded coloring of split graphs. *Lecture Notes in Computer Science*, 1120 :1–6, 1996.
- [43] B.-L. Chen and K.-W. Lih. A note on the m -bounded chromatic number of a tree. *Eur. J. Comb.*, 14(4) :311–312, 1993.
- [44] G. Chen, M. S. Jacobson, A. E. Kézdy, and J. Lehel. Tough enough chordal graphs are hamiltonian. *Networks*, 31(1) :29–38, 1998.
- [45] M. Chudnovsky, G. Cornuéjols, X. Liu, P. D. Seymour, and K. Vušković. Recognizing Berge graphs. *Combinatorica*, 25(2) :143–186, 2005.
- [46] C. J. Colbourn. The complexity of completing partial latin squares. *Discrete Appl. Math.*, 8 :25–30, 1984.
- [47] D. Cornaz. On forests, stable sets and polyhedras associated with clique partitions, 2006. soumis.
- [48] D. Cornaz. Personal communication, april 2006.
- [49] G. Cornuéjols. *Combinatorial optimization. Packing and covering*. CBMS-NSF Regional Conference Series in Applied Mathematics. 74. SIAM, 2001.
- [50] G. Cornuéjols, J. Fonlupt, and D. Naddef. The traveling salesman problem on a graph and some related integer polyhedra. *Math. Program.*, 33 :1–27, 1985.
- [51] P. Crescenzi and V. Kann. A compendium of NP optimization problems. <http://www.nada.kth.se/viggo/problemlist/compendium.html>.
- [52] E. Dahlhaus and M. Karpinski. Matching and multidimensional matching in chordal and strongly chordal graphs. *Discrete Appl. Math.*, 84(1-3) :79–91, 1998.
- [53] S. Darnil and M. Le Roux. *80 hommes pour changer le monde. Entreprendre pour la planète*. JC Lattès, Paris., 2005.

- [54] D. de Werra. Optimization in edge chromatic scheduling. In *Survey of mathematical programming, Proc. int. Symp.*, volume 2, pages 379–382, Budapest, 1976.
- [55] D. de Werra. Some uses of hypergraphes in timetabling. *Asia-Pacific J. Oper. Res.*, 2 :2–12, 1985.
- [56] D. de Werra. Extensions of coloring models for scheduling purposes. *Eur. J. Oper. Res.*, 92(3) :474–492, 1996.
- [57] D. de Werra. Restricted coloring models for timetabling. *Discrete Math.*, 165-166 :161–170, 1997.
- [58] D. de Werra. On a multiconstrained model for chromatic scheduling. *Discrete Appl. Math.*, 94(1-3) :171–180, 1999.
- [59] D. de Werra and Y. Gay. Chromatic scheduling and frequency assignment. *Discrete Appl. Math.*, 49(1-3) :165–174, 1994.
- [60] D. de Werra and A. Hertz. On perfectness of sums of graphs. *Discrete Math.*, 195(1-3) :93–101, 1999.
- [61] D. de Werra and N. Mahadev. Preassignment requirements in chromatic scheduling. *Discrete Appl. Math.*, 76(1-3) :93–101, 1997.
- [62] D. de Werra, N. Mahadev, and U. Peled. Edge-chromatic scheduling with simultaneity constraints. *SIAM J. Discrete Math.*, 6(4) :631–641, 1993.
- [63] D. de Werra and P. Solot. Compact cylindrical chromatic scheduling. *SIAM J. Discrete Math.*, 4(4) :528–534, 1991.
- [64] F. Della Croce, B. Escoffier, C. Murat, and V. T. Paschos. Probabilistic coloring of bipartite and split graphs. In *International Conference on Computational Science and Its Applications ICCSA'05, LNCS 3483*, 2005.
- [65] M. Demange, D. de Werra, J. Monnot, and V. Paschos. *Time slot scheduling of compatible jobs*. Université Paris IX - Dauphine, 2001. Cahier du Lamsade No. 182.
- [66] X. Deng, T. Ibaraki, and H. Nagamochi. Algorithmic aspects of the core of combinatorial optimization games. *Mathematics of Operations Research*, 24(3) :751–766, 1999.
- [67] X. Deng, T. Ibaraki, H. Nagamochi, and W. Zang. Totally balanced combinatorial optimization games. *Math. Program.*, 87(3) :441–452, 2000.
- [68] E. Desgrippes. *Coordination entre la production et la distribution dans une chaîne logistique*. PhD thesis, Laboratoire GILCO - Grenoble, 2005.
- [69] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. In *Stud. integer Program., Proc. Workshop Bonn 1975, Ann. Discrete Math. 1*, 185-204. 1977.
- [70] B. Escoffier. *Approximation polynomiale de problèmes d'optimisation : Aspects structurels et opérationnels*. PhD thesis, Université Paris IX - Dauphine, 2005.

- [71] B. Escoffier, J. Monnot, and V. T. Paschos. Weighted coloring : Further complexity and approximability results. In *ICTCS*, pages 205–214, 2005.
- [72] S. Even, A. Itai, and A. Shamir. On the complexity of timetable and multi-commodity flow problems. *SIAM J. Comput.*, 5 :691–703, 1976.
- [73] S. Even, A. L. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Inf. Control*, 61 :159–173, 1984.
- [74] U. Faigle. Cores of games with restricted cooperation. *Z. Oper. Res.*, 33(6) :405–422, 1989.
- [75] F. Fiala. NP completeness of the edge precoloring extension problem on bipartite graphs. *J. Graph Theory*, 43(2) :156–160, 2003.
- [76] G. Finke, V. Jost, M. Queyranne, and A. Sebő. *Batch processing with interval graph compatibilities between tasks*. Laboratoire Leibniz-IMAG, 2004. Cahier du Leibniz No. 108, accepté à *Discrete Applied Math*.
- [77] J. Fonlupt and D. Naddef. The traveling salesman problem in graphs with some excluded minors. *Math. Program., Ser. A*, 53(2) :147–172, 1992.
- [78] R. Gandhi, M. M. Halldórsson, G. Kortsarz, and H. Shachnai. Improved bounds for sum multicoloring and scheduling dependent jobs with minsum criteria. In *WAOA 2004, Bergen, Norway, LNCS 3351, 68-82*. 2005.
- [79] F. Gardi. *Ordonnement avec exclusion mutuelle par un graphe d'intervalle ou une classe apparentée : complexité et algorithmes*. PhD thesis, Laboratoire d'informatique fondamentale - Faculté des sciences de Luminy, 2005.
- [80] M. R. Garey and D. S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness*. Springer Verlag, 1979.
- [81] F. Gavril. Algorithms on clique separable graphs. *Discrete Math.*, 19 :159–165, 1977.
- [82] K. Giaro, R. Janczewski, M. Kubale, and M. Malafiejski. A 27/26-approximation algorithm for the chromatic sum coloring of bipartite graphs. In *APPROX 2002, Rome, Italy, LNCS 2462, 135-145*. 2002.
- [83] D. Gijswijt. Personnel communication, december 2005.
- [84] D. Gijswijt, V. Jost, and M. Queyranne. *Clique partitioning of interval graphs with submodular costs on the cliques*, 2006. accepté à *RAIRO-OR*.
- [85] P. Gilmore and A. Hoffman. A characterization of comparability graphs and of interval graphs. *Can. J. Math.*, 16 :539–548, 1964.
- [86] O. Goldreich. On promise problems (in memory of Shimon Even). 2005.
- [87] M. C. Golumbic. Trivially perfect graphs. *Discrete Math.*, 24 :105–107, 1978.
- [88] M. C. Golumbic. *Algorithmic graph theory and perfect graphs*. Computer Science and Applied Mathematics, 1980.
- [89] C. Greene. Some partitions associated with a partially ordered set. *J. Comb. Theory, Ser. A*, 20 :69–79, 1976.

- [90] C. Greene and D. J. Kleitman. The structure of sperner k -families. *J. Comb. Theory, Ser. A*, 20 :41–68, 1976.
- [91] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer Verlag, Berlin, 1993.
- [92] M. M. Halldórsson, G. Kortsarz, and H. Shachnai. Sum coloring interval and k -claw-free graphs with applications to scheduling dependent jobs. *Algorithmica*, 37 :187–209, 2003.
- [93] P. Hansen, A. Hertz, and J. Kuplinsky. Bounded vertex colorings of graphs. *Discrete Math.*, 111(1-3) :305–312, 1993.
- [94] Y. T. Herer and M. Penn. Characterizations of natural submodular graphs : A polynomially solvable class of the tsp. *Proc. Am. Math. Soc.*, 123(3) :673–679, 1995.
- [95] M. D. Hirsch, C. H. Papadimitriou, and S. A. Vavasis. Exponential lower bounds for finding Brouwer fixed points. *J. Complexity*, 5(4) :379–416, 1989.
- [96] D. Hochbaum. *Approximation Algorithms for NP-Hard Problems*. PWS Publishing Company, 1995.
- [97] I. Holyer. The NP-completeness of edge-coloring. *SIAM J. Comput.*, 10 :718–720, 1981.
- [98] M. Hujter and Z. Tuza. Precoloring extension. II. graphs classes related to bipartite graphs. *Acta Math. Univ. Comen.*, New Ser., 62(1) :1–11, 1993.
- [99] K. Jansen. A new characterization for parity graphs and a coloring problem with costs.
- [100] K. Jansen. Approximation results for the optimum cost chromatic partition problem. *J. Algorithms*, 34(1) :54–89, 2000.
- [101] K. Jansen and P. Scheffler. Generalized coloring for tree-like graphs. *Discrete Appl. Math.*, 75(2) :135–155, 1997.
- [102] M. Jarvis and B. Zhou. Bounded vertex coloring of trees. *Discrete Math.*, 232(1-3) :145–151, 2001.
- [103] P. M. Jensen and B. Korte. Complexity of matroid property algorithms. *SIAM J. Comput.*, 11 :184–190, 1982.
- [104] V. Jost. *Deux problèmes d'approximations diophantienne : Le partage proportionnel en nombres entiers et les séquences équilibrées de \mathbb{Z}* . Leibniz-IMAG, 2002. Mémoire de DEA - INPG - Grenoble.
- [105] V. Jost, G. Finke, and A. Queyranne, M. Sebő. Batch scheduling with task compatibilities. In *Seventh Workshop on Models and Algorithms for Planning and Scheduling Problems (MAPSP'05)*.
- [106] V. Jost, B. Lévêque, and F. Maffray. *Precoloring extension in co-Meyniel graphs*. Laboratoire Leibniz-IMAG, 2005. Cahier du Leibniz No. 128, soumis à *Graphs and combinatorics*.

- [107] L. Kangbok, S. Chang, and Y. Hong. Continuous slab caster scheduling and interval graphs. *Production Planning and Control*, 15(5) :495–501, 2004.
- [108] J. Krarup and D. de Werra. Chromatic optimisation : Limitations, objectives, uses, references. *Eur. J. Oper. Res.*, 11 :1–19, 1982.
- [109] J. Kratochvíl and A. Sebő. Coloring precolored perfect graphs. *J. Graph Theory*, 25(3) :207–215, 1997.
- [110] J. Kuipers. A note on the 5-person traveling salesman game. *Z. Oper. Res.*, 38(2) :131–139, 1993.
- [111] V. Lebacque, V. Jost, and N. Brauner. *Simultaneous optimization of classical objectives in JIT scheduling*. Laboratoire Leibniz-IMAG, 2005. accepté à *European Journal of O.R.*
- [112] J. Y.-T. Leung. *Handbook of scheduling. Algorithms, models, and performance analysis*. Chapman & Hall/CRC Computer and Information Science Series., 2004.
- [113] Z. Lonc. On the complexity of some chain and antichain partition problems. In *WG : Graph-Theoretic Concepts in Computer Science, International Workshop WG*, pages 97–104, 1991. LNCS 570.
- [114] F. Maffray and M. Preissmann. A translation of Gallai’s paper : ‘transitiv orientierbare graphen’. In *Ramírez Alfonsín, Jorge L. (ed.) et al., Perfect graphs. Chichester : Wiley. Wiley-Interscience Series in Discrete Mathematics and Optimization. 25-66*. 2001.
- [115] F. Maffray and B. A. Reed. A description of claw-free perfect graphs. *J. Comb. Theory, Ser. B*, 75(1) :134–156, 1999.
- [116] D. Marx. Precoloring extension on unit interval graphs. To appear in *Discrete Applied Mathematics*.
- [117] D. Marx. Precoloring extension on chordal graphs, 2004. Manuscript.
- [118] N. Megiddo and C. H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theor. Comput. Sci.*, 81(2) :317–324, 1991.
- [119] M. Morse and G. A. Hedlund. Symbolic dynamics. *Amer.J. Math*, 60 :815–866, 1938.
- [120] M. Morse and G. A. Hedlund. Symbolic dynamics II - sturmian trajectories. *Amer.J. Math*, 62 :1–42, 1940.
- [121] Y. Okamoto. Traveling salesman games with the monge property. *Discrete Appl. Math.*, 138(3) :349–369, 2004.
- [122] J. G. Oxley. *Matroid theory*. Oxford Science Publications, 1992.
- [123] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [124] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3) :498–532, 1994.
- [125] C. H. Papadimitriou and M. Yannakakis. Scheduling interval-ordered tasks. *SIAM J. Comput.*, 8 :405–409, 1979.

- [126] C. H. Papadimitriou and M. Yannakakis. Note on recognizing integer polyhedra. *Combinatorica*, 10 :107–109, 1990.
- [127] S. V. Pemmaraju and R. Raman. Approximation algorithms for the max-coloring problem. In *ICALP 2005, Lisbon, Portugal, July 11-15 L.N.C.S. 3580, 1064-1075*. 2005.
- [128] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience, 1986.
- [129] A. Schrijver. *Combinatorial Optimization : Polyhedra and Efficiency*. Springer, 2003.
- [130] L. S. Shapley. On balanced sets and cores. *Naval Res. Logist. Quart.*, 14 :453–460, 1967.
- [131] H. Shum and L. Trotter. Cardinality-restricted chains and antichains in partially ordered sets. *Discrete Appl. Math.*, 65(1-3) :421–439, 1996.
- [132] G. Steiner and J. S. Yeomans. Level schedules for mixed-model, just-in-time processes. *Management Science*, 39 :728–735, 1993.
- [133] J. Szabó and Z. Fekete. Personal communication. Budapest, November 2005.
- [134] R. Tijdeman. On complementary triples of Sturmian bisequences. *Indagationes Math.*, 7 :419–424, 1996.
- [135] R. Tijdeman. Exact covers of balanced sequences and Fraenkel’s conjecture. In F. Halter-Koch and R. F. Tichy, editors, *Algebraic Number Theory and Diophantine Analysis*, pages 467–483. Walter de Gruyter, Berlin, New York, 2000.
- [136] R. Tijdeman. Fraenkel’s conjecture for six sequences. *Discrete Mathematics*, 222 :223–234, 2000.
- [137] V. T’Kindt. e-OCEA. <http://www.teo.informatik.uni-rostock.de/isgci/japp.html>.
- [138] M. Yannakakis and F. Gavril. The maximum k-colorable subgraph problem for chordal graphs. *Inf. Process. Lett.*, 24 :133–137, 1987.

Index

- $:=$, 17
- C_n , 95
- C_{max} , 85
- $G_1 \cup G_2$, 98
- $G_1 \oplus G_2$, 98
- K_n , 96
- $L(G)$, 20
- L_{max} , 85
- $N(v)$, 20
- $N^+(v)$, 20
- P_n , 95
- X^n , 17
- \mathbb{N} , 17
- \mathbb{Q} , 17
- \mathbb{R} , 17
- \mathbb{R}_+ , 17
- $\mathcal{SN}(G)$, 157
- $\mathcal{S}(G)$, 155
- \mathbb{Z} , 17
- $\alpha(G)$, 20
- $\alpha_q(G)$, 220
- $\alpha_{(r,d)}(G)$, 220
- \setminus , 17
- $\bar{\chi}(G)$, 21
- $\bar{\chi}_b(G)$, 127
- Δ , 17
- \mathcal{M} -clique, 177
- $\mathcal{O}(f)$, 26
- $\mathcal{P}(X)$, 17
- \cap , 17
- $\chi(G)$, 20
- $\chi_=(G)$, 131
- $\chi_b(G)$, 127
- \cup , 17
- $\delta(S)$, 19
- $\delta(v)$, 19
- $\delta^*(A)$, 31
- $\delta^*(\Pi)$, 31
- $\delta^{in}(S)$, 21
- $\delta^{out}(S)$, 21
- $[x]$, 17
- $\lfloor x \rfloor$, 17
- $\stackrel{def}{=}$, 17
- $\nu(G)$, 21
- $\omega(G)$, 20
- \overline{G} , 20
- $\rho(G)$, 21
- $\rho^*(\Pi)$, 31
- $\sigma_b(G)$, 135
- \subset , 17
- \subseteq , 17
- $\sum C_j$, 85
- $\sum T_j$, 85
- $\sum U_j$, 85
- $\sum f(C_j)$, 85
- $\sum f_j(C_j)$, 85
- $\sum w_j C_j$, 85
- $\sum w_j T_j$, 85
- $\sum w_j U_j$, 85
- $\tau(G)$, 20
- \rightarrow , 18
- 1**, 17
- $\mathbf{1}_U$, 17
- b -clique, 129
- b -truncation, 177
- c -couplage, 158
- $col(G)$, 116
- $opt(I)$, 31
- $sol(I)$, 31
- $x(U)$, 17
- x_v , 17
- [3-PARTITION], 133

- [ColBNU], **164**
- [ColB], **128**
- [ColEq], **131**
- [ColPart], **225**
- [GOCCP], **218**
- [MaxColor], **188**
- [PCliqB], **129**
- [PCliqBatchGen], **190**
- [PCliqMax], **188**
- [PerfectMDM], **132**
- [Pintro], **6, 244**
- [PrExt], **229**

- adjacents, **21**
- algorithme, **25**
- alphabet, **25**
- antenne, **143**
- approximation
 - ρ , **31**
 - différence au plus δ , **31**
- arête, **19, 21**
 - cardinalité, **21**
 - disjointes, **19**
 - extrémité, **19**
 - incidentes, **19**
 - multiple, **20**
 - taille, **21**
- arbre, **104**
- arc, **21**
- articulation
 - clique, **98**
 - ensemble, **98**
 - sommet, **98**

- batch
 - coloration bornée, **129**
- bloquant, **115**
- borné-parfait, **136**
 - faiblement-, **175**
 - minimalement non-, **138**
- boucle, **20**
- box-TDI, **44**
- BT-mod, **135**

- cardinalité, **17**
- certificat
 - NP, **26**
- chaîne, **44**
- chemin, **95**
- circulation, **40**
- clique, **20**
- co-NP, **26**
- co-NPP, **33**
- cohérence , **273**
- collection, **17**
- colorant, **116**
- coloration
 - équitable, **131**
 - bornée
 - non-uniforme, **164**
 - uniforme, **128**
 - graphe, **20**
- combinaison
 - affine, **34**
 - cônique, **34**
 - convexe, **34**
 - linéaire, **34**
- combinatoire polyédrale, **45**
- conséquence (inégalité), **35**
- constellation, **50**
- contraction
 - $Y - \Delta$, **95**
 - d'arête, **94**
 - fantôme , **94**
 - voisinage-clique, **95**
- contraintes
 - d'incompatibilité, **81**
 - de cardinalité, **83**
 - de disponibilité, **82**
- corde, **96**
 - courte, **96**
 - croisées, **96**
 - non-croisées, **96**
- couleur
 - cardinalité, **128**
 - maximum, **128**
- couplage
 - c -, **21**
- couverture

- fractionnaire, **58**
- couverture par des sommets, **20**
- couvre, **18**
- cycle, **95**
 - impair, **95**
 - pair, **95**
- décroiser, **52**
- date
 - échue, **82**
 - de disponibilité, **82**
- degré, **19**
- demi-espace, **35**
- densité, **265**
- diagramme de Hasse
 - le, **19**
 - un, **18**
- digraphe, **21**
- dual
 - au sens matroïdal, **116**
- dualité
 - saut de, **37**
- dualité entière
 - propriété de, **47**
- dupliquer, **97**
- effacer, **94**
- ensemble
 - partiellement ordonné, **18**
- enveloppe
 - cônique, **34**
 - convexe, **34**
- étiquette, **147, 158**
- expressivité, **91**
- extension
 - linéaire, **18**
- FNP, **26**
- fonction d'ensemble, **19**
 - équilibrée, **58**
 - cardinale tronquée, **121**
 - entière, **19**
 - max-batch (\iff p-batch), **120**
 - monotone, **19**
 - non-décroissante, **19**
 - probabiliste, **122**
 - s-batch, **121**
 - sous-additive, **19**
 - sous-cardinale, **19**
 - sp-batch, **121**
 - super-additive, **19**
 - taille-définie, **72**
 - taille-définie concave, **121**
 - totalement-équilibrée, **59**
 - valeur-polymatroïdale, **119**
- forêt, **104**
 - clique-connectante, **50**
 - stellaire, **50**
- FP, **26**
- Ghouila-Houri
 - critère de, **44**
- graphe, **19**
 - $\chi - \omega$, **136**
 - équilibré, **111**
 - biparti, **104**
 - box-parfait, **226**
 - complémentaire, **20**
 - complet, **96**
 - d'arcs circulaires, **111**
 - d'intersection de \mathcal{F} , **92**
 - d'intervalle(s), **106**
 - de comparabilité, **104**
 - de Greene-Kleitman, **225**
 - de Meyniel, **110**
 - de parité, **111**
 - de permutation, **104**
 - des arêtes dépendantes, **50**
 - GTSP
 - (totalement-)équilibré, **67**
 - sous-modulaire, **66**
 - parfait, **101**
 - planaire-externe, **112**
 - sous-jacent, **21**
 - split-, **103**
 - triangulé, **103**
 - trivialement parfait, **109**
 - TSP-parfait, **67**
 - naturellement sous-modulaire, **66**

- simple, **19**
- unimodulaire, **111**
- Helly, **92**
 - clique, **93**
- hypergraphe, **21**
 - d'intersection de \mathcal{F} , **92**
 - héréditaire, **115**
 - rang, **22**
 - simple, **22**
 - transposé, **22**
- idéal
 - inférieur, **18**
 - supérieur, **18**
- inégalité
 - triangulaire, **18**
- incidence
 - matrice, **21**
- incidents, **21**
- indépendant
 - graphe, **20**
- induit, **93**
- insaturé, **146**
- instance, **24**
- involution, **115**
- jeu coopératif
 - cœur, **57**
 - fonction caractéristique, **55**
 - imputation, **56**
 - stable, **57**
 - joueur, **55**
 - sous-jeu, **59**
- join complet, **98**
- jumeau
 - faux, **97**
 - vrai, **97**
- laminaire, **44**
- langage, **25**
- line-graphe, **20**
- longueur
 - mot, **25**
- maison, **96**
- matrice
 - d'incidence, **21**
- matroïde, **23**
 - (in)dépendant, **23**
 - élément, **23**
 - circuit, **23**
 - de partition, **23**
 - des cycles d'un graphe, **23**
 - graphique, **23**
 - hyperplan, **23**
 - rang, **23**
 - uniforme, **23**
- max-batch, **188**
- mineur
 - graphe, **94**
- modèle
 - arithmétique, **30**
 - binaire, **30**
 - unaire, **30**
- mot, **25, 265**
- multi-graphe, **20**
- NP, **26**
 - complet, **28**
 - dur, **28**
 - fortement, **30**
- NPP, **33**
- oracle, **27**
- ordre, **18**
 - d'intervalle(s), **106**
 - linéaire, **18**
 - total, **18**
 - partiel, **18**
- P, **26**
- parfait
 - borné, **136**
- partie entière
 - inférieure, **17**
 - supérieure, **17**
- partition
 - en b cliques
 - bien étiquetée, **158**
 - fractionnaire, **58**

- partition en cliques, **21**
- pertinence applicative, **90**
- plan coupant, **45**
- PLNE, **37**
- polyèdre, **35**
- polynomial, **30**
 - semi-fortement, **30**
- poset, **18**
- pré-ordre, **18**
- préemption, **84**
- problème
 - ρ -approximable, **31**
 - d'optimisation, **31**
 - de décision, **25**
 - associé, **25**
 - polynomial, **26**
- produit cartésien
 - ensembles, **17**
 - graphe, **217**
- programme linéaire, **35**
 - dual, **36**
 - relaxé, **38**
- programme linéaire
 - nombre entiers, **37**
- promesse, **32**
- pseudo-polynomial, **30**
- réalisation, **33**
- réduction
 - de Cook, **27**
 - de Karp, **28**
 - de Turing, **27**
- répliquer, **97**
- rang
 - hypergraphe, **22**
- rapport d'approximation, **31**
- relation, **17**
 - extension d'une -, **18**
 - réflexive, **18**
 - antitransitive, **18**
 - partielle, **18**
 - symétrique, **18**
 - transitive, **18**
- relation min-max, **46**
- robuste, **34**
- robustesse, **34**
- sans- \mathcal{F} , **94**
 - isométriquement, **95**
- saturé, **146**
- segment, **145, 233**
- simplicial(e)
 - ordre, **98**
 - bloc, **155**
 - réseau, **157**
 - séquence, **155**
 - sommet, **98**
- solution
 - réalisable, **35**
- sommet, **19, 21**
 - isolé, **20**
 - universel, **98**
 - adjacent, **19**
 - voisin, **19**
- sous-ensemble propre, **17**
- sous-modulaire
 - fonction, **19**
 - sur une famille héréditaire, **196**
- stable, **20**
- supprimer, **94**
- système
 - box-TDI, **44**
 - TDAU, **45**
 - TDI, **43**
- temps d'opération, **84**
- théorème
 - de Bondareva-Shapley, **59**
 - de dualité de la programmation linéaire, **36**
 - intersection de polymatroïdes, **52**
- troncation
 - de Dilworth, **53**
 - sous-additive, **61**
 - totale-équilibrée, **60**
- trou, **95**
- TSP
 - relaxation graphique, **65, 112**
- TTU, **44**

unimodulaire
 matrice totalement, **44**
union disjointe, **98**

vecteur
 caractéristique, **17**
voisinage, **20**
 étendu, **20**
voit, **19**