



UNIVERSITÉ PARIS-DAUPHINE

ÉCOLE DOCTORALE DE DAUPHINE

THÈSE

Pour l'obtention du
DOCTORAT EN INFORMATIQUE

**Analyse structurelle des systèmes algébro-différentiels
conditionnels : complexité, modèles et polyèdres.**

Présentée par : Sébastien MARTIN

JURY

Directeur de thèse : Ali Ridha MAHJOUB
Co-encadrant : Mathieu LACROIX
Président : Alain QUILLIOT
Rapporteurs : Philippe CHRÉTIENNE
Mohamed DIDI BIHA
Examineurs : Sébastien FURIC
Imed KACEM
Vangelis Th. PASCHOS

Présentée et soutenue publiquement le 7 décembre 2011

Remerciements

Je souhaite remercier Monsieur Ali Ridha Mahjoub, Professeur à l'Université Paris-Dauphine, d'avoir accepté d'être mon directeur de thèse et de m'avoir encadré ces dernières années. Je tiens à lui exprimer toute ma gratitude pour sa disponibilité, ses conseils et l'encadrement exceptionnel qu'il m'a apporté durant l'ensemble de ma thèse. Il a su me transmettre ses connaissances, son expérience, sa rigueur mathématique et sa passion pour la recherche et l'enseignement. Pour toutes ces raisons je lui témoigne ma plus sincère gratitude.

J'aimerais aussi remercier Mathieu Lacroix, Maître de conférences à l'Université Paris 13, pour avoir co-encadré ma thèse. Il a toujours stimulé mon esprit de recherche et il m'a permis de faire un travail de qualité. Je lui suis profondément reconnaissant pour sa constante disponibilité pour les questions de recherche, et pour les relectures des articles et de ce mémoire.

Je suis très reconnaissant de l'attention que Monsieur Philippe Chrétienne, Professeur à l'Université Pierre et Marie Curie à Paris, a porté à mon travail et pour m'avoir fait l'honneur d'accepter de rapporter ma thèse.

J'adresse mes sincères remerciements à Monsieur Mohamed Didi Biha, Professeur à l'Université de Caen, pour l'intérêt qu'il a porté à mon travail et avoir accepté de rapporter ma thèse.

Je souhaite également remercier Monsieur Alain Quilliot, Professeur à l'Université Blaise Pascal à Clermont-Ferrand, d'avoir accepté de présider le jury de ma thèse. De plus, je remercie Monsieur Imed Kacem, Professeur à l'Université de Metz, et Monsieur Vangelis Paschos, Professeur à l'Université Paris-Dauphine d'avoir bien voulu examiner mes travaux et participer au jury.

Je suis également très reconnaissant envers Monsieur Sébastien Furic, Chef de projet à LMS-Imagine, qui m'a fait le plaisir de participer au jury et avec qui j'ai eu une collaboration fructueuse.

Je suis très reconnaissant envers Pierre Fouilhoux, Maître de conférences à l'Université Pierre et Marie Curie à Paris, qui m'a fait découvrir l'optimisation combinatoire. C'est grâce à lui que j'ai pu trouver un domaine qui me passionne.

Je tiens à témoigner toute mon amitié à Mathieu Lacroix, Ibrahima Diarrassouba, Raouia Taktak et Amel Benhamiche avec qui j'ai passé de très bons moments aussi bien en conférence qu'en discussions et débats. Je tiens aussi à remercier les membres de l'équipe EPOC (Equipe Polyèdre et Optimisation Combinatoire) et plus particulièrement Pierre Fouilhoux, Sylvie Borne, Pierre Pesneau et Denis Cornaz qui sont toujours disponibles pour discuter du travail et de tant d'autres choses.

Je tiens aussi à remercier tous les doctorants que j'ai pu rencontrer dans le bureau C605 qui ont su quotidiennement être de bons compagnons ; ceci a facilité la bonne entente et a permis de travailler dans de bonnes conditions. J'ai apprécié cette ambiance chaleureuse qui nous a tous rapprochés quelque soit notre domaine de recherche.

Je remercie tous les membres du LAMSADE avec qui j'ai collaboré pour l'enseignement ou simplement croisé dans les couloirs ou à la salle café. Je remercie aussi toute l'équipe administrative ainsi que notre ingénieur réseau qui se sont toujours montrés disponibles quand j'en ai eu besoin et avec qui j'ai passé de très bons moments.

Enfin, je remercie toute ma famille et mes amis qui m'ont toujours soutenu et encouragé. Ils ont su trouver les mots justes et être présents quand j'en avais besoin. Je remercie du fond du cœur mes parents sans qui je n'aurais jamais pu être ce que je suis devenu.

Résumé

Les systèmes algébro-différentiels permettent de modéliser des systèmes physiques complexes comme les circuits électriques et les mouvements dynamiques. Ils sont souvent de grande taille et difficiles à résoudre. L'analyse structurelle des systèmes algébro-différentiels permet de vérifier si un tel système ne pourra pas être résolu par des méthodes numériques. Elle consiste à résoudre un problème sous-jacent de couplage dans les graphes. Dans cette thèse, nous étudions ce problème dans le cas des systèmes algébro-différentiels conditionnels. Nous montrons que ce problème est équivalent au problème du sous-graphe sans couplage parfait. Nous montrons que ce dernier est NP-difficile. Nous proposons une formulation en termes de graphes et deux formulations en nombres entiers pour ce problème. Nous étudions le polyèdre associé et décrivons plusieurs classes de contraintes valides. Nous donnons des conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes. Nous discutons également d'algorithmes de séparation pour ces contraintes et en utilisant ces résultats, nous développons un algorithme de coupes et branchements. Nous étudions aussi une extension de ce problème pour les systèmes algébro-différentiels conditionnels imbriqués. Nous étendons la plupart des résultats précédent à ce cas. Nous identifions de nouvelles contraintes valides pour cette variante du problème.

Dans une deuxième partie, nous nous intéressons au problème de parallélisation des systèmes algébro-différentiels. Ce problème se ramène au problème du séparateur. Nous proposons plusieurs formulations en nombres entiers du modèle et pour l'une d'entre elle, nous étudions le polyèdre associé. Nous proposons quelques résultats expérimentaux obtenus suite à cette étude.

Mots clés : Analyse structurelle, système algébro-différentiel, graphe, couplage, complexité, polytope, facette, séparation, algorithme de coupes et branchements, problème du séparateur.

Abstract

Differential algebraic systems are used for modeling complex physical systems as electrical networks and dynamic movements. They are often large and difficult to solve. The structural analysis for differential algebraic systems permits to verify if these systems can not be solved with numerical methods. It consists to solve an underlying matching problem in graphs. In this thesis, we consider the structural analysis problem for differential algebraic systems with conditional equations. We show that the structural analysis problem for differential algebraic systems with conditional equations reduces to which we call the perfect matching free subgraph problem. We show the NP-completeness of this latter problem. We propose a formulation in terms of graphs and two integer programming formulations. We study the polytope associated to this problem and describe several classes of valid inequalities. We give necessary and sufficient conditions for these inequalities to be facet defining. We also discuss separation algorithms for these constraints. We develop a branch-and-cut algorithm based on these results. We also study an extension of this problem to differential algebraic systems with conditional embedded equations. We generalize the results obtained for the first variant and give new valid inequalities for this more general problem.

In a second part, we study the parallelization problem for differential algebraic systems. This problem reduces to which is called the separator problem. We give several integer programming formulations, and for one of them we study the associated polytope. We give a few experimental results associated to this polyhedral study.

Keywords : Structural analysis, differential algebraic system, graph, matching, complexity, polytope, facet, separation problem, branch-and-cut algorithm, separator problem.

Table des matières

Introduction	1
1 Notions préliminaires	3
1.1 Problèmes d'optimisation combinatoire	3
1.2 Rappels sur la théorie de la complexité	4
1.3 Éléments de la théorie des polyèdres	5
1.4 Approche polyédrale, méthode de coupes et branchements	6
1.5 Notations et définitions	8
1.6 Couplages et graphes bipartis	11
2 Systèmes algébro-différentiels	17
2.1 Systèmes algébro-différentiels	18
2.1.1 Systèmes différentiels ordinaires	19
2.1.2 Systèmes algébro-différentiels	20
2.2 Résolution des systèmes algébro-différentiels	23
2.2.1 Méthode d'Euler explicite	25
2.2.2 Méthode d'Euler implicite	26
2.3 Analyse structurelle	28
2.3.1 PAS pour les systèmes algébriques	29
2.3.2 PAS pour les SAD	33
2.4 Analyse structurelle et problèmes connexes	35
2.5 Les systèmes algébro-différentiels conditionnels	36
2.6 Conclusion	41
3 Analyse structurelle des systèmes algébro-différentiels conditionnels	43
3.1 Problème de l'analyse structurelle conditionnelle et graphes bipartis . .	43
3.2 Complexité	47

3.2.1	Relation avec le problème du stable	47
3.2.2	NP-complétude	50
3.2.3	Une autre preuve du théorème 3.3	54
3.2.4	Cas polynomial	57
3.3	Formulation du PSGSCP	57
3.4	Formulation du R-PSGSCP	59
3.5	Conclusion	61
4	Étude polyédrale	63
4.1	Polyèdre associé au PSGSCP	63
4.2	Dimension de $P(G, U_c, \pi)$	65
4.3	Inégalités valides et facettes	66
4.3.1	Inégalités triviales	66
4.3.2	Inégalités de satisfiabilités	66
4.3.3	Inégalités de couplages	67
4.3.4	Inégalités de couplages proches	71
4.3.5	Inégalités de k -couplages	76
4.3.6	Inégalités de sous-graphe régulier	79
4.4	Conclusion	80
5	Algorithme de coupes et branchements pour le PSGSCP	83
5.1	Séparation des contraintes	84
5.1.1	Séparation des inégalités de couplages	84
5.1.2	Séparation des contraintes de couplages à dominance maximale	85
5.1.3	Séparation des contraintes de couplages proches	88
5.1.4	Séparation des contraintes de 2-couplages	90
5.1.4.1	Séparation exacte des contraintes de 2-couplages	91
5.1.4.2	Séparation heuristique des contraintes de 2-couplages	97
5.2	Algorithme de coupes et branchements	99
5.3	Résultats expérimentaux	102
5.3.1	Contexte informatique	102
5.3.2	Analyse d'efficacité des contraintes valides	103
5.3.2.1	Description des instances testées	103
5.3.2.2	Résultats obtenus	104
5.3.3	Analyse des paramètres des instances	112

5.3.3.1	Paramètre k	112
5.3.3.2	Paramètre $\nu(H_\theta)$	114
5.4	Conclusion	118
6	Analyse structurelle des systèmes algébro-différentiels conditionnels imbriqués	119
6.1	Systèmes algébro-différentiels conditionnels imbriqués et analyse structurelle	119
6.2	Formulation du problème en termes de graphes	124
6.3	Formulation sous forme d'un programme linéaire en nombres entiers	128
6.4	Étude polyédrale	130
6.4.1	Dimension	130
6.4.2	Inégalités valides et facettes	131
6.4.2.1	Inégalités triviales	131
6.4.2.2	Inégalités de cliques	131
6.4.3	Inégalités de couplages-cliques	133
6.4.3.1	Inégalités de couplages à dominance maximale	135
6.4.4	Inégalités de k -couplages	137
6.5	Conclusion	138
7	Décomposition des systèmes algébro-différentiels et partitionnement de graphes	139
7.1	Matrice Jacobienne et résolution parallèle	140
7.2	Matrices Jacobiennes et graphes	143
7.3	Problème du séparateur	144
7.4	Formulations	145
7.4.1	Formulations approchées	146
7.4.2	Problème du k -séparateur équilibré	147
7.5	Étude polyédrale	148
7.5.1	Dimension de $P(G, k, q)$	148
7.5.2	Inégalités valides	149
7.5.2.1	Inégalités triviales et facettes	149
7.5.2.2	Inégalités de partition	149
7.5.2.3	Inégalités d'arêtes	152
7.5.2.4	Inégalités d'équilibre	155
7.5.2.5	Inégalités de dominants	158

7.5.2.6	Inégalités de borne	161
7.6	Étude expérimentale	163
7.6.1	Description des instances	164
7.6.2	Heuristique primale	165
7.6.3	Séparation des inégalités de dominants connexes	166
7.6.4	Algorithme de coupes et branchements	168
7.6.5	Résultats expérimentaux	169
7.7	Conclusion	176
	Conclusion	181
	Bibliographie	183

Introduction

Les systèmes algébro-différentiels apparaissent dans presque tous les domaines de la science et de l'industrie, par exemple les mathématiques, la physique, la chimie, la biologie et l'économie. Ils servent à traduire les lois qui régissent la variation des valeurs, par exemple un condensateur électrique ou la position d'une navette spatiale. La résolution de ces systèmes est un processus long et difficile. Il est donc très utile de déterminer au préalable si un système algébro-différentiel peut ou non être résolu. Le problème de l'analyse structurelle permet de vérifier une condition nécessaire à la résolution de ces systèmes par des méthodes numériques. Un autre problème intéressant à étudier, en relation avec la résolution de ces systèmes, est la possibilité de paralléliser leur résolution sur plusieurs processeurs. Un algorithme parallèle permettrait de résoudre plus rapidement les systèmes algébro-différentiels.

Les problèmes de l'analyse structurelle et de parallélisation des systèmes algébro-différentiels sont, dans certain cas, des problèmes difficiles. Les cas les plus difficiles ont très peu été étudiés dans la littérature.

L'optimisation combinatoire et en particulier l'approche polyédrale se sont avérées très puissantes pour aborder des problèmes combinatoires difficiles. Cette technique, initiée par Edmons [23] dans le cadre du problème de couplage, consiste à ramener le problème en question à la résolution d'un programme linéaire (ou d'une séquence de programmes linéaires), par la description complète (ou partielle) de son polytope des solutions par un système d'inégalités linéaires.

Dans cette thèse nous étudions le problème de l'analyse structurelle pour les systèmes dits algébro-différentiels conditionnels. Ces systèmes sont une généralisation des systèmes algébro-différentiels. Ils peuvent comporter certaines conditions permettant de modéliser des changements du système, par exemple un interrupteur dans un circuit électrique. Dans ce manuscrit, nous nous intéressons à ce problème. Nous examinons sa complexité algorithmique. Nous présentons des formulations mathématiques et en termes de graphes. Nous présentons aussi une étude polyédrale pour ce problème et nous

développons un algorithme de coupes et branchements pour le résoudre. Nous étudions aussi le problème de parallélisation des systèmes algébro-différentiels, permettant sa résolution sur plusieurs processeurs. Nous proposons des formulations mathématiques et nous donnons quelques résultats théoriques liés au polyèdre des solutions.

Cette thèse a été préparée dans le cadre du projet ANR-06-TLOG-26-01 PARADE dont les partenaires sont LMS-Imagine, Continental, LAGEP (Université Lyon1), ICJ (Université Lyon 1), INRIA Metalau, LAMSADE (Université Paris-Dauphine), LIMOS (Université Blaise Pascal, Clermont II).

Le manuscrit est organisé comme suit. Dans le premier chapitre nous introduisons quelques notions de base de graphes, polyèdres et de complexité et les notations qui seront utiles tout au long du manuscrit. Le chapitre 2 présente les systèmes algébro-différentiels, les systèmes algébro-différentiels conditionnels et les problèmes de l'analyse structurelle associés. Dans ce chapitre, est aussi donné un état de l'art pour le problème de l'analyse structurelle et des problèmes connexes. Le chapitre 3 présente une modélisation du problème de l'analyse structurelle pour les systèmes algébro-différentiels conditionnels en termes de graphes. Nous montrons que le problème se ramène à un problème de couplage que nous avons nommé : problème du sous-graphe sans couplage parfait. La complexité de ce problème est analysée et une seconde modélisation en termes de graphes est donnée ainsi que des formulations en programmes linéaires en nombres entiers. Nous étudions, dans le chapitre 4 le polytope des solutions du problème du sous-graphe sans couplage parfait. Dans le chapitre 5, nous proposons un algorithme de coupes et branchements pour résoudre ce problème. Le chapitre 6 concerne une extension de ces systèmes aux systèmes algébro-différentiels conditionnels imbriqués. Enfin, dans le chapitre 7 nous nous intéressons au problème de parallélisation des systèmes algébro-différentiels.

Chapitre 1

Notions préliminaires

Dans ce chapitre, nous donnons quelques notions de base sur les polyèdres combinatoires et la théorie de la complexité. Nous effectuons ensuite un bref rappel sur les méthodes de coupes et branchements. Nous terminons ce chapitre par les notations et définitions qui seront utilisées tout au long de ce manuscrit.

1.1 Problèmes d'optimisation combinatoire

L'*Optimisation combinatoire* est une des branches de l'informatique et des mathématiques appliquées. Elle concerne les problèmes pouvant se formuler de la façon suivante : soit $E = \{e_1, \dots, e_n\}$ un ensemble fini appelé *ensemble de base*, où chaque élément e_i possède un *poids* $c(e_i)$. Soit \mathcal{S} une famille de sous-ensembles de E . Si $S \in \mathcal{S}$, alors $c(S) = \sum_{e_i \in S} c(e_i)$ est le poids de S . Le problème consiste à déterminer un élément de \mathcal{S} , ayant le plus petit (ou le plus grand) poids. Un tel problème est appelé un *problème d'optimisation combinatoire*. L'ensemble \mathcal{S} est appelé l'*ensemble des solutions* du problème.

Le mot *combinatoire*, qui désigne la discipline des mathématiques concernée par les structures discrètes ou finies, évoque l'existence d'une structure sous-jacente discrète (généralement un graphe). L'ensemble de solutions \mathcal{S} est défini dans cette structure et peut avoir un nombre exponentiel d'éléments. Le mot *optimisation* signifie que l'on recherche le meilleur élément de l'ensemble de solutions \mathcal{S} .

De nombreux problèmes peuvent se formuler comme des problèmes d'optimisation combinatoire, par exemple le problème du sac à dos, du voyageur de commerce, de

conception de réseaux, de transport, de localisation... L'optimisation combinatoire se trouve au carrefour de la théorie des graphes, de la programmation linéaire et de la programmation linéaire en nombres entiers. Elle est très liée à la théorie de la complexité d'algorithmes.

1.2 Rappels sur la théorie de la complexité

La théorie de la complexité est basée sur les travaux d'Edmonds [22] et de Cook [15]. Elle permet de classer un problème donné parmi les problèmes faciles ou difficiles. Dans cette section, nous rappelons quelques éléments de base. La théorie de la NP-complétude est présentée en détail dans Garey et Johnson [32].

Un *problème* est une question générale possédant des paramètres dont la valeur n'est pas connue. Un problème est décrit en donnant : une description générale de tous les paramètres, et une énumération des propriétés que la solution doit satisfaire. Une *instance* d'un problème est obtenue en spécifiant la valeur de chaque paramètre du problème. Un *algorithme* de résolution d'un problème donné est une procédure, décomposable en opérations élémentaires, qui pour chaque instance du problème, produit une solution. La *taille* d'un problème reflète le nombre de données nécessaires pour décrire une instance.

Un algorithme est dit en $O(f(n))$ s'il existe un scalaire c et un entier n_0 tels que nombre d'opérations élémentaires nécessaires pour résoudre une instance de taille n au est plus $c \cdot f(n)$ pour tout $n \geq n_0$. Si f est une fonction polynomiale, alors l'algorithme est dit *polynomial*. Un problème appartient à la *classe* P s'il existe, pour toute instance du problème, un algorithme polynomial en la taille de l'instance permettant de résoudre le problème. Les problèmes de la classe P sont dits *faciles*.

Un *problème de décision* est un problème ayant deux réponses possibles : *oui* ou *non*. Soient \mathcal{P} un problème de décision et \mathcal{I} les instances de ce problème pour lesquelles la réponse est oui. \mathcal{P} appartient à la *classe* NP (Nondeterministic Polynomial) s'il existe un algorithme polynomial qui permet de vérifier que la réponse est oui pour toute instance de \mathcal{I} . Il est clair que la classe P est contenue dans la classe NP. La différence entre P et NP n'a pas été prouvée, mais la conjecture est considérée comme hautement probable.

Nous distinguons également dans la classe NP, la classe des problèmes NP-complets. La NP-complétude s'appuie sur la notion de réduction polynomiale. Un problème de décision P_1 se réduit polynomialement en un problème de décision P_2 s'il existe une

fonction polynomiale f telle que, pour toute instance I de P_1 , la réponse est oui si et seulement si la réponse de $f(I)$ pour P_2 est oui. Nous noterons alors $P_1 \alpha P_2$. Un problème \mathcal{P} est *NP-complet*, s'il appartient à la classe NP et s'il existe un problème Q connu comme étant NP-complet tels que $Q \alpha P$. Cook a été le premier à montrer la NP-complétude d'un problème, celui de la satisfiabilité [15].

À tout problème d'optimisation combinatoire peut être associé un problème de décision. Tout problème d'optimisation combinatoire dont le problème de décision associé est NP-complet est dit *NP-difficile*.

1.3 Éléments de la théorie des polyèdres

Dans cette section, nous introduisons quelques définitions et propriétés de la théorie des polyèdres. Pour plus de détails, on peut se référer à Pulleyblank [74] et Schrijver [77].

Soit $x \in \mathbb{R}^n$. On dit que x est une *combinaison linéaire* des points $x_1, \dots, x_k \in \mathbb{R}^n$ s'il existe k scalaires $\lambda_1, \lambda_2, \dots, \lambda_k \in \mathbb{R}$ tels que $x = \sum_{i=1}^k \lambda_i x_i$. Si, de plus, $\sum_{i=1}^k \lambda_i = 1$ alors on dit que x est une *combinaison affine* de ces points. De même, si $\lambda_i \geq 0 \forall i \in \{1, \dots, k\}$ avec $\sum_{i=1}^k \lambda_i = 1$, alors on dit que x est une *combinaison convexe* de ces points.

Des points $x_1, \dots, x_k \in \mathbb{R}^n$ sont dits *linéairement indépendants* (resp. *affinement indépendants*) si le système

$$\sum_{i=1}^k \lambda_i x_i = 0$$

$$\text{(resp. } \sum_{i=1}^k \lambda_i x_i = 0 \text{ et } \sum_{i=1}^k \lambda_i = 0 \text{)}$$

admet une solution unique, $\lambda_i = 0, \forall i = 1, \dots, k$.

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

Un *polyèdre* P est un ensemble de points de \mathbb{R}^n engendré par l'intersection d'un nombre fini de demi-espaces de \mathbb{R}^n . D'une manière équivalente, P est l'ensemble des solutions d'un système d'inégalités linéaires, c'est-à-dire $P = \{x \in \mathbb{R}^n \mid Ax \leq b\}$, où A est une matrice à m lignes et n colonnes, et b un vecteur de m composantes. Nous pouvons alors dire que le système $Ax \leq b$ caractérise le polyèdre P .

Un *polytope* est un polyèdre borné. Ainsi, un polyèdre $P \subseteq \mathbb{R}^n$ est un polytope si et seulement s'il existe $l, u \in \mathbb{R}^n$ tels que $l \leq x \leq u$, pour tout $x \in P$. Par exemple, l'enveloppe convexe d'un nombre fini de points est un polytope.

Un polyèdre P de \mathbb{R}^n est de *dimension* d , notée $\dim(P)$, si le nombre maximum de points de P affinement indépendants est égal à $d + 1$. Un polyèdre est de *pleine dimension* si $\dim(P) = n$.

Un point $x \in \mathbb{R}^n$ est un *point extrême* d'un polyèdre P s'il ne peut s'écrire comme combinaison convexe d'autres points de P .

Une inégalité (ou *contrainte*) $a^T x \leq \alpha$, où a est un vecteur à n composantes et α un scalaire, est dite *valide* pour un polyèdre P si elle est vérifiée par tous les points de P , soit $P \subseteq \{x \in \mathbb{R}^n \mid a^T x \leq \alpha\}$. Étant donné un point $x^* \in P$, nous dirons que l'inégalité $a^T x \leq \alpha$ est *serrée* (resp. *lâche*) pour x^* si $a^T x^* = \alpha$ (resp. $a^T x^* < \alpha$). Soit $a^T x \leq \alpha$ une contrainte valide pour le polyèdre P . Alors le sous-ensemble $F \subseteq P$ défini par $F = \{x \in P \mid a^T x = \alpha\}$ est une *face* de P . On dit alors que la face F est définie (ou induite) par la contrainte $a^T x \leq \alpha$. Si, de plus, $F \neq \emptyset$ et $F \neq P$ alors F est une *face propre*. Si F est propre et que $\dim(F) = \dim(P) - 1$, alors F est une *facette*.

Une inégalité est dite *essentielle* pour un polyèdre P si elle induit une facette pour P . Une inégalité est dite *redondante* dans un système $Ax \leq b$ définissant un polyèdre P , si le sous-système obtenu à partir de $Ax \leq b$ en supprimant cette inégalité définit le même polyèdre P .

1.4 Approche polyédrale, méthode de coupes et branchements

Soient \mathcal{P} un problème d'optimisation combinatoire et \mathcal{S} l'ensemble de ses solutions. Le problème \mathcal{P} s'écrit alors

$$\min \{cx \mid x \in \mathcal{S}\}$$

où c est un vecteur coût associé aux variables du problème. Considérons l'enveloppe convexe $\text{conv}(\mathcal{S})$ des solutions de \mathcal{P} . Le problème \mathcal{P} est équivalent au programme linéaire

$$\min \{cx \mid x \in \text{conv}(\mathcal{S})\}$$

L'*approche polyédrale*, introduite par Edmonds [23] dans le cadre du problème de couplages, consiste à décrire le polyèdre $\text{conv}(\mathcal{S})$ par un système d'inégalités linéaires,

ramenant ainsi le problème \mathcal{P} à la résolution d'un programme linéaire. Ceci permet donc de résoudre le problème \mathcal{P} en utilisant les algorithmes de la programmation linéaire [16, 44, 49]. Rappelons que la résolution d'un programme linéaire peut être effectuée en temps polynomial [44, 49].

Une description complète du polyèdre peut cependant comporter un nombre exponentiel d'inégalités. Le problème d'optimisation sur le polytope $\text{conv}(\mathcal{S})$ ne peut donc être résolu comme un programme linéaire ayant explicitement toutes ses contraintes. Cependant, un nombre réduit de ces inégalités peut être suffisant pour résoudre le problème à l'aide d'une *méthode de coupes*. Cette méthode s'appuie sur le problème suivant, appelé *problème de séparation*.

Problème Soit \mathcal{C} une classe d'inégalités valides pour $\text{conv}(\mathcal{S})$. Le problème de séparation associé à \mathcal{C} consiste, étant donné un point $x \in \mathbb{R}^n$, à décider si x satisfait toutes les inégalités de \mathcal{C} et sinon, à trouver une inégalité de \mathcal{C} violée par x .

Le problème de séparation est un des points essentiels de l'approche polyédrale. En effet, Grötschel, Losvász et Schrijver [36] ont montré qu'un problème d'optimisation combinatoire sur un ensemble de contraintes \mathcal{C} peut être résolu en temps polynomial si et seulement si le problème de séparation associé à \mathcal{C} peut être résolu en temps polynomial. Ainsi, une méthode de coupes permet de résoudre un problème d'optimisation combinatoire en temps polynomial si l'on sait résoudre en temps polynomial le problème de séparation pour un système d'inégalités caractérisant le polyèdre associé. En effet, la méthode de coupes permet de résoudre un problème \mathcal{P} comme une séquence de programmes linéaires, chacun contenant un nombre raisonnable de contraintes. Pour cela, la méthode commence par résoudre un programme linéaire contenant un sous-ensemble de contraintes de $\text{conv}(\mathcal{S})$. Notons par x^* la solution optimale obtenue. En appliquant les problèmes de séparation associés aux différentes classes d'inégalités définissant $\text{conv}(\mathcal{S})$, il est possible de savoir si x^* vérifie toutes les contraintes de $\text{conv}(\mathcal{S})$, *i.e.*, si $x^* \in \text{conv}(\mathcal{S})$. Si tel est le cas, alors x^* est la solution du problème. Autrement, les contraintes violées par x^* générées lors de l'application des problèmes de séparation sont ajoutées au programme linéaire. Ce processus de résolution est répété jusqu'à ce que la solution optimale appartienne à $\text{conv}(\mathcal{S})$.

Cependant, il est généralement difficile d'obtenir la caractérisation complète pour les problèmes NP-difficiles. De plus, le problème de séparation associé à certaines classes d'inégalités peut lui-même être NP-difficile. On ne peut dans ce cas obtenir que des techniques de séparation approchées. Ainsi, une méthode de coupes seule peut ne fournir que des solutions non réalisables (fractionnaires). Dans ce cas, on exécute une étape de branchement qui consiste à diviser le problème en plusieurs sous-problèmes tel que

l'ensemble des solutions réalisables des sous-problèmes forme un recouvrement (idéalement une partition) des solutions du problème. Ce branchement peut, par exemple, se faire en choisissant une variable fractionnaire x_i et à considérer plusieurs (autant que le nombre de valeurs possibles pour x_i) sous-problèmes du problème courant en fixant dans chacun x_i à une de ses valeurs permises. On applique alors la méthode de coupes pour chacun des sous-problèmes. Ce processus continue jusqu'à l'obtention d'une solution optimale. Cette combinaison de la méthode de branchement et d'une méthode de coupes au niveau de chaque nœud de l'arbre est appelée *méthodes de coupes et branchements*.

Cette approche est largement utilisée pour les problèmes d'optimisation combinatoire.

1.5 Notations et définitions

Nous donnons dans cette section certaines définitions et notions préliminaires de la théorie des graphes.

Un *graphe* non orienté est noté $G = (V, E)$ où V est l'ensemble des sommets et E l'ensemble des arêtes.

Si e est une arête reliant deux sommets u et v , alors u et v seront appelés les *extrémités* de e , et nous écrirons $e = uv$ ou $e = \{u, v\}$. Si u est une extrémité de e , alors u (resp. e) est dit *incident* à e (resp. u). De même, deux sommets u et v formant une arête sont dits *adjacents*.

Si $F \subseteq E$ est un sous-ensemble d'arêtes, alors $V(F)$ représente l'ensemble des extrémités des arêtes de F . Si $W \subseteq V$ est un sous-ensemble de sommets, alors $E(W)$ denote l'ensemble des arêtes ayant leurs deux extrémités dans W .

Un *sous-graphe* $H = (U, F)$ de G est un graphe tel que $U \subseteq V$ et $F \subset E$. Un sous-graphe $H = (U, F)$ de G est dit *couvrant* si $U = V$. Soit $W \subseteq V$, $H = (W, E(W))$ est dit sous-graphe de G *induit* par W et sera noté par $G(W)$.

Si $F \subset E$ (resp. $W \subset V$), on notera par $G \setminus F$ (resp. $G \setminus W$) le graphe obtenu à partir de G en supprimant les arêtes de F (resp. les sommets de W et les arêtes incidentes à W). Si F (resp. W) est réduit à une seule arête e (resp. un seul sommet v), nous écrirons $G \setminus e$ (resp. $G \setminus v$).

Soit $W \subseteq V$, $\emptyset \neq W \neq V$, un sous-ensemble de sommets de V . L'ensemble des arêtes ayant une extrémité dans W et l'autre dans $V \setminus W$ est appelé *coupe* et noté $\delta(W)$. En

posant $\overline{W} = V \setminus W$, nous avons $\delta(W) = \delta(\overline{W})$. Si W est réduit à un seul sommet v , nous écrirons $\delta(v)$. La cardinalité de la coupe $\delta(W)$ d'un sous-ensemble W est appelée *degré* de W et notée $\deg(W)$.

Étant donnés W et W' deux sous-ensembles disjoints de V , alors $[W, W']$ représente l'ensemble des arêtes de G qui ont une extrémité dans W et l'autre dans W' .

Si $\{V_1, \dots, V_p\}$, $p \geq 2$, est une partition de V , alors $\delta(V_1, \dots, V_p)$ est l'ensemble des arêtes $e = uv$ telles que $u \in V_i$, $v \in V_j$ et $i \neq j$.

Soient u et v deux sommets de V . Une *chaîne* P entre u et v est une séquence alternée de sommets et d'arêtes $(v_0, e_1, v_1, e_2, v_2, \dots, v_{k-1}, e_k, v_k)$ où $v_0 = u$, $v_k = v$, $e_i = v_{i-1}v_i$ pour $i = 1, \dots, k$. P est dite *élémentaire* si elle passe au plus une fois par le même sommet (à l'exception de v_0 et v_k si ces derniers représentent le même sommet de G). Une chaîne élémentaire est donc totalement identifiée par son ensemble d'arêtes.

Deux chaînes entre u et v sont dites *arête-disjointes* (resp. *sommet-disjointes*) s'il n'existe pas d'arête (resp. de sommet différent de u et v) apparaissant dans les deux chaînes.

Un ensemble de sommets ou d'arêtes S dans un graphe est dit *maximal* respectant la propriété *PROP*, s'il vérifie la propriété *PROP* et s'il n'existe pas d'ensemble contenant S et vérifiant cette propriété. L'ensemble S est dit *maximum* respectant la propriété *PROP* si, parmi tous les sous-ensembles de G respectant la propriété *PROP*, S est celui qui à la cardinalité la plus grande. Les termes *minimal* et *minimum* sont définis de manières analogues.

Étant donné un graphe $G = (V, E)$, un *stable* dans G est un ensemble de sommets deux à deux disjoints. Le problème du stable de cardinalité maximum consiste à trouver un stable dont le nombre de sommets est maximum [78]. La cardinalité du stable maximum est notée par $\alpha(G)$.

Étant donné un graphe $G = (V, E)$, une *couverture* de sommets dans G est un ensemble de sommets contenant au moins une extrémité de chaque arête. Le problème de couverture de sommets minimum consiste à trouver une couverture dont le nombre de sommets est minimum [78]. La cardinalité de la couverture de sommets minimum est notée par $\tau(G)$.

Étant donné un graphe $G = (V, E)$, une *couverture* d'arêtes dans G est un ensemble d'arêtes incident à tous les sommets de G . Le problème de couverture d'arêtes minimum consiste à trouver une couverture dont le nombre d'arêtes est minimum [78]. La cardinalité de la couverture d'arêtes minimum est notée par $\rho(G)$.

Etant donné un graphe $G = (V, E)$, un *couplage* dans G est un ensemble d'arêtes deux à deux disjointes. Le problème du couplage de cardinalité maximum consiste à trouver un couplage dont le nombre d'arêtes est maximum [60],[23]. La cardinalité du couplage maximum de G est notée par $\nu(G)$. Un couplage est dit *parfait* si tous les sommets du graphe appartiennent à des arêtes du couplage.

Un graphe est dit *biparti* s'il existe une partition de son ensemble de sommets en deux sous-ensembles U et V telle que chaque arête ait une extrémité dans U et l'autre dans V . En d'autres termes, un graphe est biparti si l'on peut regrouper les sommets en deux groupes distincts sans lien entre les sommets de chaque groupe.

Un graphe est dit *triparti* s'il existe une partition de son ensemble de sommets en trois sous-ensembles V_1 , V_2 et V_3 telle que chaque arête ait soit une extrémité dans V_1 et l'autre dans V_2 soit une extrémité dans V_2 et l'autre dans V_3 soit une extrémité dans V_1 et l'autre dans V_3 . En d'autres termes, un graphe est triparti si l'on peut regrouper les sommets en trois groupes distincts sans lien entre les sommets de chaque groupe.

Un *graphe orienté* est noté $D = (V, A)$ où V est l'*ensemble de sommets* et A l'*ensemble des arcs*.

Si $a \in A$ est un arc reliant un sommet u à un sommet v , alors u sera appelé *extrémité initiale* et v *extrémité terminale* et nous écrirons $a = (u, v)$. On dit que a est un *arc sortant* de u et un *arc entrant* de v . Les sommets u et v sont appelés *extrémités* de a . Si v est une extrémité (initiale ou terminale) de a , alors v (resp. a) est dit *incident* à a (resp. v).

Si $B \subseteq A$ est un sous-ensemble d'arcs, alors $V(B)$ représente l'ensemble des extrémités des arcs de B . Si $W \subseteq V$ est un sous-ensemble de sommets, $A(W)$ représente l'ensemble des arcs ayant leurs deux extrémités dans W .

Un *sous-graphe* $H = (U, F)$ de D est un graphe tel que $U \subseteq V$ et $F \subset A$. Un sous-graphe $H = (U, F)$ de D est dit *couvrant* si $U = V$.

Si $F \subset A$ (resp. $W \subset V$), on notera par $D \setminus F$ (resp. $D \setminus W$) le graphe obtenu à partir de D en supprimant les arcs de F (resp. les sommets de W et les arêtes incidentes à W). Si F (resp. W) est réduit à un seul arc a (resp. un seul sommet v), nous écrirons $D \setminus a$ (resp. $D \setminus v$).

Soit $W \subseteq V$, $\emptyset \neq W \neq V$, un sous-ensemble de sommets de V . L'ensemble des arcs ayant leur extrémité initiale dans W et leur extrémité terminale dans $V \setminus W$ est appelé *coupe sortante* et noté $\delta^{\text{out}}(W)$. La cardinalité de la coupe sortante $\delta^{\text{out}}(W)$

d'un sous-ensemble W est appelée *degré sortant* de W et notée $\deg^{\text{out}}(W)$. Si $u \in W$ et $v \in V \setminus W$, alors la coupe sortante est aussi appelée *uv-coupe sortante*. Si W est réduit à un seul sommet v , nous écrirons respectivement $\delta^{\text{out}}(v)$ et $\deg^{\text{out}}(v)$ au lieu de $\delta^{\text{out}}(\{v\})$ et $\deg^{\text{out}}(\{v\})$.

L'ensemble des arcs ayant l'extrémité terminale dans W et l'extrémité initiale dans $V \setminus W$ est appelé *coupe entrante* et noté $\delta^{\text{in}}(W)$. La cardinalité de la coupe entrante $\delta^{\text{in}}(W)$ d'un sous-ensemble W est appelée *degré entrant* de W et notée $\deg^{\text{in}}(W)$. Si $u \in W$ et $v \in V \setminus W$, alors la coupe entrante est aussi appelée *uv-coupe entrante*. Si W est réduit à un seul sommet v , nous écrirons respectivement $\delta^{\text{in}}(v)$ et $\deg^{\text{in}}(v)$ au lieu de $\delta^{\text{in}}(\{v\})$ et $\deg^{\text{in}}(\{v\})$.

La *coupe* d'un ensemble $W \subseteq V$, $\emptyset \neq W \neq V$, est notée $\delta(W)$ et correspond à l'union des arcs de la coupe entrante et de la coupe sortante, *i.e.*, $\delta(W) = \delta^{\text{out}}(W) \cup \delta^{\text{in}}(W)$. La cardinalité de la coupe est appelée *degré* de W et notée $\deg(W)$. Si $u \in W$ et $v \in V \setminus W$, alors la coupe est aussi appelée *uv-coupe*. Si W est réduit à un seul sommet v , nous écrirons respectivement $\delta(v)$ et $\deg(v)$ au lieu de $\delta(\{v\})$ et $\deg(\{v\})$.

Soient D un graphe orienté et u et v deux sommets de V . Un *chemin* P de u à v dans D est une séquence d'arcs $((u_1, v_1), (u_2, v_2), \dots, (u_k, v_k))$ avec $u_1 = u$, $v_k = v$, $v_i = u_{i+1}$ pour tout $i = 1, 2, \dots, k-1$ et $(u_i, v_i) \in A$ pour tout $i = 1, 2, \dots, k$. Les sommets u et v sont respectivement appelés *sommet initial* et *sommet terminal* de P . Le nombre d'arcs de P est appelé *cardinalité* de P et noté $|P|$. P est appelé *circuit* si $u = v$. P est dit *élémentaire* si chaque sommet a au plus un arc entrant et un arc sortant parmi les arcs de P . Si P est un chemin élémentaire, alors P est défini par l'ensemble d'arcs apparaissant dans P . Si P est un circuit élémentaire, alors P est défini par l'ensemble d'arcs apparaissant dans P et par le sommet initial de P . Soit $Q = ((s_1, t_1), (s_2, t_2), \dots, (s_l, t_l))$ un chemin dans D . Q est dit *sous-chemin* de P s'il existe $j \in \{1, 2, \dots, k+1-l\}$ tel que $(s_i, t_i) = (u_{i+j}, v_{i+j})$ pour tout $i = 1, 2, \dots, l$.

Deux chemins entre u et v sont dits *arête-disjoints* (resp. *sommet-disjoints*) s'il n'existe pas d'arête (resp. de sommet différent de u et v) apparaissant dans les deux chemins.

1.6 Couplages et graphes bipartis

Les problèmes de couplages sont fondamentaux en optimisation combinatoire et en théorie des graphes. Les couplages permettent de modéliser des problèmes dans de nombreux domaines aussi bien théoriques (le couplage peut être utilisé comme routine

pour la résolution d'autres problèmes de graphes comme la recherche en profondeur) et pratiques (utilisés pour modéliser des problèmes de reconnaissance de formes, d'affectation, de mariage...).

Dans cette section, nous allons nous intéresser aux propriétés des couplages dans les graphes et plus particulièrement les graphes bipartis. Nous présentons aussi quelques algorithmes.

Dans les graphes, bipartis ou non, il existe plusieurs résultats permettant de lier les valeurs $\tau(G)$, $\rho(G)$, $\alpha(G)$ et $\nu(G)$. Nous allons présenter deux relations qui ont été montrées par Gallai en 1959.

Le premier lemme montre la relation entre la cardinalité du stable maximum et la cardinalité de la couverture de sommets minimum.

Lemme 1.1 [30] *Soit $G = (V, E)$ un graphe, $\alpha(G) + \tau(G) = |V|$.* □

Le deuxième lemme montre la relation entre la cardinalité du couplage maximum et la cardinalité de la couverture d'arêtes minimum.

Lemme 1.2 [30] *Soit $G = (V, E)$ un graphe ne contenant pas de sommet isolé, $\nu(G) + \rho(G) = |V|$.* □

Dans le cas des graphes bipartis, il existe une relation forte entre le couplage maximum et la couverture de sommets minimum. Cette relation a été montrée par König en 1931.

Théorème 1.1 [50] *(Théorème de König). Soit $G = (V, E)$ un graphe biparti, alors $\tau(G) = \nu(G)$.* □

On peut déduire du théorème et des lemmes précédents la relation suivante :

Corollaire 1.3 *Soit $G = (V, E)$ un graphe biparti, alors $\nu(G) + \alpha(G) = |V|$.*

Le théorème de König a donné lieu à plusieurs travaux qui ont abouti sur d'autres théorèmes équivalents. L'un des théorèmes les plus utilisés est donné par Hall en 1935.

Théorème 1.2 [38](Théorème de Hall). *Un graphe biparti $G = (U \cup V, E)$ admet un couplage parfait si et seulement si pour tout sous-ensemble X de U (resp. de V), le nombre de sommets de V (resp. de U) adjacents aux sommets de X est supérieur ou égal à la cardinalité de X .* \square

Nous allons désormais introduire les définitions permettant de donner quelques propriétés sur les couplages. Soient $G = (V, E)$ un graphe et M un couplage de G . Une chaîne élémentaire Γ est dite *M -alternante* si les arêtes de Γ alternent entre des arêtes appartenant et n'appartenant pas au couplage M . Notons que Γ peut commencer par une arête appartenant à M ou une arête n'appartenant pas à M . Si Γ commence et finit par une arête n'appartenant pas au couplage M , alors Γ est une chaîne *M -augmentante*. L'existence d'une chaîne M -augmentante dans G permet de savoir si un couplage est ou non maximum. De plus les chaînes augmentantes sont utilisées dans certains algorithmes permettant de trouver un couplage maximum. Les chaînes augmentantes ont été proposées par Petersen [69] en 1891.

Théorème 1.3 [60] *Soient un graphe $G = (V, E)$ un graphe et M un couplage de G , M est un couplage maximum si et seulement s'il n'existe pas de chaîne M -augmentante dans G .* \square

Il est clair que si G contient une chaîne Γ M -augmentante, il est possible d'obtenir un couplage M' plus grand, en supprimant les arêtes de $\Gamma \cap M$ et ajoutant les arêtes de $\Gamma \setminus M$.

Théorème 1.4 [60] *Les problèmes de couplages maximums, couplages de poids maximum et couplages parfaits de poids maximum (resp. minimum) peuvent se résoudre en temps polynomial.* \square

Le problème du couplage maximum a longuement été étudié. König [50] en 1931 a proposé un algorithme en $O(nm)$ en utilisant n chaînes augmentantes. Soit M un couplage quelconque (qui peut être vide). L'idée de l'algorithme est de trouver une chaîne Γ M -augmentante puis en remplaçant dans le couplage M les arêtes de $M \cap \Gamma$ par les arêtes de $\Gamma \setminus M$, on obtient un couplage M' tel que $|M'| = |M| + 1$. Il est possible de trouver une chaîne augmentante en effectuant un parcours en profondeur et en autorisant uniquement un parcours qui alterne entre une arête appartenant à M et une arête n'appartenant pas à M . Hopcroft et Karp [39] en 1971 ont proposé un algorithme en $O(n^{1/2}m)$ pour résoudre le problème de couplages dans les graphes

bipartis. L'idée est de transformé le graphe d'origine en un graphe orienté. Le problème de couplages se ramène alors à un problème de flots.

Le problème de flot maximum consiste à trouver un flot réalisable depuis une source unique vers un puits unique qui soit de cardinalité maximum. Ce problème peut se résoudre en temps polynomial [25], il existe de nombreux algorithmes permettant de résoudre rapidement ce problème [26], [29].

Soit $G = (U \cup V, E)$ un graphe biparti. Considérons le graphe orienté $D = ((s \cup U) \cup (V \cup t), A)$ où A est formé de la manière suivante

- 1) pour tout $uv \in E$ on ajoute l'arc (u, v) avec une capacité maximale de ∞ ,
- 2) pour tout $u \in U$ on ajoute l'arc (s, u) avec une capacité maximale de 1,
- 3) pour tout $v \in V$ on ajoute l'arc (v, t) avec une capacité maximale de 1.

Soit f le flot maximum dans D . Alors le nombre maximum de chemins sommets disjoints $s - t$ donné par le flot f dans D est égal à $\nu(G)$ [26].

La figure 1.1 montre un exemple de la transformation d'un graphe G en un graphe D .

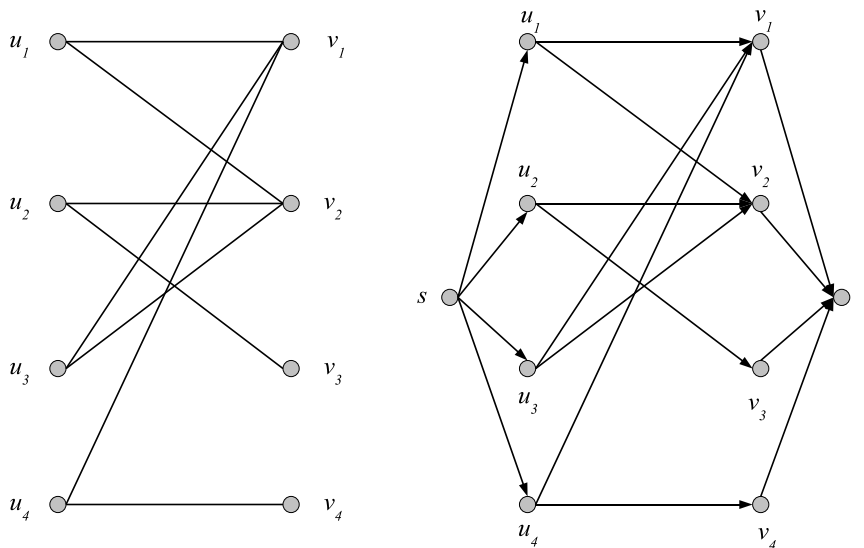


FIGURE 1.1 – Graphe G et le graphe D associé.

Edmonds [23] proposa le premier algorithme polynomial pour résoudre le problème

du couplage parfait de poids maximum (resp. minimum). Il permet de résoudre le problème, dans le cas général (pas seulement pour les graphes bipartis), en $O(n^2m)$. Il est basé sur une formulation en programme linéaire. Soient $G = (V, E)$ un graphe et \mathcal{O} l'ensemble des sous-ensembles de sommets de V de cardinalités impaires contenant au moins 3 sommets. Considérons un vecteur $(x_e : e \in E)$ et un ensemble $H \subseteq E$, soit $x(H)$ donné par $\sum(x_e : e \in H)$. Un vecteur d'incidence correspondant à un couplage parfait dans G satisfait le système linéaire

$$x(\delta(v)) = 1 \quad \forall v \in V, \quad (1.1)$$

$$x_e \geq 0 \quad \forall e \in E, \quad (1.2)$$

$$x(\delta(S)) \leq \frac{|S| - 1}{2} \quad \forall S \in \mathcal{O}. \quad (1.3)$$

Le couplage parfait de poids maximum (resp. minimum) peut être résolu par le programme linéaire

$$\max \text{ (resp. } \min) \{wx \mid x \text{ satisfait (1.1)-(1.3)}\}, \quad (1.4)$$

où w est le vecteur correspondant aux poids des arêtes.

Cet algorithme a été amélioré par Galil, Micali et Gabow en 1982 [28] permettant de résoudre le problème du couplage parfait de poids maximum (resp. minimum) en $O(nm \log(n))$. C'est cette version de l'algorithme qui est la plus souvent utilisée, comme par exemple dans la bibliothèque LEMON GRAPH en C++ [4]. Ball et Derigs ont proposé un algorithme similaire en 1983 [9].

Nous pouvons remarquer que dans le cas des graphes bipartis trouver un couplage parfait de coût minimum se ramène à trouver un st -flot de valeur n et de coût minimum. Un st -flot est un flot, depuis la source s vers un puits t , vérifiant la conservation du flot pour tous les sommets, où on ajoute au graphe un arc de retour entre le sommet t et le sommet s de capacité infinie. La recherche d'un st -flot de valeur n et de coût minimum se résout en temps polynomial, il existe de nombreux algorithmes permettant de résoudre rapidement ce problème [27], [35].

Chapitre 2

Systèmes algébro-différentiels

Un système algébro-différentiel est un ensemble d'équations différentielles. Une équation différentielle est une relation entre une ou plusieurs fonctions inconnues et leurs dérivées. La résolution d'un système algébro-différentiel consiste alors à déterminer un ensemble de fonctions vérifiant le système. On distingue les systèmes différentiels ordinaires qui sont les systèmes algébro-différentiels les plus simples. Dans les systèmes différentiels ordinaires, chaque fonction dérivée est exprimée de façon explicite par des fonctions non dérivées.

Les systèmes algébro-différentiels remontent à l'invention du calcul différentiel et intégral, dans les années 1670-1680. À l'origine, ces systèmes servaient à résoudre des problèmes géométriques, comme la détermination d'une courbe dont les tangentes sont soumises à une condition donnée. C'est seulement vers 1730 que le mathématicien Euler a commencé à les utiliser pour traiter des problèmes de dynamique. Aujourd'hui, elles apparaissent dans presque tous les domaines de la science et de la technique, par exemple les mathématiques, la physique, la chimie, la biologie et l'économie. Elles servent à traduire les lois qui régissent la variation des valeurs, par exemple un condensateur électrique ou la position d'une navette spatiale.

La résolution des systèmes algébro-différentiels peut se faire de deux manières. La résolution analytique fournit une solution exacte en donnant un ensemble de fonctions satisfaisant les équations différentielles. Cette méthode permet de résoudre uniquement des systèmes simples. La résolution numérique donne, sur un intervalle, une approximation de la solution par opérations successives. Cette résolution est souvent longue. Il est alors intéressant de savoir à l'avance si un système algébro-différentiel peut ou non être résolu.

Le problème de l'analyse structurelle permet de vérifier si un système algébro-différentiel satisfait une condition nécessaire pour être solvable, c'est-à-dire pour qu'il possède une unique solution. Si cette condition n'est pas vérifiée, il est alors inutile d'essayer de résoudre le système avec des méthodes numériques.

Dans ce chapitre, nous donnons une vue d'ensemble et les définitions nécessaires à la compréhension du problème de l'analyse structurelle des systèmes algébro-différentiels. Nous présentons aussi les cas particuliers des systèmes différentiels ordinaires et des systèmes algébro-différentiels semi-explicites. Dans la section 2.1, nous donnons un aperçu des systèmes différentiels ordinaires et des systèmes algébro-différentiels. La section 2.2 donne quelques idées permettant de résoudre ces systèmes. Nous définissons, dans la section 2.3, le problème de l'analyse structurelle. Nous donnons un état de l'art sur le problème de l'analyse structurelle dans la section 2.4. Nous présentons ensuite les équations différentielles conditionnelles permettant de généraliser les systèmes algébro-différentiels. Enfin nous donnons une généralisation du problème de l'analyse structurelle pour les systèmes comportant des équations différentielles conditionnelles.

2.1 Systèmes algébro-différentiels

Dans cette section, nous présentons d'abord les systèmes différentiels ordinaires. Nous donnons ensuite la forme générale, en présentant les systèmes algébro-différentiels. Nous donnons aussi les définitions et notations associées à ces systèmes.

Étant donnée une fonction

$$\begin{aligned} y(t) &: \mathbb{R} \mapsto \mathbb{R}^m, \\ t &\mapsto (y_1(t), \dots, y_m(t)), \end{aligned}$$

la dérivée de $y(t)$ en un point t_0 mesure le taux de croissance de $y(t)$ en ce point. La dérivée de $y(t)$ est notée

$$\begin{aligned} \dot{y}(t) &: \mathbb{R} \mapsto \mathbb{R}^m, \\ t &\mapsto (\dot{y}_1(t), \dots, \dot{y}_m(t)). \end{aligned}$$

Pour simplifier les notations, on note $y = y(t)$, $\dot{y} = \dot{y}(t)$, $y_i = y_i(t)$ et $\dot{y}_i = \dot{y}_i(t)$, pour $i \in \{1, \dots, m\}$. Par abus de langage, une fonction $y_i(t)$ est aussi appelée variable.

Les systèmes algébro-différentiels ne considèrent que les dérivées des variables en fonction du temps. Nous ne nous intéressons pas aux dérivées partielles permettant de considérer les dérivées par rapport aux autres variables.

2.1.1 Systèmes différentiels ordinaires

Les systèmes différentiels ordinaires, aussi appelés dans la littérature équations différentielles ordinaires, sont fondamentaux et représentent une approche classique permettant de modéliser les systèmes dynamiques simples.

Etant donnée une fonction y définie sur un intervalle de $\mathbb{R} \mapsto \mathbb{R}^m$ et continûment dérivable, un *système différentiel ordinaire* (SDO) est un système de la forme :

$$\dot{y} = f(y, t)$$

où $f : \mathbb{R}^m \times \mathbb{R} \mapsto \mathbb{R}^m$ et t représente le temps. Les SDO sont caractérisés par le fait que chaque dérivée par rapport au temps \dot{y}_i , pour $i \in \{1, \dots, m\}$, est exprimée de manière explicite par y et t .

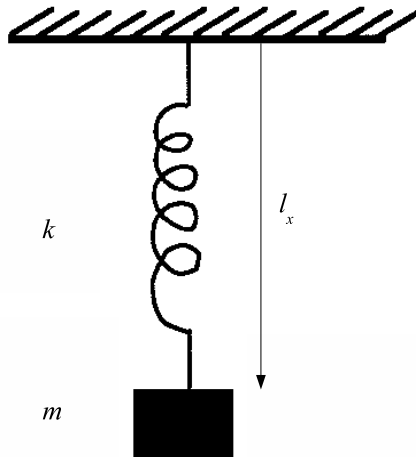


FIGURE 2.1 – Système masse-ressort

Nous présentons l'exemple d'un SDO donné dans [73]. Considérons le système masse-ressort de la figure 2.1. Dans ce système, un ressort de raideur k exerce une force sur une masse m . Cette force est exercée sur l'axe des ordonnées. Ce système peut-être modélisé par le SDO suivant.

$$\dot{v} = -\frac{kl_x}{m}, \quad (2.1)$$

$$\dot{l}_x = v, \quad (2.2)$$

où v représente la vitesse de la masse et l_x représente la position de la masse sur l'axe des ordonnées. Dans le système masse-ressort, l'évolution dans le temps indique que la force de réaction du ressort $[-kl_x]$ est exactement égale au produit de la masse par l'accélération \dot{v} . Les variables de notre SDO sont données par $y = (l_x, v)$.

Si une variable y_i est dérivable p fois, alors la *dérivée d'ordre p* de y_i s'appelle la dérivée p -ième de y_i , et on la note $y_i^{(p)}$. On note $\ddot{y}_i = y_i^{(2)}$. Un SDO peut contenir des variables dérivées d'ordre supérieur à 1. Dans ce cas, on dit qu'un SDO est d'ordre p , où p est l'ordre maximum des variables dérivées du système. Il est toujours possible de transformer un système différentiel d'ordre p en un système différentiel d'ordre 1. Il suffit pour chaque variable $y_i^{(k)}$, d'ajouter la variable v_i et l'équation différentielle $v_i = y_i^{(k-1)}$ puis on remplace $y_i^{(k)}$ par \dot{v}_i . Par exemple,

$$M\ddot{l}_x = A \Rightarrow M\dot{v} = A, \quad (2.3)$$

$$v = \dot{l}_x. \quad (2.4)$$

Nous pouvons donc considérer, sans perte de généralité, que tous les SDO sont d'ordre 1.

Par exemple, le système masse-ressort présenté ci-dessus peut aussi s'écrire de la manière suivante :

$$\ddot{l}_x = -\frac{kl_x}{m}, \quad (2.5)$$

où \ddot{l}_x est la dérivée d'ordre 2 de la variable l_x ($y = (l_x)$). Ce système est d'ordre 2.

Nous avons vu les systèmes différentiels ordinaires permettant de modéliser les systèmes dynamiques simples. Nous allons voir dans la section suivante les systèmes algébro-différentiels qui sont une généralisation des SDO [70].

2.1.2 Systèmes algébro-différentiels

Les systèmes algébro-différentiels décrivent les systèmes dynamiques, d'une manière plus générale que les SDO.

Étant donnée une fonction y définie sur un intervalle de $\mathbb{R} \mapsto \mathbb{R}^m$ et continûment dérivable, un *système algébro-différentiel* (SAD) est un système de la forme :

$$f(\dot{y}, y, t) = 0$$

où $f : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R} \mapsto \mathbb{R}^m$ et t représente le temps. Nous considérons sans perte de généralité uniquement les SAD d'ordre 1. La méthode permettant de transformer un SAD d'ordre p en SAD d'ordre 1 est la même que la méthode utilisée pour les SDO.

Considérons le système de deux masses M_1 et M_2 couplées par une barre fixe, de longueur l , donné par la figure 2.2. Ce système peut-être modélisé par le SAD suivant.

$$M_1 \dot{v}_1 = F, \quad (2.6)$$

$$M_2 \dot{v}_2 = -F + f(t), \quad (2.7)$$

$$\dot{l}_{x_1} = v_1, \quad (2.8)$$

$$\dot{l}_{x_2} = v_2, \quad (2.9)$$

$$l_{x_1} - l_{x_2} = l. \quad (2.10)$$

où l_{x_1} (resp. l_{x_2}) représente la position de la masse M_1 (resp. M_2) sur l'axe des abscisses, v_1 (resp. v_2) représente la vitesse de la masse M_1 (resp. M_2) sur l'axe des abscisses, F représente la force agissant entre les deux masses et $f(t)$ représente la force extérieure exercée sur la masse M_2 . Les variables sont $y = (l_{x_1}, l_{x_2}, v_1, v_2, F)$. L'équation (2.6) (resp. (2.7)) représente l'accélération sur l'axe des abscisses de la masse M_1 (resp. M_2) en fonction de la force exercée par la barre. Les équations (2.8) et (2.9) font le lien entre la vitesse et la position des masses sur l'axe des abscisses. L'équation (2.10) représente la distance, entre les deux masses, définie par la barre fixe.

Nous allons désormais présenter un cas particulier des SAD permettant de faire le lien entre les SDO et les équations algébriques.

Définition 2.1 *Une variable algébrique est une variable x n'apparaissant pas dans le système sous la forme dérivée \dot{x} .*

Il existe des SAD particuliers définis par un ensemble d'équations différentielles ordinaires et un ensemble d'équations algébriques, aussi appelée contraintes algébriques. Ces systèmes sont alors représentés comme suit :

$$\dot{y} = f(y, x, t)$$

$$0 = g(y, x, t)$$

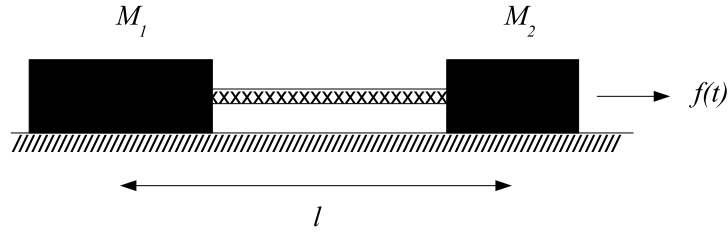


FIGURE 2.2 – Système de deux masses.

où x est le vecteur des variables algébriques et y est le vecteur des variables dérivées. Ces systèmes sont appelés *SAD semi-explicites*.

Par exemple, le système de deux masses présenté ci-dessus possède comme variable algébrique $x = (F)$ et comme variables dérivées $y = (l_{x_1}, l_{x_2}, v_1, v_2)$. Nous pouvons voir qu'il n'est pas possible d'expliciter \dot{F} en fonction de y , x et t car la variable F est une variable algébrique. On en déduit que ce système n'est pas un SDO. Par contre, on voit que ce système est un SAD semi-explicite, où les équations (2.6) et (2.7) sont réécrites de la manière suivante

$$\dot{v}_1 = \frac{F}{M_1}, \quad (2.11)$$

$$\dot{v}_2 = \frac{-F + f(t)}{M_2}. \quad (2.12)$$

Les équations (2.8), (2.9), (2.11) et (2.12) forment l'ensemble des équations différentielles ordinaires et l'équation (2.10) est une équation algébrique.

Dans la littérature, les SAD sont aussi représentés sous une forme matricielle. La forme matricielle permet de définir quelques propriétés sur la structure des SAD. Dans ce cas les SAD s'écrivent sous la forme :

$$E\dot{y} = Ay + g(t) \quad (2.13)$$

où E et A sont des matrices $m \times m$, $g(t)$ est le vecteur de fonction ne dépendant pas des variables y .

La forme matricielle permet de définir quelques propriétés sur la structure des SAD.

Définition 2.2 Une matrice A est dite régulière si elle est inversible. Autrement la matrice A est dite singulière.

Si la matrice E est régulière alors le système (2.13) peut se réécrire sous la forme d'un SDO

$$\dot{y} = \hat{A}y + \hat{g}(t) \quad (2.14)$$

où $\hat{A} = E^{-1}A$ et $\hat{g}(t) = E^{-1}g(t)$.

Si la matrice E est singulière alors le système (2.13) est dit *SAD propre*, car il ne peut pas se réécrire sous la forme d'un SDO.

Proposition 2.3 [31] *Une matrice carrée E est régulière si son déterminant est non nul. Dans le cas contraire, E est singulière.*

Le système du pendule présenté ci-dessus peut donc se réécrire sous la forme

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{v}_1 \\ \dot{v}_2 \\ \dot{l}_{x_1} \\ \dot{l}_{x_2} \\ \dot{F} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{1}{M_1} \\ 0 & 0 & 0 & 0 & -\frac{1}{M_2} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ l_{x_1} \\ l_{x_2} \\ F \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{f(t)}{M_2} \\ 0 \\ 0 \\ l \end{pmatrix}.$$

Nous allons voir, dans la prochaine section, les méthodes utilisées pour résoudre les SDO et les SAD.

2.2 Résolution des systèmes algébro-différentiels

Résoudre un système algébro-différentiel revient à trouver une fonction $y(t)$ qui vérifie les équations du système.

Par exemple, le système

$$\begin{aligned} \dot{u} &= v, \\ \dot{v} &= -u, \end{aligned}$$

possède une infinité de solutions, toutes de la forme

$$\begin{aligned}u(t) &= A \cos(t + T), \\v(t) &= -A' \sin(t + T),\end{aligned}$$

où A , A' et T sont des constantes. Les constantes A , A' et T dépendent de l'initialisation du système. Par exemple dans le cas du système masse-ressort de la figure 2.1, il est nécessaire de connaître la position initiale de la masse l_{x_0} et la vitesse initiale v_0 . Les valeurs permettant d'initialiser le système s'appellent valeurs initiales. Dans le cas général, il suffit d'ajouter au système les équations $y(t_0) = c_0$, où c_0 est un vecteur de constantes.

La résolution des systèmes différentiels peut se faire à l'aide de méthodes analytiques ou numériques. Les méthodes analytiques cherchent une expression mathématique exacte des solutions. Un exemple de résolution analytique est donné ci-dessus. Ces méthodes ne fonctionnent que dans un registre limité, de plus, proportionnellement, très peu de systèmes différentiels possèdent une solution analytique, exprimable avec les fonctions usuelles. La résolution numérique d'un SAD consiste à obtenir une approximation de la fonction $y(t)$, solution du SAD, sur un intervalle $[t_0, t_0 + T]$ en estimant la valeur de cette fonction en un nombre fini de points sur cet intervalle. Les méthodes numériques calculent une bonne approximation de la solution. En l'absence de solution analytique, il est alors nécessaire de résoudre les SAD avec des méthodes numériques.

La résolution numérique des SAD revient à trouver une approximation de la fonction $y(t)$, en certains points de l'ensemble $[t_0, t_0 + T]$, vérifiant le système :

$$\begin{aligned}0 &= f(\dot{y}, y, t), \\y(t_0) &= c_0.\end{aligned}$$

L'idée de la résolution numérique est de discrétiser de façon régulière $[t_0, t_0 + T]$ en N morceaux. Généralement, plus N est grand plus l'approximation de la solution est améliorée.

Dans les méthodes que nous présentons nous supposons que la valeur initiale $y(t_0) = c_0$ est donnée.

Dans la suite, nous nous intéressons uniquement aux méthodes numériques. Nous présentons dans un premier temps la méthode d'Euler explicite, nous verrons les limitations de cette méthode. Nous présentons ensuite la méthode d'Euler implicite permettant de corriger ces limitations.

Les méthodes d'Euler, l'une explicite et l'autre implicite, permettent d'approximer la valeur de la fonction au point $y(t_n)$ en fonction de la valeur de la fonction au point $y(t_{n-1})$. Ces méthodes permettent, de proche en proche, d'approximer l'ensemble des N points sur l'intervalle choisi $[t_0, t_0 + T]$. On note $t_n = t_0 + nh$, $n \in \{0, \dots, N\}$ une discrétisation régulière de $[t_0, t_0 + T]$ avec $h = T/N$ le pas de temps. Considérons la représentation matricielle d'un SAD, on a alors $E\dot{y}(t) = Ay(t) + g(t), \forall t \in [t_0, t_0 + T]$ ce qui entraîne $E\dot{y}(t_n) = Ay(t_n) + g(t_n), \forall n \in \{0, \dots, N\}$. Cette équation va nous permettre d'obtenir les schémas définissant les méthodes d'Euler. On pose $y^n = y(t_n)$ pour tout $n \in \{0, \dots, N\}$.

2.2.1 Méthode d'Euler explicite

La méthode d'Euler explicite se déduit directement de la formule des différences finies progressives [73]. Elle est donnée par le schéma

$$\begin{aligned} Ey^{n+1} &= (E + hA)y^n + hg(t_n), \forall n \in \{0, \dots, N - 1\}, \\ y^0 &= c_0. \end{aligned}$$

Si la matrice E est singulière, il est alors impossible de résoudre le système en utilisant cette méthode. En effet, pour calculer y_{n+1} il est nécessaire de reformuler le schéma de la manière suivante

$$\begin{aligned} y^{n+1} &= (I + hE^{-1}A)y^n + hE^{-1}g(t_n), \forall n \in \{0, \dots, N - 1\}, \\ y^0 &= c_0. \end{aligned}$$

La méthode d'Euler explicite ne peut donc pas résoudre les SAD propres puisque la matrice E est singulière pour de tels systèmes. De plus, cette méthode n'est pas stable, c'est-à-dire que l'accumulation des erreurs relatives à chaque approximation est importante [82].

Considérons le système

$$\dot{y}_1 + \dot{y}_2 = y_1 + y_2, \quad (2.15)$$

$$\dot{y}_1 + \dot{y}_2 = 2y_1 + 5, \quad (2.16)$$

$$y_1(t_0) = 2, \quad (2.17)$$

$$y_2(t_0) = 1. \quad (2.18)$$

Supposons que nous essayons de résoudre ce système en utilisant la méthode d'Euler explicite. Nous obtenons alors

$$y_1^{n+1} + y_2^{n+1} = y_1^n + y_2^n + h(y_1^n + y_2^n),$$

$$y_1^{n+1} + y_2^{n+1} = y_1^n + y_2^n + h(2y_1^n + 5).$$

Il est clair que quelques soient les valeurs de y_1^0 et y_2^0 , il n'est pas possible de trouver les valeurs de y_1^{n+1} et y_2^{n+1} . En effet, on ne peut pas exprimer y_1^{n+1} ou y_2^{n+1} uniquement en fonction de y_1^n et y_2^n .

Nous allons désormais présenter la méthode d'Euler implicite, cette méthode permet de résoudre les SAD et surtout elle permet une meilleure approximation que la méthode explicite.

2.2.2 Méthode d'Euler implicite

La méthode d'Euler implicite, quand à elle, est donnée par le schéma

$$(E - hA)y^{n+1} = Ey^n + hg(t_{n+1}), \forall n \in \{0, \dots, N-1\},$$

$$y^0 = c_0.$$

Ici le schéma est inversé. Au lieu de considérer y^{n+1} en fonction de y^n , on considère y^n en fonction de y^{n+1} .

Considérons à nouveau le système donné par les equations (2.15)-(2.18). En utilisant la méthode d'Euler implicite nous obtenons

$$\frac{y_1^{n+1} - y_1^n + y_2^{n+1} - y_2^n}{h} = y_1^{n+1} + y_2^{n+1},$$

$$\frac{y_1^{n+1} - y_1^n + y_2^{n+1} - y_2^n}{h} = 2y_1^{n+1} + 5,$$

qui est équivalent à

$$y_1^{n+1} = \frac{y_1^n + y_2^n}{2(1-h)} - \frac{5}{2},$$

$$y_2^{n+1} = \frac{y_1^n + y_2^n}{2(1-h)} + \frac{5}{2}.$$

Si $h \neq 1$ alors il est possible de résoudre le système.

Dans [80], les auteurs expliquent en détail l'avantage de la méthode d'Euler implicite par rapport à la méthode d'Euler explicite.

Il existe d'autres méthodes permettant de résoudre les SDO et SAD [11, 34, 61, 67, 73, 75, 81, 79].

Il peut être très difficile de résoudre directement un SAD avec une méthode de résolution numérique. L'index d'un système permet de déterminer la difficulté de résolution d'un SAD [21]. On appelle index d'un système le nombre de dérivations nécessaires permettant de transformer un SAD en SDO. Un SDO est un SAD d'index 0. Si l'index du système est élevé (supérieur à 3) alors les méthodes de résolutions donneront de très mauvaises approximations [10, 13].

Il est possible de réduire l'index d'un SAD afin de faciliter sa résolution. Nous allons illustrer la réduction d'index permettant de résoudre les SAD les plus difficiles. Considérons encore le système de deux masses M_1 et M_2 couplées par une barre fixe. Nous allons voir comment réduire ce SAD semi-explicite en un SAD semi-explicite d'index 1. La partie algébrique du système, donnée dans l'exemple, est constituée de l'équation (2.10) et lie les variables l_{x_1} et l_{x_2} en fonction d'une constante. Elle lie également les variables v_1 et v_2 de manière implicite. En dérivant l'équation (2.10) en fonction du temps, on voit le lien entre v_1 et v_2 .

$$l_{x_1} - l_{x_2} = l \Rightarrow \dot{l}_{x_1} - \dot{l}_{x_2} = 0 \quad (2.19)$$

$$\Leftrightarrow v_1 - v_2 = 0 \quad (2.20)$$

Afin d'obtenir un SAD semi-explicite d'index 1, il est nécessaire de dériver à nouveau cette équation :

$$v_1 - v_2 = 0 \Rightarrow \dot{v}_1 - \dot{v}_2 = 0, \quad (2.21)$$

$$\Leftrightarrow \frac{F}{M_1} - \frac{-F + f(t)}{M_2} = 0, \quad (2.22)$$

$$\Leftrightarrow F = \frac{M_1}{M_1 + M_2} f(t). \quad (2.23)$$

Nous obtenons un SAD semi-explicite d'index 1

$$\dot{v}_1 = \frac{f(t)}{M_1 + M_2}, \quad (2.24)$$

$$\dot{v}_2 = \frac{f(t)}{M_2} \left(\frac{M_1}{M_1 + M_2} + 1 \right), \quad (2.25)$$

$$\dot{l}_{x_1} = v_1, \quad (2.26)$$

$$\dot{l}_{x_2} = v_2, \quad (2.27)$$

$$F = \frac{M_1}{M_1 + M_2} f(t). \quad (2.28)$$

Nous remarquons que si nous dérivons à nouveau l'équation permettant de calculer F on obtient alors un SDO. On peut en déduire que le SAD possède un index de différentiation de 3. Dans [68], Pantelides propose un algorithme basé sur les couplages dans un graphe biparti afin de trouver le sous-ensemble d'équations à dériver.

Comme nous venons de le voir, la résolution d'un SAD est un processus long et difficile. Il est donc très intéressant de savoir à l'avance si un SAD peut ou non être résolu. Nous allons présenter le problème de l'analyse structurelle qui donne une condition nécessaire à la résolution d'un SAD.

2.3 Analyse structurelle

Le *problème de l'analyse structurelle* (PAS) est une condition nécessaire, mais pas suffisante, permettant la résolution numérique des SAD. L'idée du PAS vient des propriétés des équations purement algébriques, sans variables dérivées. Nous allons voir,

dans un premier temps, le PAS pour les équations purement algébriques. Ensuite nous verrons comment ces propriétés se transmettent pour les SAD.

Nous parlons d'analyse structurelle, car nous ne nous intéressons pas aux valeurs que peuvent prendre les variables, nous travaillons uniquement sur la structure du système, c'est-à-dire si une variable apparaît ou non dans une équation.

Remarque 2.4 *Nous considérons des systèmes ne possédant pas d'équation redondante et des systèmes carrés possédant autant de variables que d'équations.*

Si nous avons plus d'équations que de variables ou plus de variables que d'équations et qu'il n'existe pas d'équation redondante alors, le système ne possède pas de solution ou une infinité de solutions [62].

2.3.1 PAS pour les systèmes algébriques

Les systèmes algébriques sont des systèmes d'équations possédant uniquement des variables algébriques. La forme générale des systèmes algébriques est donnée par :

$$Ax = b,$$

où A est une matrice $m \times m$, $x \in \mathbb{R}^m$ le vecteur correspondant aux variables du système et b le vecteur des constantes du système.

Si la matrice A est régulière, c'est-à-dire que la matrice A est inversible, le système possède une unique solution qui est donnée par

$$x = A^{-1}b.$$

Par contre si la matrice A est singulière alors le système ne possède pas de solution ou une infinité de solutions [31].

D'après la définition 2.3 une matrice A est régulière si et seulement si le déterminant de A est différent de 0.

Dans [64], Murota montre le lien entre la solvabilité d'un système, le déterminant des matrices et les couplages parfaits dans les graphes bipartis.

L'idée est de passer par une matrice intermédiaire appelée matrice structurelle afin de définir une condition nécessaire sur la résolution des systèmes. Cette définition a été introduite par Gantmacher [31].

Définition 2.5 *Soit une matrice A , $m \times m$. La matrice structurelle $Astr$ associée à la matrice A est définie de la manière suivante*

$$Astr_{i,j} = \begin{cases} t_{ij} & \text{si } A_{i,j} \neq 0, \\ 0 & \text{sinon.} \end{cases}$$

où les t_{ij} , pour tout $i, j \in \{1, \dots, n\}$, sont indépendants.

On peut alors associer à la matrice $Astr$ le graphe biparti $G_{Astr} = (U \cup V, E)$ où $U = \{1, \dots, m\}$, $V = \{1, \dots, m\}$ et il existe une arête $ij \in E$ entre un sommet $i \in U$ et un sommet $j \in V$ si $Astr_{i,j} \neq 0$. Le graphe G est appelé *graphe d'incidence*.

Théorème 2.6 [63] *La matrice structurelle $Astr$ est régulière si et seulement si le graphe G possède un couplage parfait.* \square

La preuve est donnée en détail dans [59].

Définition 2.7 *La matrice A est dite structurellement régulière si la matrice $Astr$ est régulière. Autrement, elle est dite structurellement singulière.*

En effet, si la matrice $Astr$ est singulière alors la matrice A l'est aussi. Par contre si la matrice $Astr$ est régulière alors on ne peut rien dire sur A , car les éléments de la matrice $Astr$ sont indépendants mais pas forcément les éléments de A . L'analyse structurelle donne donc une condition nécessaire sur la résolution des systèmes algébriques.

Par abus de langage, on dit qu'un système est structurellement régulier (resp. singulier) si la matrice structurelle associée est régulière (resp. singulière).

Remarque 2.8 [31] Si la matrice $Astr$ est singulière alors il existe un sous-ensemble d'équations possédant plus de variables que d'équations et donc, comme le système est carré (possède autant de variables que d'équations), il existe un sous-ensemble d'équations possédant moins de variables que d'équations.

Nous allons voir deux exemples pour illustrer le problème de l'analyse structurelle. Le premier exemple montre un système structurellement régulier et possédant une solution unique.

Exemple 2.9 Considérons le système algébrique suivant :

$$\begin{aligned}x_1 + 2x_2 + 3x_3 &= 4, \\x_1 + 3x_3 &= 1, \\x_1 + x_2 &= 3.\end{aligned}$$

La matrice $A1$ associée au système est équivalente à

$$A1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix}.$$

On en déduit la matrice

$$A1str = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & 0 & t_{23} \\ t_{31} & t_{32} & 0 \end{pmatrix}.$$

Le graphe biparti G_{A1str} , associé à la matrice $Astr$, est donné par la figure 2.3

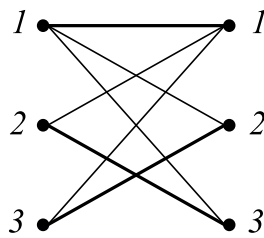


FIGURE 2.3 – Graphe biparti associé à la matrice $A1str$

Le graphe G_{A1str} possède un couplage parfait, donné par les arêtes en gras. Le système est donc structurellement régulier et possède comme solution unique $x_1 = 3/2$, $x_2 = 3/2$ et $x_3 = -1/6$.

Le deuxième exemple donne un système structurellement régulier mais qui ne possède pas de solution.

Exemple 2.10 Considérons le système algébrique suivant :

$$\begin{aligned}x_1 + 2x_2 + 3x_3 &= 4, \\2x_1 + 20x_3 &= 1, \\x_1 + 10x_3 &= 3.\end{aligned}$$

La matrice A_2 associée au système est équivalente à

$$A_2 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 0 & 20 \\ 1 & 0 & 10 \end{pmatrix}.$$

On en déduit la matrice

$$A_{2str} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & 0 & t_{23} \\ t_{31} & 0 & t_{33} \end{pmatrix}.$$

Le graphe biparti $G_{A_{2str}}$ associé à la matrice A_{2str} est donné par la figure 2.4

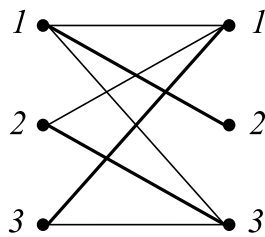


FIGURE 2.4 – Graphe biparti associé à la matrice A_{2str}

Le graphe $G_{A_{2str}}$ possède un couplage parfait, donné par les arêtes en gras. Le système est donc structurellement régulier mais pourtant il ne possède pas de solution. En effet A_2 est singulière puisque $\det(A_2) = 0$.

Nous allons voir dans la section suivante comment l'idée de l'analyse structurelle a été étendue dans le cas des SAD.

2.3.2 PAS pour les SAD

Dans [82], les auteurs rappellent qu'une définition raisonnable de la solvabilité d'un SAD est donnée par l'existence d'une solution unique pour toute valeur initiale consistante.

Comme nous l'avons vu, pour résoudre un SAD par la méthode d'Euler implicite donnée par

$$\begin{aligned}(E - hA)y^{n+1} &= Ey^n + hg(t_{n+1}), & \forall n \in \{0, \dots, N-1\}, \\ y^0 &= c_0,\end{aligned}$$

il est nécessaire que la matrice $(E - hA)$ soit inversible et donc que $\det(E - hA) \neq 0$. Luenberger [62] a généralisé cette condition quelque soit la méthode de résolution utilisée.

Théorème 2.11 [62] *Un système est soluble si et seulement si $\det(E - \lambda A) \neq 0$, où λ est un paramètre défini en fonction de la méthode de résolution choisie et de h .*

On s'intéresse à la structure de la matrice $(E - \lambda A)$. Pour cela, nous allons définir la matrice structurelle $\hat{A}str$, où $\hat{A} = (E - \lambda A)$.

Définition 2.12 *Soit la matrice $\hat{A}str$ d'un SAD est définie par :*

$$\hat{A}str_{i,j} = \begin{cases} t_{ij} & \text{si } E_{i,j} \neq 0 \text{ et/ou } A_{i,j} \neq 0, \\ 0 & \text{sinon.} \end{cases}$$

où les t_{ij} , pour tout $i, j \in \{1, \dots, n\}$, sont indépendants.

En effet, $\hat{A}_{i,j} = 0$ si et seulement si $E_{i,j} = 0$ et $A_{i,j} = 0$. Par contre si $E_{i,j} \neq 0$ et/ou $A_{i,j} \neq 0$ alors il existe un λ tel que $\hat{A}_{i,j} \neq 0$.

Corollaire 2.13 [82] *Un système n'est pas soluble si sa matrice structurelle $\hat{A}str$ est singulière.*

Remarque 2.14 La condition $E_{i,j} \neq 0$ et/ou $A_{i,j} \neq 0$ correspond au fait que la variable y_i ou \dot{y}_i apparaît dans l'équation j

Il est facile de voir, d'après la remarque 2.14, que pour résoudre le PAS il n'est pas nécessaire de passer par la matrice structurelle. Il nous suffit de vérifier qu'une variable apparaît ou non dans une équation. Il est alors possible de redéfinir le graphe d'incidence de la manière suivante. Soit le graphe biparti $G = (U \cup V, E)$ où chaque sommet de U correspond à une équation, chaque sommet de V à une variable, et il existe une arête $uv \in E$ entre un sommet $u \in U$ et un sommet $v \in V$ si la variable ou sa dérivée correspondant au sommet v apparaît dans l'équation correspondant au sommet u .

Si la matrice \hat{A}_{str} est singulière alors le SAD n'est pas solvable. Le PAS dans le cas des SAD revient à vérifier que la matrice \hat{A} est structurellement régulière.

Considérons à nouveau le SAD suivant :

$$\begin{aligned} \dot{y}_1 + \dot{y}_2 &= y_1 + y_2, \\ \dot{y}_1 + \dot{y}_2 &= 2y_1 + 5, \end{aligned} \tag{2.29}$$

La matrice \hat{A} associée à ce système est équivalente à

$$\hat{A} = \begin{pmatrix} 1 - \lambda & 1 - \lambda \\ 1 - 2\lambda & 1 \end{pmatrix}.$$

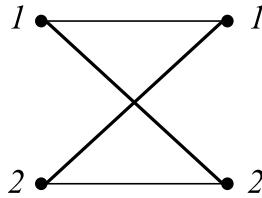
On en déduit la matrice :

$$\hat{A}_{str} = \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix}.$$

Le graphe biparti $G_{\hat{A}_{str}}$ associé à la matrice \hat{A}_{str} est donné par la figure 2.5

D'après le théorème 2.6, comme le graphe $G_{\hat{A}_{str}}$ possède un couplage parfait, donné par les arêtes en gras, le SAD est donc structurellement régulier. Par ailleurs, dans la section 2.2.2 nous avons montré que ce système peut être résolu numériquement sous certaines conditions.

Nous allons voir dans la prochaine section les méthodes existantes permettant de savoir si un SAD est structurellement régulier ou non.

FIGURE 2.5 – Graphe biparti associé à la matrice $\hat{A}str$

2.4 Analyse structurelle et problèmes connexes

Il peut être utile de résoudre certains problèmes avant la résolution d'un SAD par des méthodes numériques.

Plusieurs techniques basées sur la théorie des graphes ont été proposées pour résoudre le PAS [12, 20, 64, 65]. D'autres méthodes, basées sur les matroïdes [64, 76] ont également été proposées pour le PAS.

Une des difficultés, quand un SAD est structurellement singulier, est de savoir la raison pour laquelle le système est structurellement singulier. L'une des méthodes utilisées est de détecter le ou les sous-systèmes structurellement singuliers.

Murota [63] a montré qu'il est possible de détecter les sous-systèmes structurellement singuliers en passant par le graphe biparti G associé. Pour cela, il utilise la DM-Décomposition [60] qui permet de décomposer le graphe biparti G en plusieurs sous-graphes bipartis. Cette décomposition se base sur un algorithme flots. Murota a alors montré qu'exactly deux de ces sous-graphes correspondent à des sous-systèmes structurellement singuliers. Considérons le graphe biparti de la figure 2.6. La DM-Décomposition correspondant à ce graphe est donnée par la figure 2.7. Les sous-systèmes associés aux graphes G_1 et G_4 sont structurellement singuliers. Dans [42], les auteurs proposent une méthode pour analyser et détecter les sous-systèmes minimaux structurellement singuliers. Cette méthode se base sur la DM-décomposition afin d'isoler les sous-systèmes d'équations structurellement singuliers. Pour chaque sous-système, un ensemble d'équations fictives est ajouté. Le système résultant est lui aussi décomposé, et ce, jusqu'à ce qu'un sous-système structurellement singulier minimum soit détecté. La méthode est appliquée de manière récursive jusqu'à ce que tous les sous-systèmes structurellement singuliers minimaux soient localisés. Cette méthode permet

d'affiner la décomposition donnée par Murota.

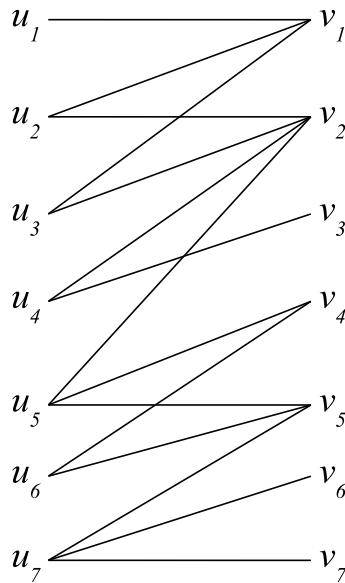


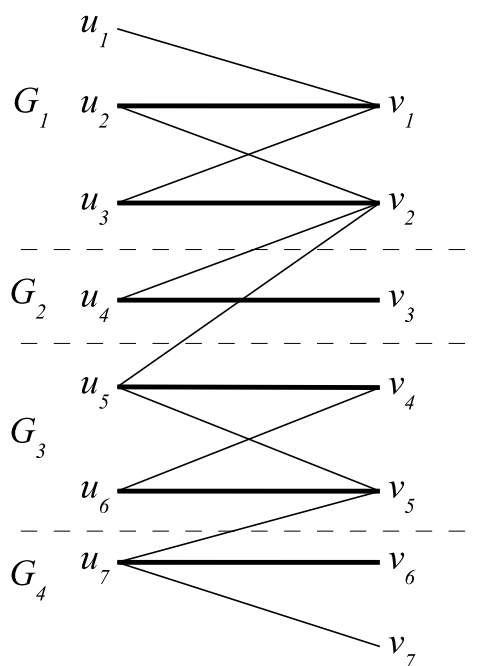
FIGURE 2.6 – Exemple de graphe G

Il est également intéressant de réduire l'index d'un SAD afin que la résolution soit plus stable. Pantelides [68] a donné un algorithme exact, basé sur les couplages, pour le calcul de l'index d'un SAD. Il donne également une méthode pour réduire l'index d'un SAD, également basée sur les couplages. D'autres méthodes, basées sur les graphes [64, 57], sur les matrices [64, 73] et les matroïdes [64] ont également été proposées pour la réduction d'index.

Nous allons voir dans la prochaine section, une généralisation des SAD. Cette généralisation permet de modéliser des changements dans le système. Cela permet, par exemple, de modéliser un interrupteur dans un circuit électrique.

2.5 Les systèmes algébro-différentiels conditionnels

Dans cette section, nous présentons les systèmes algébro-différentiels conditionnels, c'est-à-dire les systèmes dont la forme de certaines équations différentielles dépend de

FIGURE 2.7 – Exemple de DM-Décomposition de G

certaines conditions. Nous présentons ensuite le problème de l'analyse structurelle pour de tels systèmes.

Une équation différentielle conditionnelle est une équation associée à une condition et générant deux équations différentielles selon que la condition soit vraie ou fausse.

Les équations conditionnelles permettent de modéliser les changements d'état du système, tels que l'utilisation d'un interrupteur dans un circuit électrique ou un changement de température dans un système hydraulique.

Un système algébro-différentiel conditionnel (SADC) est un SAD comportant des équations différentielles conditionnelles. Considérons par exemple le SADC (S_1) suivant :

$$eq_1 : \text{if } a > 0 \quad (2.30)$$

$$\text{then} \quad 0 = 4x_1^2 + 2\dot{x}_1 + 4x_2 + 2,$$

$$\text{else} \quad 0 = \dot{x}_2 + 2x_3 + 4,$$

$$eq_2 : \text{if } b > 0 \quad (2.31)$$

$$\text{then} \quad 0 = 6\dot{x}_2 + 2\dot{x}_3 + 2,$$

$$\text{else} \quad 0 = x_1 + \dot{x}_2 + 1,$$

$$eq_3 : \text{if } c > 0 \quad (2.32)$$

$$\text{then} \quad 0 = 6\dot{x}_1 + x_2 + 2,$$

$$\text{else} \quad 0 = 3\dot{x}_2 + x_3 + 3,$$

$$eq_4 : \quad 0 = 6\dot{x}_4 + x_1 + x_3 + 1. \quad (2.33)$$

Les équations (2.30) - (2.32) sont les équations différentielles conditionnelles.

On suppose que pour de tels systèmes, les différentes conditions associées aux équations différentielles conditionnelles sont indépendantes, c'est-à-dire qu'il n'existe pas de relation entre les conditions. De plus nous considérons des systèmes carrés, c'est-à-dire qu'ils possèdent autant de variables que d'équations. Il est clair que si le système n'est pas carré alors il est structurellement singulier.

On appelle *état du système*, une valeur vraie/faussee de toutes les conditions du système. Notons que chaque état du SADC induit un SAD.

L'état donné par les conditions $a > 0$, $b > 0$, $c > 0$ vraies, donne alors le système (S_2) suivant :

$$eq_1 : \quad 0 = 4x_1^2 + 2\dot{x}_1 + 4x_2 + 2, \quad (2.34)$$

$$eq_2 : \quad 0 = 6\dot{x}_2 + 2\dot{x}_3 + 2, \quad (2.35)$$

$$eq_3 : \quad 0 = 6\dot{x}_1 + x_2 + 2, \quad (2.36)$$

$$eq_4 : \quad 0 = 6\dot{x}_4 + x_1 + x_3 + 1. \quad (2.37)$$

Le système donné par l'état $a > 0$, $b > 0$ vraies et $c > 0$ fausse est le système (S_3) suivant :

$$eq_1 : \quad 0 = 4x_1^2 + 2\dot{x}_1 + 4x_2 + 2, \quad (2.38)$$

$$eq_2 : \quad 0 = 6\dot{x}_2 + 2\dot{x}_3 + 2, \quad (2.39)$$

$$eq_3 : \quad 0 = 3\dot{x}_2 + x_3 + 3, \quad (2.40)$$

$$eq_4 : \quad 0 = 6\dot{x}_4 + x_1 + x_3 + 1. \quad (2.41)$$

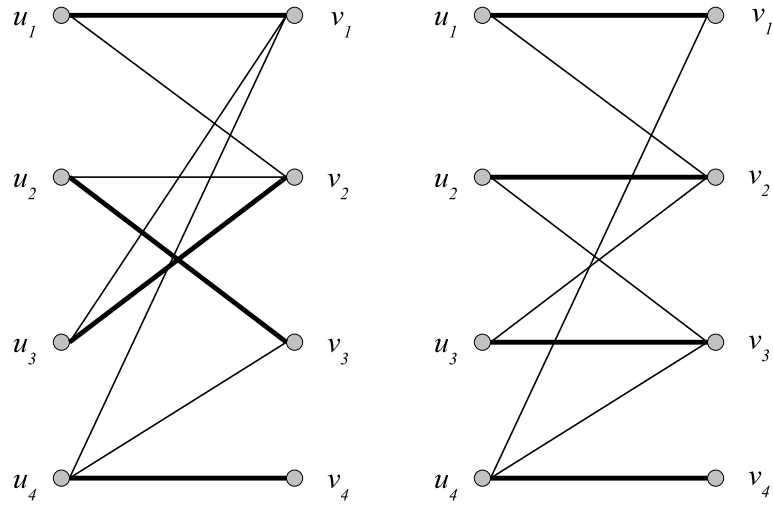


FIGURE 2.8 – Graphes d'incidences des systèmes (S_2) et (S_3) .

La figure 2.8 montre les graphes d'incidence associés aux systèmes (S_2) et (S_3) . Les sommets u_1, u_2, u_3 et u_4 correspondent respectivement aux équations eq_1, eq_2, eq_3 et eq_4 . Les sommets v_1, v_2, v_3 et v_4 correspondent respectivement aux variables x_1, x_2, x_3 et x_4 . Les graphes d'incidence associés aux systèmes (S_2) et (S_3) , possèdent un couplage parfait (les arêtes en gras), ce qui implique que les systèmes (S_2) et (S_3) sont structurellement réguliers. Si on considère l'état défini par, les conditions $a > 0, c > 0$ fausses et $b > 0$ vraie, nous obtenons le système (S_4) :

$$eq_1 : \quad 0 = \dot{x}_2 + 2x_3 + 4, \quad (2.42)$$

$$eq_2 : \quad 0 = 6\dot{x}_2 + 2\dot{x}_3 + 2, \quad (2.43)$$

$$eq_3 : \quad 0 = 3\dot{x}_2 + x_3 + 3, \quad (2.44)$$

$$eq_4 : \quad 0 = 6\dot{x}_4 + x_1 + x_3 + 1. \quad (2.45)$$

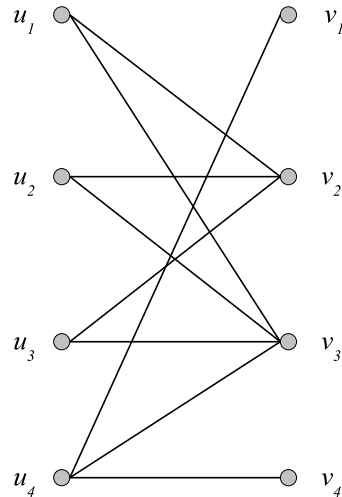


FIGURE 2.9 – Graphe d'incidence du système (S_4) .

La figure (2.9) montre le graphe d'incidence associé au système (S_4) . Ce graphe ne possède pas de couplage parfait et donc le système (S_4) est structurellement singulier.

Le PAS pour les SADC consiste à vérifier si les systèmes induits par l'ensemble des états du système sont structurellement réguliers ou à trouver au moins un état du système induisant un système structurellement singulier. Le PAS pour un SADC revient donc à vérifier si les graphes d'incidence associés à chaque état possèdent un couplage parfait. Le SADC (S_1) est structurellement singulier, car il existe au moins un état induisant un système structurellement singulier, par exemple l'état correspondant au système (S_4) . On appelle *problème de l'analyse structurelle conditionnelle* (PASC) le PAS associé aux SADC.

Comme il existe un nombre exponentiel d'états possibles nous ne pouvons, par conséquent, pas tous les énumérer. Nous devons donc développer une technique efficace pour résoudre le problème. À notre connaissance, le PASC n'a pas été étudié.

2.6 Conclusion

Dans ce chapitre, nous avons présenté les systèmes algébro-différentiels. Nous avons aussi donné les idées permettant de résoudre ces systèmes et nous avons expliqué l'intérêt du problème de l'analyse structurelle et comment il était résolu dans la littérature. Nous avons ensuite étendu les systèmes algébro-différentiels aux systèmes algébro-différentiels conditionnels et nous avons défini le problème de l'analyse structurelle pour ces systèmes. Dans le chapitre suivant, nous proposons une formulation pour le problème de l'analyse structurelle conditionnelle en terme de graphes. Nous montrons ensuite que ce problème est difficile à résoudre. Nous proposons enfin deux modélisations sous la forme de programmes linéaires en nombres entiers.

Chapitre 3

Analyse structurelle des systèmes algébro-différentiels conditionnels

Comme indiqué dans le chapitre 2, un système algébro-différentiel conditionnel est un système algébro-différentiel contenant des équations différentielles conditionnelles, c'est-à-dire des équations dont la forme dépend de certaines conditions. Nous rappelons que nous considérons dans notre étude des SADC carrés.

Le problème de l'analyse structurelle pour de tels systèmes consiste alors à déterminer s'il existe un état du système, c'est-à-dire une valeur pour chaque condition, induisant un SAD structurellement singulier.

Dans la section 1, nous formulons le problème de l'analyse structurelle conditionnelle en termes de graphes. Nous montrons ensuite, dans la section 2, que ce problème est NP-complet. Nous donnons dans la section 3 une première modélisation sous forme de programme linéaire en nombres entiers. Nous terminons par une deuxième modélisation du PASC.

3.1 Problème de l'analyse structurelle conditionnelle et graphes bipartis

Dans cette section, nous présentons une formulation du problème de l'analyse structurelle conditionnelle en termes de graphes. Nous montrons que le problème se ramène à déterminer s'il existe, dans un graphe biparti, un sous-graphe vérifiant certaines propriétés et ne contenant pas de couplage parfait.

Étant donné un SADC, à chaque équation du SADC, on associe un sommet et l'on note par U l'ensemble de ces sommets. On note par U_c l'ensemble des sommets de U associés aux équations conditionnelles. À chaque variable, on associe un sommet et l'on note par V cet ensemble de sommets.

Entre un sommet $u \in U_c$ et un sommet $v \in V$, on considère une arête, dite *vraie* (resp. *fausse, vraie/fausse*), si la variable associée à v apparaît dans l'équation associée à u , uniquement quand la condition de l'équation est vraie (resp. fausse, à la fois vraie et fausse). Entre un sommet $u \in U \setminus U_c$ et un sommet $v \in V$, on considère une arête uv , dite *vraie/fausse*, si la variable associée à v apparaît dans l'équation associée à u . On note par $G = (U \cup V, E)$ le graphe biparti ainsi obtenu où E est l'ensemble des arêtes.

Pour tout sommet $u \in U$, on note par E_u^t (resp. E_u^f, E_u^{tf}) les arêtes vraies (resp. fausses, vraies/fausses) incidentes à u . On remarque que ces ensembles sont disjoints. De plus, les ensembles E_u^t et E_u^f sont vides pour tout sommet $u \in U \setminus U_c$. Soit $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U\}$. On remarque que π forme une partition de E . On pose $E^{tf} = \bigcup_{u \in U} E_u^{tf}$.

Une instance de notre problème est donc définie par le triplet (G, U_c, π) où $G = (U \cup V, E)$, $U_c \subseteq U$ et π est une partition de E .

Notons par \mathcal{E} le sous-ensemble de π donné par $\mathcal{E} = \{E_u^t, E_u^f : u \in U_c\}$. On remarque que \mathcal{E} ne contient pas les ensembles d'arêtes vraies/fausses de π , ni les ensembles vides E_u^t, E_u^f des sommets $u \in U \setminus U_c$ associés à des équations non-conditionnelles.

La figure 3.1 présente le graphe biparti G et les labels correspondant au système (S_1) suivant, contenant trois équations conditionnelles et une équation non-conditionnelle :

$$eq_1 : \text{if } a > 0 \quad (3.1)$$

$$\text{then} \quad 0 = 4\dot{x}_1^2 + 2\dot{x}_1 + 4x_2 + 2,$$

$$\text{else} \quad 0 = \dot{x}_2 + 2x_3 + 4,$$

$$eq_2 : \text{if } b > 0 \quad (3.2)$$

$$\text{then} \quad 0 = 6\dot{x}_2 + 2\dot{x}_3 + 2,$$

$$\text{else} \quad 0 = x_1 + \dot{x}_2 + 1,$$

$$eq_3 : \text{if } c > 0 \quad (3.3)$$

$$\text{then} \quad 0 = 6\dot{x}_1 + x_2 + 2,$$

$$\text{else} \quad 0 = 3\dot{x}_2 + x_3 + 3,$$

$$eq_4 : \quad 0 = 6\dot{x}_4 + x_1 + x_3 + 1. \quad (3.4)$$

Ici nous avons $U = \{u_1, u_2, u_3, u_4\}$, $V = \{v_1, v_2, v_3, v_4\}$ et $U_c = \{u_1, u_2, u_3\}$. La partition π se déduit des labels des arêtes. Elle contient 12 sous-ensembles.

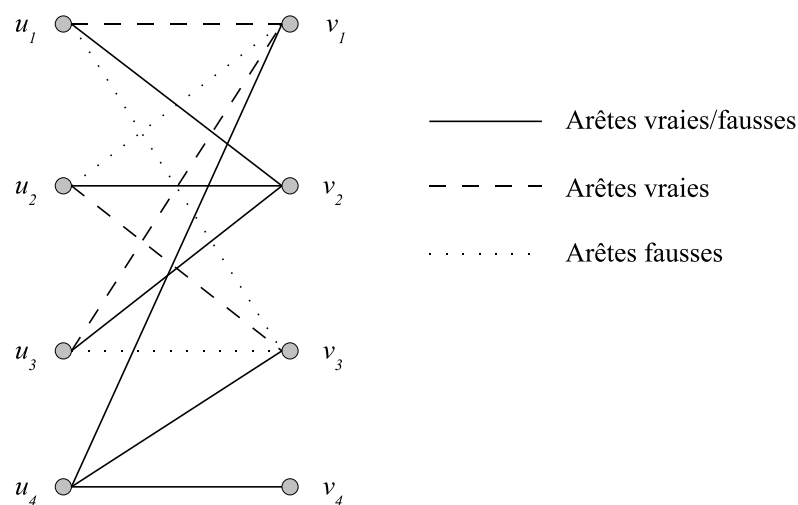


FIGURE 3.1 – Graphe biparti G avec les labels vrais, faux et vrais/faux du système (S_1) .

Nous introduisons maintenant la notion de graphe associé à un sous-ensemble de l'ensemble \mathcal{E} . Étant donné $\mathcal{F} \subseteq \mathcal{E}$, on appelle graphe associé à \mathcal{F} le sous-graphe de G

dont l'ensemble des sommets est $U \cup V$ et l'ensemble des arêtes est constitué des arêtes vraies/fausses de G ainsi que des arêtes appartenant à un élément de \mathcal{F} . Le graphe associé à $\mathcal{F} \subseteq \mathcal{E}$ sera par la suite noté $H_{\mathcal{F}} = (U \cup V, E_{\mathcal{F}})$, où $E_{\mathcal{F}} = E^{tf} \cup (\cup_{F \in \mathcal{F}} F)$.

Nous définissons maintenant le problème dit problème du sous-graphe sans couplage parfait, correspondant à la formulation du problème de l'analyse structurelle en termes de graphes.

Définition 3.1 *Étant donné une instance (G, U_c, π) où $G = (U \cup V, E)$, $U_c \subseteq U$ et $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U\}$ est une partition de E , le Problème du Sous-Graphe Sans Couplage Parfait (PSGSCP) consiste à déterminer s'il existe un sous-ensemble $\mathcal{F} \subseteq \mathcal{E}$ tel que*

- (i) *pour tout $u \in U_c$, soit $E_u^t \in \mathcal{F}$ soit $E_u^f \in \mathcal{F}$ (mais pas les deux),*
- (ii) *le graphe $H_{\mathcal{F}}$ ne possède pas de couplage parfait.*

Comme nous allons le voir, les deux problèmes PASC et PSGSCP sont équivalents. On remarque tout d'abord que tout graphe $H_{\mathcal{F}}$ vérifiant la condition (i) correspond au graphe d'incidence d'un état du système. En effet, les arêtes qui ne dépendent pas des conditions du système correspondent aux arêtes vraies/fausses de G . Elles appartiennent au graphe d'incidence de tout état du système. De plus, la condition (i) implique que pour toute condition, sa valeur est soit fixée à vrai et donc $E_u^t \in \mathcal{F}$, soit fixée à faux et donc $E_u^f \in \mathcal{F}$. Par ailleurs, on remarque que le graphe d'incidence de tout état du système correspond à un graphe $H_{\mathcal{F}}$ vérifiant la condition (i).

Maintenant supposons qu'il existe un sous-ensemble \mathcal{F} vérifiant les conditions (i) et (ii). Cela implique alors qu'il existe un état C du système dont le graphe d'incidence ne contient pas de couplage parfait. D'après le théorème 2.6, cela signifie que le système donné par l'état C est structurellement singulier. Et par conséquent, le SADC l'est aussi.

Réciproquement, s'il n'existe aucun sous-ensemble \mathcal{F} vérifiant les conditions (i) et (ii), cela implique que les graphes d'incidence de tous les états du système ont un couplage parfait. On en déduit alors que les systèmes induits par chaque état sont structurellement réguliers, et par conséquent, le SADC est structurellement régulier.

On remarque par exemple que le système (S_1) est structurellement singulier. En effet, le graphe associé au sous-ensemble $\mathcal{F} = \{E_1^f, E_2^t, E_3^f\}$, donné par la figure 3.2, vérifie les conditions (i) et (ii) de la définition 3.1. En effet, l'ensemble de sommets $\{u_1, u_2, u_3\}$ possède seulement deux voisins donc le graphe de la figure 3.2 ne possède

pas de couplage parfait. Ceci implique que le système induit par l'état $a > 0$, $c > 0$ faux et $b > 0$ vrai est structurellement singulier.

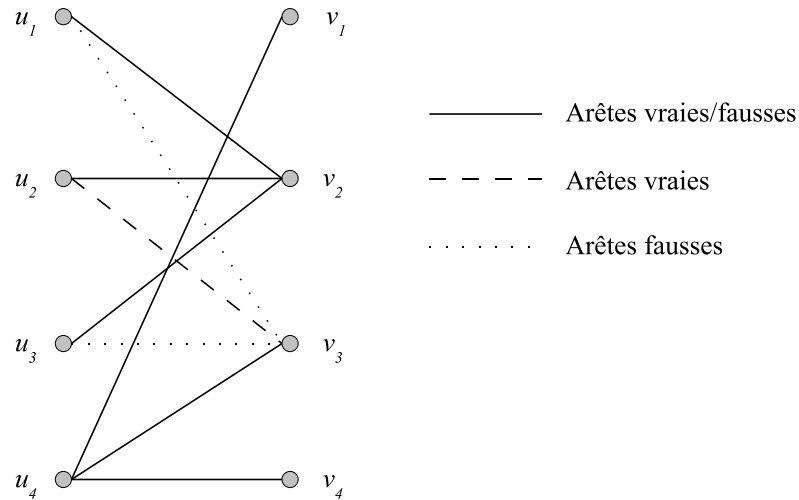


FIGURE 3.2 – Graphe H , correspondant à l'état $a > 0$, $c > 0$ faux et $b > 0$ vrai du système (S_1) .

3.2 Complexité

Comme il a été montré dans le chapitre 2, dans le cas des SAD non conditionnels, le PAS se ramène à vérifier l'existence d'un couplage parfait dans un graphe biparti, ce qui peut être réalisé en temps polynomial. Dans cette section, nous montrons que le PAS pour les SADC est NP-complet. Pour ce faire, nous montrons dans un premier temps que le problème PSGSCP est équivalent au problème du stable dans un graphe triparti particulier. Nous montrons ensuite que ce dernier est NP-complet.

3.2.1 Relation avec le problème du stable

Soit $H = (V^1 \cup V^2 \cup V^3, F)$ un graphe triparti où $|V^1| = |V^2| = |V^3| = n$, $V^j = \{v_1^j, \dots, v_n^j\}$ pour $j = 1, 2, 3$ et V^1 et V^2 sont connectés uniquement par un couplage

parfait. Considérons le problème qui consiste à déterminer s'il existe un stable dans H de taille supérieure ou égale $n + 1$. Ce problème sera appelé *problème du stable triparti avec couplage parfait* (PSTCP).

Le théorème suivant montre que les problèmes PSGSCP et PSTCP sont équivalents.

Théorème 3.2 *PSGSCP et PSTCP sont équivalents.*

Preuve. Nous considérons dans cette preuve que les équations non-conditionnelles sont des équations conditionnelles qui génèrent toujours les mêmes équations quelques soient les valeurs des conditions. Nous avons donc $U = U_c$. On notera par (G, U_c, π) une instance du PSGSCP, avec $G = (U \cup V, E)$, et par $H = (V^1 \cup V^2 \cup V^3, F)$ une instance du PSTCP.

Nous montrons dans un premier temps qu'une instance du PSTCP peut se réduire en une instance du PSGSCP. On suppose, sans perte de généralité, que les sommets de V_1 et V_2 sont reliés par le couplage parfait $M = \{v_1^1 v_1^2, v_2^1 v_2^2, \dots, v_n^1 v_n^2\}$. Pour chaque arête $v_i^1 v_i^2$ du couplage parfait où $v_i^1 \in V^1$ et $v_i^2 \in V^2$ considérons un sommet u_i dans U_c . Pour chaque sommet v_i^3 de V^3 considérons un sommet v_i dans V . De plus, si $v_i^1 v_k^3$ et $v_i^2 v_k^3$ sont dans F pour $i, k \in \{1, \dots, n\}$, alors nous ajoutons une arête $u_i v_k$ dans $E_{u_i}^{tf}$. Et si $v_i^1 v_k^3$ (resp. $v_i^2 v_k^3$) est dans F mais pas $v_i^2 v_k^3$ (resp. $v_i^1 v_k^3$) pour $i, k \in \{1, \dots, n\}$, alors nous ajoutons une arête $u_i v_k$ dans $E_{u_i}^t$ (resp. $E_{u_i}^f$). On pose $E = \bigcup_{u \in U} E_u^t \cup E_u^f \cup E_u^{tf}$ et $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U_c\}$.

La figure 3.3 illustre la transformation d'une instance du PSTCP en une instance du PSGSCP. Notons que le graphe $H = (V^1 \cup V^2 \cup V^3, F)$ peut être obtenu depuis le graphe $G = (U \cup V, E)$ en faisant l'opération inverse.

Dans ce qui suit, nous allons montrer qu'il existe un stable dans H de taille $n + 1$ si et seulement s'il existe un sous-ensemble $\mathcal{F} \subseteq \mathcal{E}$ vérifiant les conditions (i) et (ii). Supposons d'abord qu'il existe un sous-ensemble $\mathcal{F} \subseteq \mathcal{E}$ vérifiant les conditions (i) et (ii). Dans ce cas, le graphe $H_{\mathcal{F}}$ associé à \mathcal{F} ne possède pas de couplage parfait. Tout couplage dans $H_{\mathcal{F}}$ contient au plus $n - 1$ arêtes. Et comme $|U \cup V| = 2n$, par le corollaire 1.3, il existe un stable dans $H_{\mathcal{F}}$, disons S' , de taille $|S'| \geq n + 1$. Nous allons construire, dans la suite, un stable dans H de même cardinalité. Soit S le sous-ensemble de sommets de H obtenu de la manière suivante. Pour chaque sommet $v_j \in V \cap S'$, on considère un sommet v_j^3 dans S . Et pour chaque sommet $u_i \in U \cap S'$, on considère un sommet v_i^1 dans S si $E_{u_i}^t \in \mathcal{F}$ et un sommet v_i^2 dans S si $E_{u_i}^f \in \mathcal{F}$. Comme $|S'| \geq n + 1$, nous avons $|S| \geq n + 1$. De plus, S est un stable. En effet, étant donné que les arêtes

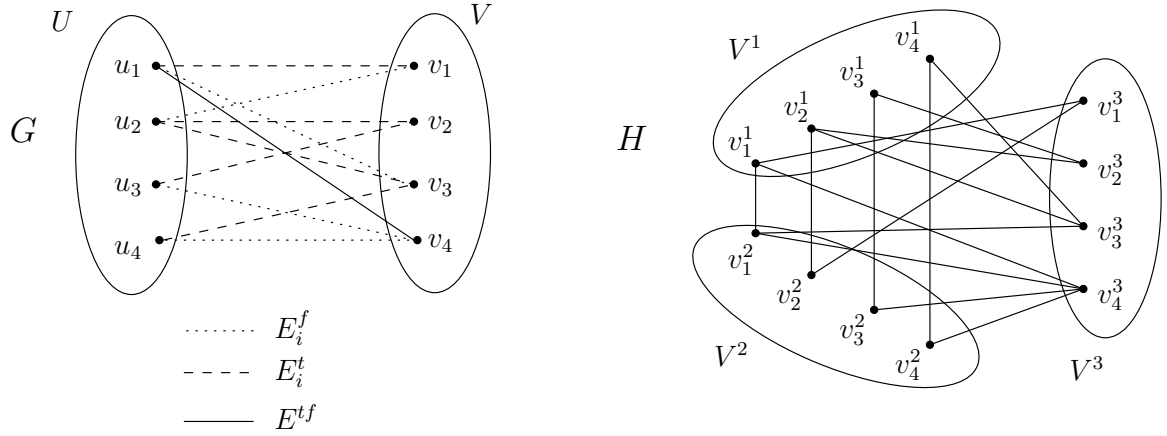


FIGURE 3.3 – Deux instances équivalentes du PSGSCP et du PSTCP.

entre V^1 et V^2 sont celles du couplage parfait M , et S intersecte chaque arête de M en au plus un sommet, alors la restriction de S sur $V^1 \cup V^2$ est un stable. Supposons maintenant que S contient deux sommets adjacents, disons $v_i^1 \in V^1$ et $v_j^3 \in V^3$. Ceci implique que u_i et v_j appartiennent à S' . Comme $u_i v_j \in E_{u_i}^t$ et $E_{u_i}^t \in \mathcal{F}$, cela contredit le fait que S' est un stable de $H_{\mathcal{F}}$. En utilisant le même argument, nous déduisons que S ne peut contenir deux sommets $v_i^2 \in V^2$ et $v_j^3 \in V^3$ adjacents. Par conséquent, S est une solution du PSTCP.

Inversement, supposons que nous avons un stable S dans H de taille supérieure ou égale à $n + 1$. Soit \mathcal{F} le sous-ensemble de \mathcal{E} obtenu de la manière suivante. Pour chaque sommet $v_i^1 \in V^1$ on ajoute dans \mathcal{F} l'ensemble $E_{u_i}^t$ si $v_i^1 \in S$, et $E_{u_i}^f$ sinon. Nous montrons que $H_{\mathcal{F}}$ contient un stable de taille supérieure ou égale à $n + 1$. Soit $S' \subseteq U \cup V$ un ensemble de sommets obtenu à partir de S comme suit. Pour un sommet v_i^3 de $V^3 \cap S$ on ajoute le sommet v_i de V dans S' . Et pour chaque sommet v_i^1 (resp. v_i^2) de $V^1 \cap S$ (resp. $V^2 \cap S$), on ajoute le sommet u_i de U dans S' . Comme S ne contient pas les sommets v_i^1 et v_i^2 pour chaque i , et $|S| \geq n + 1$, il en résulte que $|S'| \geq n + 1$. De plus S' est un stable. En effet, supposons que S' contient deux sommets, disons $u_i \in U$ et $v_j \in V$ tels que $u_i v_j \in E_{\mathcal{F}}$. Sans perte de généralité, on peut supposer que u_i correspond au sommet v_i^1 dans S (la preuve est similaire si $v_i^2 \in S$). Par construction de $E_{\mathcal{F}}$, on en déduit que $E_{u_i}^t \in \mathcal{F}$, et donc $u_i v_j \in E_{u_i}^t$. Il s'ensuit alors que $v_i^1 v_j^3 \in F$. Comme $v_i^1, v_j^3 \in S$, on a une contradiction, ce qui achève la preuve. \square

3.2.2 NP-complétude

Dans cette section nous montrons que le PSGSCP est NP-complet. Par le théorème 3.2, il suffit de montrer que le PSTCP est NP-complet. Dans [71], Phillips et Warnow ont montré que le problème du stable est NP-complet dans les graphes tripartis avec couplage parfait. (Rappelons que le problème peut être résolu en temps polynomial dans les graphes bipartis). Nous allons montrer dans la suite que la variante plus restrictive du problème du stable dans un graphe triparti, donnée par le PSTCP, est aussi NP-complet. En d'autres termes, le problème du stable dans les graphes tripartis reste NP-complet même quand les ensembles de la partition du graphe sont de mêmes tailles et deux de ces ensembles sont connectés uniquement par un couplage parfait. Pour prouver ce résultat, nous allons utiliser une transformation à partir du problème un-parmi-trois 3SAT. Une instance du *un-parmi-trois 3SAT* (1-parmi-3 3SAT) est composée de n littéraux l_1, \dots, l_n et m clauses C_1, \dots, C_m avec trois variables par clause. Chaque clause est la disjonction de trois variables où une variable correspond soit à un littéral soit à sa négation. Soit x_i la variable qui représente soit l_i soit \bar{l}_i , alors $\bar{x}_i = \bar{l}_i$ (resp. $\bar{x}_i = l_i$) si $x_i = l_i$ (resp. $x_i = \bar{l}_i$) où la barre représente la négation. Le problème est de déterminer s'il existe une affectation "vrai" et "faux" aux variables de telle manière que chaque clause possède exactement un littéral vrai.

Théorème 3.3 *PSTCP est NP-complet.*

Preuve. La preuve utilise des idées de [71]. Il est clair que le PSTCP est dans NP. Pour prouver ce théorème, nous utilisons une réduction polynomiale du problème 1-parmi-3 3SAT. Considérons une instance du 1-parmi-3 3SAT donnée par un ensemble de n littéraux $L = \{l_1, \dots, l_n\}$ et un ensemble de m clauses $C = \{C_1, \dots, C_m\}$. Nous allons construire une instance du PSTCP sur un graphe $H = (V^1 \cup V^2 \cup V^3, F)$ où $|V^1| = |V^2| = |V^3| = p = 3n + m - 1$ et l'ensemble des arêtes entre V^1 et V^2 forme un couplage parfait. Nous montrons que H possède un stable de taille $p+1$ si et seulement si le problème 1-parmi-3 3SAT admet une affectation admissible.

À chaque littéral $l_i \in L$, on associe les sommets $v_i^1, \bar{v}_i^1 \in V^1$, $v_i^2, \bar{v}_i^2 \in V^2$ et $v_i^3, \bar{v}_i^3 \in V^3$. Ces sommets sont appelés *sommets-littéraux*. À chaque clause $C_j = (x_r, x_s, x_t)$, on associe les sommets $w_{j_r}^1 \in V^1$, $w_{j_s}^2 \in V^2$, $w_{j_t}^3 \in V^3$. Ces sommets sont appelés *sommets-clauses*. Enfin on ajoute les sommets $z_q^1 \in V^1$, $z_q^2 \in V^2$, $z_q^3 \in V^3$ pour $q = 1, \dots, n-1$. Ces sommets sont appelés *sommets-fictifs*. Notons que $|V^1| = |V^2| = |V^3| = p$. Dans ce qui suit nous construisons l'ensemble d'arêtes F . Pour chaque littéral $l_i \in L$, considérons les arêtes $v_i^1 \bar{v}_i^2, \bar{v}_i^2 v_i^3, v_i^3 \bar{v}_i^1, \bar{v}_i^1 v_i^2, v_i^2 \bar{v}_i^3, \bar{v}_i^3 v_i^1$ dans F . Celles-ci sont appelées *arêtes-littéraux*.

Remarquons que ces arêtes forment un cycle de longueur 6, que nous notons par la suite par Γ_i pour $i = 1, \dots, n$. Pour chaque clause $C_j = (x_r, x_s, x_t)$ ajoutons dans F les arêtes $w_{j_r}^1 w_{j_s}^2, w_{j_s}^2 w_{j_t}^3, w_{j_t}^3 w_{j_r}^1$. Elles sont appelées *arêtes-clauses*. Notons que ces arêtes forment un triangle, que nous notons par la suite par T_j , pour $j = 1, \dots, m$. Nous ajoutons aussi dans F les arêtes $z_q^1 z_q^2$ pour $q = 1, \dots, n - 1$. On remarque que les arêtes entre V^1 et V^2 forment un couplage parfait donné par les arêtes $v_i^1 \bar{v}_i^2, \bar{v}_i^1 v_i^2, i = 1, \dots, n, w_{j_r}^1 w_{j_s}^2, j = 1, \dots, m$, et $z_q^1 z_q^2, q = 1, \dots, n - 1$. En plus, en fonction de la valeur des variables, nous ajoutons des arêtes dans E de la manière suivante. Pour chaque clause (x_r, x_s, x_t) :

- si $x_r = l_r$, on ajoute les arêtes $w_{j_r}^1 \bar{v}_r^3, w_{j_s}^2 v_r^3, w_{j_t}^3 v_r^1, w_{j_t}^3 v_r^2$,
- si $x_r = \bar{l}_r$, on ajoute les arêtes $w_{j_r}^1 v_r^3, w_{j_s}^2 \bar{v}_r^3, w_{j_t}^3 \bar{v}_r^1, w_{j_t}^3 \bar{v}_r^2$,
- si $x_s = l_s$, on ajoute les arêtes $w_{j_r}^1 v_s^3, w_{j_s}^2 \bar{v}_s^3, w_{j_t}^3 v_s^1, w_{j_t}^3 v_s^2$,
- si $x_s = \bar{l}_s$, on ajoute les arêtes $w_{j_r}^1 \bar{v}_s^3, w_{j_s}^2 v_s^3, w_{j_t}^3 \bar{v}_s^1, w_{j_t}^3 \bar{v}_s^2$,
- si $x_t = l_t$, on ajoute les arêtes $w_{j_r}^1 v_t^3, w_{j_s}^2 v_t^3, w_{j_t}^3 \bar{v}_t^1, w_{j_t}^3 \bar{v}_t^2$,
- si $x_t = \bar{l}_t$, on ajoute les arêtes $w_{j_r}^1 \bar{v}_t^3, w_{j_s}^2 \bar{v}_t^3, w_{j_t}^3 v_t^1, w_{j_t}^3 v_t^2$.

Ces arêtes sont appelées *arêtes de satisfiabilité*. Enfin, on lie tout sommet-fictif dans $V^1 \cup V^2$ à tous les sommets de V^3 , et tout sommet-fictif dans V^3 à tous les sommets non fictifs dans $V^1 \cup V^2$.

Donc, à partir d'une instance du 1-parmi-3 3SAT avec n littéraux et m clauses, nous obtenons un graphe tripartite avec $9n + 3m - 3$ sommets et $10n^2 + 4nm - 5n + 14m + 1$ arêtes. La figure 3.4 montre un exemple de graphe H quand $L = \{l_1, l_2, l_3\}$ et $C = \{(\bar{l}_1, l_2, \bar{l}_3), (l_1, l_2, l_3)\}$. Pour un souci de visibilité, nous avons affiché uniquement les arêtes de satisfiabilité sur le premier graphe et les autres arêtes sur le second graphe.

Assertion 3.4 *Un stable de H de cardinalité $3n + m$ ne contient pas de sommet-fictif. De plus, tous les stables de H ont une cardinalité inférieure ou égale à $3n + m$.*

Preuve. Soit S un stable de H . Dans un premier temps, nous montrons que si S contient un sommet-fictif, alors $|S| \leq 3n + m - 1$. Supposons que S contient un sommet-fictif z . Si $z \in V^3$, comme z est adjacent à tous les sommets de $V^1 \cup V^2$, alors $|S| \leq |V^3| = 3n + m - 1$. Supposons maintenant que $z \in V_1 \cup V_2$. Comme z est incident à tous les sommets de V_3 , alors $S \cap V_3 = \emptyset$. Le graphe induit par $V_1 \cup V_2$ correspond à un graphe biparti possédant $2p$ sommets, avec $p = 3n + m - 1$. Par définition, le graphe $H(V_1 \cup V_2)$ possède un couplage parfait, ce qui implique, par le corollaire 1.3, que tout stable de cardinalité maximum est de taille p . Comme $S \subseteq V_1 \cup V_2$, alors $|S| \leq p$. Par conséquent, si S contient un sommet-fictif alors $|S| < 3n + m$.

Maintenant, supposons que S ne contient pas de sommet-fictif. Alors tous les sommets de S appartiennent aux cycles $\Gamma_i, i = 1, \dots, n$ et aux triangles $T_j, j = 1, \dots, m$. Comme

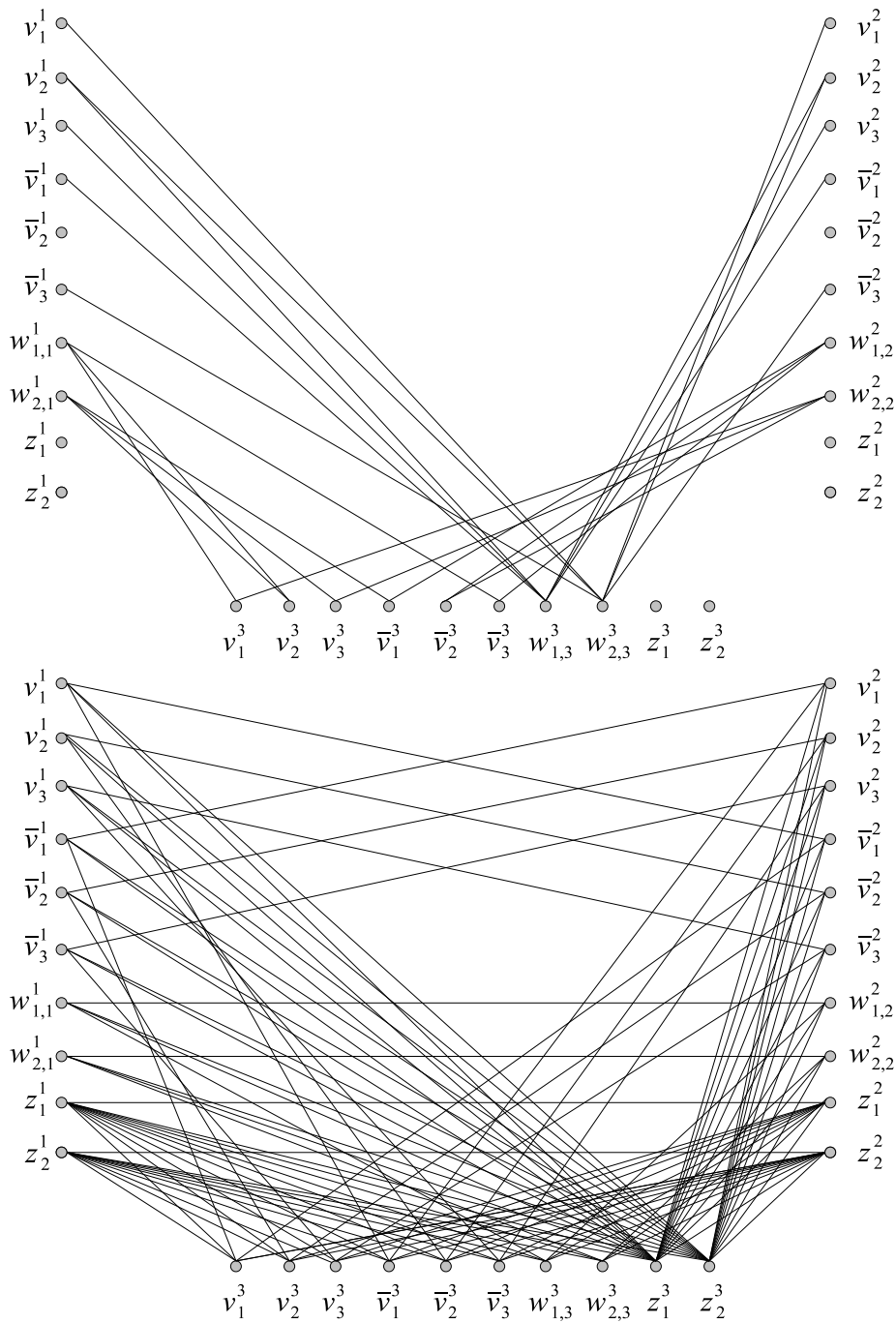


FIGURE 3.4 – Les graphes du PSTCP résultant de l'instance du 1-parmi-3 3SAT $L = \{l_1, l_2, l_3\}$ et $C = \{(\bar{l}_1, l_2, \bar{l}_3), (l_1, l_2, l_3)\}$.

S peut intersecter chaque Γ_i en au plus 3 sommets et chaque triangle en au plus un sommet, il s'ensuit que $|S| \leq 3n + m$. \square

Assertion 3.5 *Il existe dans H un stable de taille $3n+m$ si et seulement si le problème 1-parmi-3 3SAT admet une solution.*

Preuve. (\Rightarrow) Soit S un stable dans H de taille $3n + m$. Par l'assertion 3.4, S ne contient pas de sommet-fictif. Donc, comme $|S| = 3n + m$, S intersecte chaque cycle Γ_i en exactement trois sommets et chaque triangle T_j en exactement un sommet. De plus, nous avons soit $S \cap \Gamma_i = \{v_i^1, v_i^2, v_i^3\}$ soit $S \cap \Gamma_i = \{\bar{v}_i^1, \bar{v}_i^2, \bar{v}_i^3\}$, pour $i = 1, \dots, n$. Considérons la solution I du 1-parmi-3 3SAT définie de la manière suivante. Si $v_i^k \in S$ (resp. $\bar{v}_i^k \in S$), $k = 1, 2, 3$, alors on associe la valeur vraie (resp. fausse) au littéral l_i , pour $i = 1, \dots, n$. Dans ce qui suit nous montrons que pour chaque clause $C_j = (x_r, x_s, x_t)$, nous avons exactement un littéral à vrai. Pour cela il suffit de montrer que chaque sommet-clause de T_j est dans S si et seulement si le littéral correspondant est de valeur vraie. On suppose, sans perte de généralité, que le sommet de T_j appartenant à S est le sommet w_{jr}^1 , $r \in \{1, \dots, n\}$, la preuve pour les autres sommets de T_j étant similaire. La variable associée à w_{jr}^1 est la variable x_r de la clause C_j . Si $x_r = l_r$ (resp. $x_r = \bar{l}_r$), alors le graphe H contient, par construction, l'arête $w_{jr}^1 \bar{v}_r^3$ (resp. $w_{jr}^1 v_r^3$). Comme S est un stable, il s'ensuit que $\bar{v}_r^3 \notin S$ (resp. $v_r^3 \notin S$). Les remarques précédentes impliquent alors que les sommets v_r^1, v_r^2, v_r^3 appartiennent à S . Par construction de I , le littéral l_r est de valeur vraie dans la solution I , x_r est donc de valeur vraie.

Inversement, si $x_r = \text{vrai}$ ($= l_r$), alors par définition de I , $v_r^1, v_r^2, v_r^3 \in S$. De plus, les arêtes de satisfiabilité $w_{js}^2 v_r^3, w_{jt}^3 v_r^2$ appartiennent à F . Comme $|S \cap T_r| = 1$, il s'ensuit que $w_{jr}^1 \in S$.

En conséquence, comme S contient exactement un sommet-clause pour chaque T_i , il s'ensuit que chaque clause possède exactement un littéral à vrai.

(\Leftarrow) Soit I une solution de 1-parmi-3 3SAT. Considérons l'ensemble de sommets S obtenu comme suit :

- Si (dans I) $l_i = \text{vrai}$, alors on ajoute v_i^1, v_i^2, v_i^3 dans S .
Si $l_i = \text{faux}$, alors on ajoute $\bar{v}_i^1, \bar{v}_i^2, \bar{v}_i^3$ dans S .
- Si pour une clause $C_j = (x_r, x_s, x_t)$,
 $x_r = \text{vrai}$, on ajoute w_{jr}^1 dans S ,
 $x_s = \text{vrai}$, on ajoute w_{js}^2 dans S ,
 $x_t = \text{vrai}$, on ajoute w_{jt}^3 dans S .

Comme chaque clause possède exactement un littéral vrai respectant la solution I , nous avons $|S| = 3n+m$. Maintenant, il suffit juste de montrer que S est un stable. Pour cela, il suffit de montrer qu'aucun sommet-littéral de S n'est adjacent aux sommets-clauses de S .

Supposons le contraire, c'est-à-dire qu'il existe un sommet-littéral, disons u , adjacent à un sommet-clause, disons v . Par construction de H , l'arête uv est une arête de satisfiabilité. Sans perte de généralité, supposons que le sommet-clause est le sommet w_{jr}^1 , $j \in \{1, \dots, m\}$ et $r \in \{1, \dots, n\}$ et $C_j = (l_r, l_s, l_t)$, la preuve étant similaire pour les autres cas. Par construction de H , les sommets-littéraux adjacents à w_{jr}^1 sont les sommets \bar{v}_r^3 , v_s^3 et v_t^3 . Par construction de S , il s'ensuit que $l_r = \text{vrai}$ et $l_s = l_t = \text{faux}$, ce qui implique que $v_r^3, \bar{v}_s^3, \bar{v}_t^3 \in S$. Il s'ensuit que \bar{v}_r^3, v_s^3 et $v_t^3 \notin S$, une contradiction avec le fait que S n'est pas un stable. \square

L'assertion 3.5 implique que l'on a une solution du PSTCP si et seulement si on a une solution du 1-parmi-3 3SAT. Comme la construction du PSTCP à partir du 1-parmi-3 3SAT se fait en temps polynomial, on a donc une réduction polynomiale du 1-parmi-3 3SAT en PSTCP. Comme 1-parmi-3 3SAT est NP-complet [32], le PSTCP est NP-complet. \square

Par les théorèmes 3.2 et 3.3, nous déduisons le corollaire suivant.

Corollaire 3.6 *Le PSGSCP est NP-complet.* \square

3.2.3 Une autre preuve du théorème 3.3

Dans cette section, nous proposons une seconde preuve pour montrer que PSTCP est NP-complet. Pour cela, nous donnons une réduction polynomiale à partir du problème du stable.

Rappelons que le problème du stable est défini de la manière suivante. Étant donné un graphe $G = (V, E)$ avec $|V| = n$, $|E| = m$, et un entier $k \geq 0$, existe-t-il un stable de taille supérieure ou égale à k ? Ce problème est NP-complet [32].

La preuve se décompose en deux étapes. Dans un premier temps, on construit à partir de G un graphe triparti $H' = (V'_1 \cup V'_2 \cup V'_3, F')$ où V'_1 et V'_2 sont connectés uniquement par un couplage parfait et $|V'_1| = |V'_2|$. On montre que H' possède un stable de taille

$m + k$ si et seulement si G possède un stable de taille k . Par la suite, on construit à partir de H' un autre graphe triparti $H = (V_1 \cup V_2 \cup V_3, F)$ tel que V_1 et V_2 sont reliés par un couplage parfait et $|V_1| = |V_2| = |V_3| = m + k - 1$. Cette construction consiste simplement à ajouter des sommets dans les sous-ensembles de manière à ce que ces sommets n'appartiennent à aucun stable de taille $m + k$. On montre finalement que H possède un stable de taille $m + k$ si et seulement si H' possède un stable de même taille. Ceci implique alors la NP-complétude du PSTCP.

Construisons à partir de G le graphe triparti $H' = (V'_1 \cup V'_2 \cup V'_3, F')$ de la manière suivante. Posons $V'_1 = \{v_1^1, \dots, v_1^m\}$, $V'_2 = \{v_2^1, \dots, v_2^m\}$ et $V'_3 = V$. Pour $e_i = \{u, v\} \in E$, $i = 1, \dots, m$, on ajoute dans F' les arêtes : uv_1^i , $v_1^i v_2^i$ et $v_2^i v$. On a donc un graphe contenant $2m + n$ sommets et $3m$ arêtes. De plus, on a $|V'_1| = |V'_2| = m$ et les seules arêtes entre V'_1 et V'_2 sont les arêtes $v_1^i v_2^i$, $i = 1, \dots, m$ qui forment un couplage parfait entre V'_1 et V'_2 . La figure 3.5 illustre cette construction.

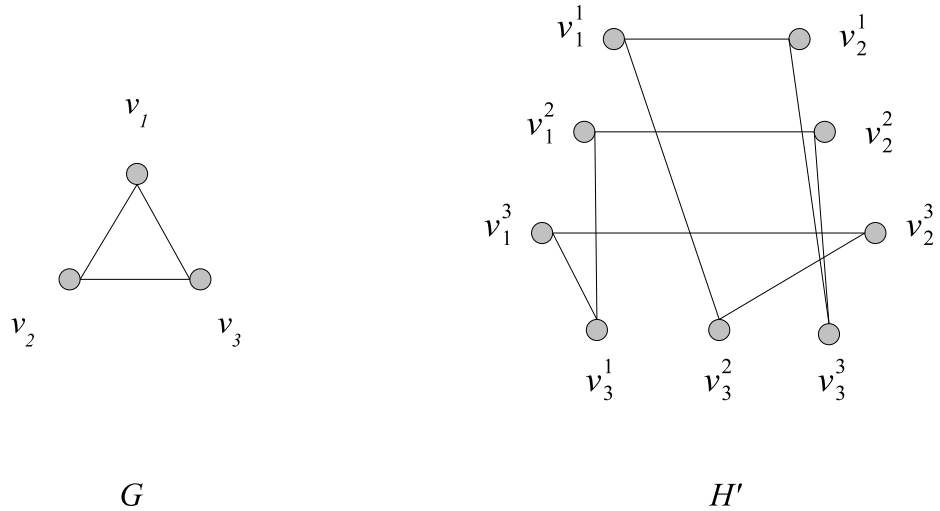


FIGURE 3.5 – Construction de H' à partir du graphe G .

Assertion 3.7 G possède un stable de taille k si et seulement si H' possède un stable de taille $m + k$.

Preuve. (\Rightarrow) On note $E = \{e_1, \dots, e_m\}$. Soit S un stable de G de cardinalité k . Construisons le stable S' de H' de la manière suivante. Posons d'abord $S' = S$. Par ailleurs, comme S est un stable, pour toute arête $e_i = uv \in E$, $i = 1, \dots, m$, au plus un sommet parmi u et v appartient à S . Si u n'appartient pas à S alors on ajoute v_1^i à S' . Sinon, on ajoute v_2^i . Cette construction assure que S' est un stable. De plus, S' est de taille $m + k$.

(\Leftarrow) Soit S' un stable de taille $m + k$ dans H' . Comme V'_1 et V'_2 sont reliés par un couplage, on a $S' \cap \{V'_1, V'_2\} \leq m$. Si $S' \cap \{V'_1, V'_2\} < m$ alors il existe $i \in \{1, \dots, m\}$ tel que $v_1^i, v_2^i \notin S'$, où $v_1^i \in V_1$ et $v_2^i \in V_2$. Comme v_1^i est voisin d'un unique sommet de V'_2 , disons u , on transforme S' en ajoutant v_1^i et en supprimant u dans S' . On peut donc supposer, sans perte de généralité, que $S' \cap \{V'_1, V'_2\} = m$. Le stable $S' \cap \{V'_3\}$ est donc de taille k et correspond à un stable dans G . \square

Construisons maintenant le graphe $H = (V_1 \cup V_2 \cup V_3, F)$ à partir de $H' = (V'_1 \cup V'_2 \cup V'_3, F')$. Posons $V_i = V'_i \cup \{v_i^{m+1}, \dots, v_i^{m+k-1}\}$, $i = 1, 2$ et $V_3 = V'_3 \cup \{v_3^{n+1}, \dots, v_3^{m+k-1}\}$. Posons $F = F'$ et ajoutons dans F les arêtes suivantes.

- Pour tout $i = m + 1, \dots, m + k - 1$ on ajoute les arêtes $v_1^i v_2^i$,
- pour tout $u \in V_3, v \in \{v_i^j : j \in \{m + 1, \dots, m + k - 1\} \text{ et } i \in \{1, 2\}\}$ on ajoute les arêtes uv ,
- pour tout $u \in \{v_3^{n+1}, \dots, v_3^{m+k-1}\}, v \in V'_1 \cup V'_2$ on ajoute les arêtes uv .

H contient alors $3(m + k - 1)$ sommets et $3m + k - 1 + 2(k - 1)(k + m - 1) + 2m(k + m - 1 - n)$ arêtes. De plus on a $|V_1| = |V_2| = |V_3| = m + k - 1$ et les arêtes du graphe induit par V_1 et V_2 forment un couplage parfait.

Assertion 3.8 H possède un stable de taille $m + k$ si et seulement si H' possède un stable de taille $m + k$.

Preuve. Comme $V'_i \subseteq V_i$, $i = 1, 2, 3$ et aucune arête de $F \setminus F'$ n'est incidente à un sommet de $V'_1 \cup V'_2 \cup V'_3$, tout stable de H' est stable de H .

Considérons un stable S de H de taille $m + k$. Par construction, S ne peut pas contenir de sommet de $\{v_3^{n+1}, \dots, v_3^{m+k-1}\}$. Autrement, comme ces sommets sont tous adjacents à tous les sommets de V_1 et de V_2 , on aurait $S \subseteq V_3$, une contradiction avec le fait que $|S| = m + k$. De la même manière, S ne peut pas contenir un sommet de $\{v_i^{m+1}, \dots, v_i^{m+k-1}\}$, $i = 1, 2$. Autrement, comme tous ces sommets sont adjacents à tous les sommets de V_3 , on aurait $S \subseteq V_1 \cup V_2$. Comme il existe un couplage couvrant les sommets de V_1 et V_2 , tout stable inclus dans $V_1 \cup V_2$ est de taille au plus $|V_1| = |V_2| = m + k - 1$, ce qui donne de nouveau une contradiction. Donc tout stable S de H de taille $m + k$ ne contient que des sommets de H' , ce qui implique que S est un stable de H' . \square

Les assertions 3.7 et 3.8 impliquent que l'instance H du PSTCP admet une solution si et seulement si G possède un stable de cardinalité supérieure ou égale à k . Comme la construction du PSTCP à partir du problème du stable se fait en temps polynomial, on a donc une réduction polynomiale du problème du stable en PSTCP. Comme le problème du stable est NP-complet [32], le PSTCP est NP-complet.

3.2.4 Cas polynomial

En conclusion de cette section nous remarquons que le PSGSCP peut être résolu en temps polynomial s'il existe un sommet dans V qui ne possède pas d'arête incidente labellée à vraie/fausse. Supposons en effet qu'il n'existe pas d'arête incidente au sommet $v_j \in V$ appartenant à E^{tf} . Dans ce cas, nous considérons le sous-graphe $H_{\mathcal{F}}$ où \mathcal{F} contient, pour tout $u_i \in U_c$, où $u_i v_j \in E$, les ensembles d'arêtes $E_{u_i}^t$ si $u_i v_j \in E_{u_i}^f$ et $E_{u_i}^f$ autrement. Observons que $E_{\mathcal{F}}$ ne contient aucune arête incidente à v_j . Ceci implique que $H_{\mathcal{F}}$ ne contient pas de couplage parfait. Par conséquent, $H_{\mathcal{F}}$ est solution du PSGSCP. Nous remarquons aussi que, dans ce cas, l'instance H du PSTCP associée au graphe G est telle que le sommet v_j^3 est adjacent à au plus un sommet de v_i^1 et v_i^2 , pour $i = 1, 2, \dots, n$. Une solution du PSTCP est donc obtenue en considérant un stable contenant v_j^3 et, pour $i = 1, 2, \dots, n$, soit v_i^1 , si v_i^1 n'est pas adjacent à v_j^3 , soit v_i^2 .

3.3 Formulation du PSGSCP

Nous formulons le PSGSCP à l'aide d'un programme linéaire en nombres entiers. Soit (G, U_c, π) une instance du PSGSCP avec $G = (U \cup V, E)$, $U_c \subseteq U$ et $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U\}$ une partition de E . Rappelons que le problème consiste à vérifier s'il existe un sous-ensemble $\mathcal{F} \subseteq \mathcal{E}$, où $\mathcal{E} = \{E_u^t, E_u^f : u \in U_c\}$, vérifiant les conditions (i) et (ii) de la définition (3.1). Notons par $x \in \{0, 1\}^{|U_c|}$ le vecteur d'incidence tel que

$$x_u = \begin{cases} 1 & \text{si } E_u^t \in \mathcal{F}, \\ 0 & \text{si } E_u^f \in \mathcal{F}, \end{cases} \quad \text{pour tout } u \in U_c.$$

Le vecteur x représente l'état du système où pour tout $u \in U_c$, la condition associée à u est vraie si $x_u = 1$ et fausse sinon.

Considérons un couplage parfait M de G . Soit $x \in \{0, 1\}^{|U_c|}$ vérifiant $x_u = 1$ si $M \cap E_u^t \neq \emptyset$ et $x_u = 0$ si $M \cap E_u^f \neq \emptyset$. Alors le graphe induit par x contient un couplage parfait, et par conséquent, x satisfait l'équation

$$\sum_{u \in U_c: (E_u^t \cup E_u^{tf}) \cap M \neq \emptyset} x_u + \sum_{u \in U_c: (E_u^f \cup E_u^{tf}) \cap M \neq \emptyset} (1 - x_u) = |U_c|.$$

Il faut noter que les arêtes E^{tf} appartiennent toujours au graphe induit par une solution du PSGSCP. Comme toute solution du PSGSCP induit un graphe sans couplage

parfait, le vecteur x doit alors vérifier la contrainte suivante,

$$\sum_{u \in U_c: (E_u^t \cup E_u^{tf}) \cap M \neq \emptyset} x_u + \sum_{u \in U_c: (E_u^f \cup E_u^{tf}) \cap M \neq \emptyset} (1 - x_u) \leq |U_c| - 1.$$

Considérons par exemple le graphe biparti associé au système (S_1) de la figure 3.1. Soient les deux couplages parfaits donnés par les arêtes en gras dans les figures 3.6 (a) et 3.6 (b). Nous pouvons donc en déduire les deux inégalités suivantes :

$$x_{u_1} - x_{u_3} \leq 0, \quad (3.5)$$

$$x_{u_2} + x_{u_3} \leq 1. \quad (3.6)$$

Par conséquent, le PSGSCP est équivalent au programme linéaire en nombres entiers (Q) suivant.

$$\begin{aligned} & \min 0x \\ & \sum_{u \in U_c: (E_u^t \cup E_u^{tf}) \cap M \neq \emptyset} x_u + \sum_{u \in U_c: (E_u^f \cup E_u^{tf}) \cap M \neq \emptyset} (1 - x_u) \leq |U_c| - 1, \quad \text{pour tout } M \in \mathcal{M}, \end{aligned} \quad (3.7)$$

$$x_u \in \{0, 1\}, \quad \text{pour tout } u \in U_c, \quad (3.8)$$

où \mathcal{M} est l'ensemble des couplages parfaits de G .

Les contraintes (3.7) sont appelées *inégalités de couplages*. Les contraintes (3.8) sont les *contraintes d'intégrités*.

Les contraintes de couplages (3.7) assurent que les solutions du programme (Q) induisent des graphes ne possédant pas de couplage parfait. Donc si le programme (Q) possède une solution, alors il existe un état du système induisant un graphe sans couplage parfait. De plus, si le PSGSCP ne possède pas de sous-ensemble \mathcal{F} respectant les conditions (i)-(ii) de la définition 3.1, alors le programme ci-dessus ne possède pas de solution.

Notons que les inégalités de couplages sont en nombre exponentiels. Comme il sera mentionné dans le chapitre 5, les inégalités de couplages peuvent être séparées en temps polynomial. Cela implique que la relaxation linéaire du programme (Q) peut être résolue en temps polynomial.

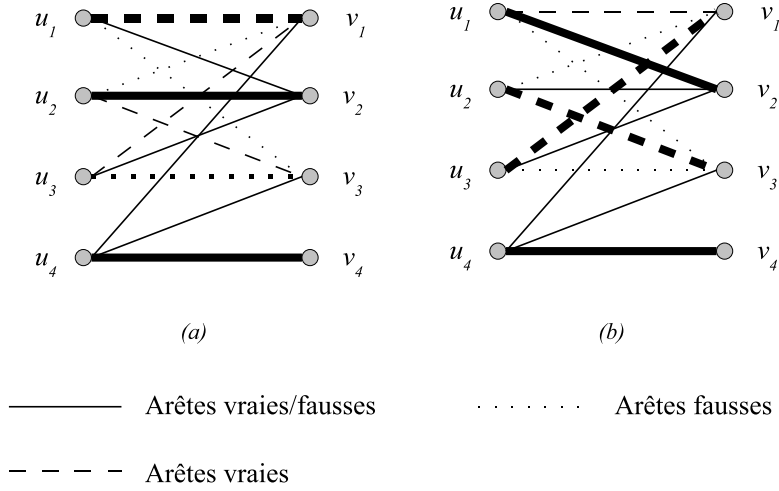


FIGURE 3.6 – Couplages parfaits du graphe biparti associé au système (S_1) .

3.4 Formulation du R-PSGSCP

Le programme linéaire en nombres entiers (Q) que nous avons donné pour le PSGSCP ne possède pas de solution quand le système est structurellement régulier. Dans ce cas, le polyèdre associé est vide et une étude polyédrale ne serait donc pas possible. C'est pour cette raison que nous proposons une seconde formulation donnant lieu à un polyèdre de solutions non vide. Cette seconde formulation s'attache à résoudre la variante relaxée du problème PSGSCP définie comme suit.

Définition 3.9 *Étant donnée une instance (G, U_c, π) où $G = (U \cup V, E)$, $U_c \subseteq U$ et $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U\}$ est une partition de G , le problème relaxé du PSGSCP (R-PSGSCP) consiste à déterminer un sous-ensemble $\mathcal{F} \subseteq \mathcal{E}$ vérifiant la condition (ii) de la définition (3.1) et les conditions*

- (i) *pour tout $u \in U_c$, $E_u^t \notin \mathcal{F}$ et/ou $E_u^f \notin \mathcal{F}$,*
- (iii) *le nombre d'éléments de \mathcal{F} est maximum.*

On remarque que, dans une solution de R-PSGSCP, un sommet $u \in U_c$ peut être uniquement incident, dans $H_{\mathcal{F}}$, à des arêtes de E_u^{tf} . Comme il sera montré ci-dessous les deux problèmes PSGSCP et R-PSGSCP sont équivalents.

Rappelons que nous notons par \mathcal{E} le sous-ensemble de π donné par $\mathcal{E} = \{E_u^t, E_u^f : u \in U_c\}$. Si une solution optimale \mathcal{F} de R-PSGSCP contient $|U_c|$ éléments de \mathcal{E} , alors cette solution est aussi réalisable pour PSGSCP. Si \mathcal{F} possède moins de $|U_c|$ ensembles d'arêtes de \mathcal{E} , alors PSGSCP n'admet pas de solution. La réciproque est aussi vraie. Donc les deux problèmes PSGSCP et R-PSGSCP sont équivalents. Dans la suite nous considérons le problème R-PSGSCP. L'avantage ici est que R-PSGSCP admet un ensemble non vide de solutions.

Nous donnons dans la suite une formulation du R-PSGSCP. Soit (G, U_c, π) une instance du R-PSGSCP. Soit $x \in \{0, 1\}^{|\mathcal{E}|}$ tel que

$$x_F = \begin{cases} 1 & \text{si } F \in \mathcal{F}, \\ 0 & \text{sinon,} \end{cases} \quad \text{pour tout } F \in \mathcal{E}.$$

La valeur $x_{E_u^t} = 1$ (resp. $x_{E_u^f} = 1$) implique que la condition de l'équation associée à u est vraie (resp. fausse).

Considérons un couplage parfait M de G , on note par F_M l'ensemble des éléments de \mathcal{E} intersectant le couplage M . On a alors $F_M = \{F \in \mathcal{E} : F \cap M \neq \emptyset\}$. Étant donné $x \in \{0, 1\}^{|\mathcal{E}|}$, l'inégalité de couplages associée à M s'écrit désormais,

$$\sum_{F \in F_M} x_F \leq |F_M| - 1.$$

On en déduit que le problème R-PSGSCP (et donc PSGSCP) est équivalent au programme linéaire en nombres entiers (P) suivant :

$$\max \sum_{F \in \mathcal{E}} x_F \tag{3.9}$$

$$x_{E_u^t} + x_{E_u^f} \leq 1 \quad \text{pour tout } u \in U_c, \tag{3.10}$$

$$\sum_{F \in F_M} x_F \leq |F_M| - 1 \quad \text{pour tout } M \in \mathcal{M}, \tag{3.11}$$

$$x_F \geq 0 \quad \text{pour tout } F \in \mathcal{E}, \tag{3.12}$$

$$x_F \leq 1 \quad \text{pour tout } F \in \mathcal{E}, \tag{3.13}$$

$$x_F \text{ entier} \quad \text{pour tout } F \in \mathcal{E}. \tag{3.14}$$

où \mathcal{M} est l'ensemble des couplages parfaits de G . Nous appelons dans la suite inégalités de couplages, les contraintes (3.11). Les contraintes (3.10) sont appelées *inégalités de satisfiabilité*. Les inégalités (3.12) et (3.13) sont les inégalités triviales et les inégalités (3.14) sont les contraintes d'intégrité.

Les inégalités de satisfiabilité permettent de vérifier la condition (i) de la définition (3.9). Les inégalités de couplages permettent de satisfaire la condition (ii) de la définition (3.1).

Pour illustrer les contraintes (3.11), considérons le graphe biparti associé au système (S_1) de la figure 3.2. Les contraintes de couplages associées aux graphes données par les arêtes en gras sur les figures 3.6 (a) et 3.6 (b) s'écrivent :

$$x_{E_{u_1}^t} + x_{E_{u_3}^f} \leq 1 \quad (3.15)$$

$$x_{E_{u_2}^t} + x_{E_{u_3}^t} \leq 1 \quad (3.16)$$

La relaxation linéaire du programme (P) peut, elle aussi, être résolue en temps polynomial. En effet, seul les inégalités de couplages sont en nombre exponentiel et, comme nous le verrons dans la suite, le problème de séparation des inégalités de couplages peut être résolu en temps polynomial.

3.5 Conclusion

Dans ce chapitre, nous avons montré que le PASC se ramène au problème de couplages particulier PSGSCP dans un graphe biparti. Nous avons montré par la suite que ce dernier est NP-complet en utilisant une réduction d'une variante du problème du stable dans les graphes tripartis. Pour cela, nous étions amenés à montrer dans un premier temps que cette variante du problème du stable est elle-même NP-complet. Comme conséquence, nous avons obtenu que le PASC est NP-complet. Dans un deuxième temps, nous avons étudié des formulations en nombres entiers pour le problème PSGSCP. Ce travail a donné lieu aux publications [55, 54, 51, 53]

Le chapitre suivant sera dédié à l'étude polyédrale de la formulation (3.9)-(3.14). Nous identifierons des inégalités valides pour notre formulation et nous développerons un algorithme de coupes et branchements pour résoudre ce problème.

Chapitre 4

Étude polyédrale

Dans ce chapitre, on étudie l'enveloppe convexe des solutions du problème de l'analyse structurelle des systèmes algébro-différentiels conditionnels. On caractérise la dimension du polytope des solutions et on donne des conditions nécessaires et suffisantes pour que les contraintes de base de la formulation définissent des facettes. Par la suite, on discute de nouvelles contraintes valides et on étudie également leur aspect facial.

4.1 Polyèdre associé au PSGSCP

Nous allons commencer par quelques rappels et définitions nous permettant d'étudier l'enveloppe convexe des solutions du PSGSCP.

Comme nous l'avons défini dans le chapitre précédent, une instance du PSGSCP est représentée par le triplet (G, U_c, π) où $G = (U \cup V, E)$, $U_c \subseteq U$ et $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U\}$ est une partition de G . Nous rappelons que $\mathcal{E} = \{E_u^t, E_u^f : E_u^t, E_u^f \in \pi \text{ et } u \in U_c\}$. De plus, rappelons que pour un couplage parfait M de G , on note par F_M l'ensemble des éléments de \mathcal{E} contenant une arête de M . On a alors $F_M = \{F \in \mathcal{E} : F \cap M \neq \emptyset\}$. Soit \mathcal{F} un sous-ensemble de \mathcal{E} . On note le graphe associé à \mathcal{F} par $H_{\mathcal{F}} = (U \cup V, E_{\mathcal{F}})$, où $E_{\mathcal{F}} = E^{tf} \cup (\cup_{F \in \mathcal{F}} F)$. Pour $F \in \mathcal{E}$ un ensemble d'arêtes vraies (resp. fausses) incidentes à un sommet $u \in U_c$, on note par \overline{F} l'ensemble d'arêtes fausses (resp. vraies) incidentes au sommet u . Si $F = E_u^t$ (resp. $F = E_u^f$), alors $\overline{F} = E_u^f$ (resp. $\overline{F} = E_u^t$).

Nous rappelons aussi que le problème PSGSCP est équivalent au programme linéaire en nombres entiers P donné par :

$$\max \sum_{F \in \mathcal{E}} x_F$$

$$x_{E_u^t} + x_{E_u^f} \leq 1 \quad \text{pour tout } u \in U_c, \quad (4.1)$$

$$\sum_{F \in F_M} x_F \leq |F_M| - 1 \quad \text{pour tout } M \in \mathcal{M}, \quad (4.2)$$

$$x_F \geq 0 \quad \text{pour tout } F \in \mathcal{E}, \quad (4.3)$$

$$x_F \leq 1 \quad \text{pour tout } F \in \mathcal{E}, \quad (4.4)$$

$$x_F \text{ entier} \quad \text{pour tout } F \in \mathcal{E}, \quad (4.5)$$

où \mathcal{M} est l'ensemble des couplages parfaits de G .

Si $T \subset \mathcal{E}$ est un sous-ensemble d'ensembles d'arêtes, alors le vecteur d'incidence x^T de T est donné par

$$x_F^T = \begin{cases} 1 & \text{si } F \in T, \\ 0 & \text{sinon,} \end{cases} \quad \text{pour tout } F \in \mathcal{E}.$$

Dans la suite du travail, on considérera que l'instance (G, U_c, π) vérifie l'hypothèse suivante

Hypothèse 4.1 *Les graphes associés à $\{F\}$ avec $F \in \mathcal{E}$, ne contiennent pas de couplage parfait.*

En effet, s'il existe $u \in U_c$ tel que le sous-graphe $H_{\{E_u^t\}}$ (resp. $H_{\{E_u^f\}}$) contient un couplage parfait, il est clair que tout sous-graphe respectant la condition (i) de la définition 3.9 et contenant E_u^t (resp. E_u^f) possède un couplage parfait. Dans ce cas, nous cherchons à savoir s'il existe un sous-graphe contenant E_u^f (resp. E_u^t) et respectant les conditions (i)-(ii) de la définition 3.9. Nous pouvons donc transformer cette instance en supprimant E_u^t (resp. E_u^f) et en ajoutant E_u^f (resp. E_u^t) à l'ensemble E^{tf} car cet ensemble apparaît dans tous les sous-graphes. Dans ce cas, l'équation associée au sommet u n'est plus considérée comme une équation conditionnelle. Il est clair que ces transformations peuvent se faire en temps polynomial. La figure 4.1 (a) donne un graphe G tel que le sous-graphe composé des arêtes $E^{tf} \cup E_{u_1}^t$ possède un couplage parfait. La figure 4.1 (b) donne le nouveau graphe résultant de la transformation.

Soit $P(G, U_c, \pi)$ l'enveloppe convexe des solutions du programme (P) , c'est-à-dire,

$$P(G, U_c, \pi) = \text{conv}(\{x \in \{0, 1\}^{\mathcal{E}} \mid x \text{ satisfait (4.1), (4.2)}\}).$$

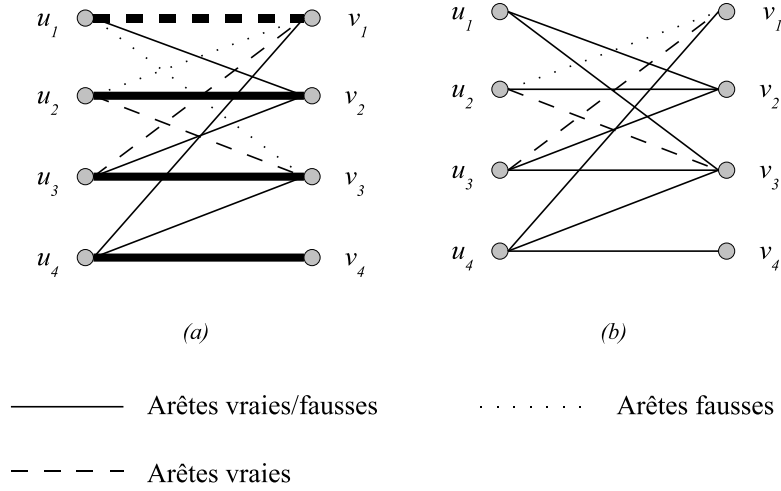


FIGURE 4.1 – Suppression dans G de l'ensemble $E_{u_1}^t$.

4.2 Dimension de $P(G, U_c, \pi)$

Dans cette section, on caractérise la dimension de $P(G, U_c, \pi)$.

Théorème 4.2 $P(G, U_c, \pi)$ est de pleine dimension.

Preuve. Il suffit d'exhiber $|\mathcal{E}| + 1$ sous-ensembles $S_i \subset \mathcal{E}$, $i = 0, \dots, |\mathcal{E}|$, vérifiant les conditions (i) et (ii) de la définition 3.9 et dont les vecteurs d'incidence x^{S_i} , $i = 1, \dots, |\mathcal{E}| + 1$, sont affinement indépendants. Considérons les ensembles $S_0 = \emptyset$ et $S_F = \{F\}$ pour tout $F \in \mathcal{E}$. Ces ensembles contiennent au plus un élément de \mathcal{E} , ce qui implique que la condition (i) est vérifiée. De plus, d'après l'hypothèse 4.1, les graphes associés à chacun de ces ensembles ne contiennent pas de couplage parfait. Nous avons donc $|\mathcal{E}| + 1$ solutions associées aux ensembles S_0 et S_F tel que leurs vecteurs d'incidence x^{S_F} , $F \in \mathcal{E} \cup \{\emptyset\}$, sont affinement indépendants. \square

4.3 Inégalités valides et facettes

Dans cette section, nous allons voir, dans un premier temps, des conditions nécessaires et suffisantes pour que les contraintes de base de la formulation définissent des facettes. Nous introduisons les contraintes dites de couplages à dominance maximale permettant de renforcer les contraintes de couplages. Nous discutons dans la suite de trois nouvelles familles de contraintes valides : les inégalités de couplages proches, les inégalités de k -couplages et les inégalités de sous-graphe régulier.

4.3.1 Inégalités triviales

Dans cette section on considère les contraintes triviales (4.3) et (4.4). Dans un premier temps, on montre que les inégalités (4.3) définissent toujours des facettes.

Théorème 4.3 *Les inégalités triviales (4.3) $x_F \geq 0$ définissent des facettes de $P(G, U_c, \pi)$.*

Preuve. Comme $P(G, U_c, \pi)$ est de pleine dimension, il suffit d'exhiber $|\mathcal{E}|$ solutions tels que leurs vecteurs d'incidence sont affinement indépendants. Considérons les ensembles $S_{F'} = \{F'\}$, $F' \in \mathcal{E} \setminus \{F\}$ et $S_\emptyset = \emptyset$. Ces ensembles contiennent au plus un élément de \mathcal{E} , ce qui implique que la condition (i) est vérifiée. De plus, d'après l'hypothèse 4.1, les graphes associés à chacun de ces ensembles ne contient pas de couplage parfait. Nous avons donc $|\mathcal{E}|$ solutions associées aux ensembles S_\emptyset et S_F tel que leurs vecteurs d'incidence x^{S_F} , $F \in \mathcal{E} \setminus \{F\} \cup \{\emptyset\}$ satisfont $x_F = 0$, sont affinement indépendants. \square

On remarque que les inégalités triviales (4.4) $x_F \leq 1$ sont dominées par les contraintes de satisfiabilités (4.1) $x_F + x_{\bar{F}} \leq 1$. Comme cette contrainte domine la contrainte (4.4) et définit une face de $P(G, U_c, \pi)$ différente de celle définie par la contrainte (4.4), cette dernière ne peut pas définir une facette de $P(G, U_c, \pi)$.

4.3.2 Inégalités de satisfiabilités

Nous discutons ici de conditions nécessaires et suffisantes pour que les contraintes de satisfiabilités définissent des facettes de $P(G, U_c, \pi)$.

Par la suite, étant donné un sommet $u \in U_c$, si $F \in \{E_u^t, E_u^f\}$, alors la contrainte de satisfiabilité associée à u sera également considérée comme la contrainte (4.1) associée à F et \bar{F} .

Théorème 4.4 *L'inégalité $x_F + x_{\bar{F}} \leq 1$ définit une facette de $P(G, U_c, \pi)$ si et seulement si pour tout $F' \in \mathcal{E} \setminus \{F, \bar{F}\}$, au moins un des deux graphes associés aux ensembles $\{F, F'\}$ et $\{\bar{F}, F'\}$ ne contient pas de couplage parfait.*

Preuve. Supposons qu'il existe $F' \in \mathcal{E} \setminus \{F, \bar{F}\}$ tel que les graphes $H_{\{F, F'\}}$ et $\bar{H}_{\{\bar{F}, F'\}}$ contiennent tous les deux un couplage parfait. Dans ce cas, nous avons les contraintes de couplages

$$x_F + x_{F'} \leq 1, \quad (4.6)$$

$$x_{\bar{F}} + x_{F'} \leq 1. \quad (4.7)$$

En additionnant ces contraintes et la contrainte $x_F + x_{\bar{F}} \leq 1$ en divisant par 2 et en arrondissant le second membre à l'entier inférieur, nous obtenons la contrainte

$$x_F + x_{\bar{F}} + x_{F'} \leq 1. \quad (4.8)$$

Cette contrainte domine la contrainte (4.1) et définit une face de $P(G, U_c, \pi)$ différente de celle définie par la contrainte (4.1). Ceci implique que cette dernière ne peut pas définir une facette de $P(G, U_c, \pi)$.

Supposons maintenant que, pour tout $F' \in \mathcal{E} \setminus \{F, \bar{F}\}$, au moins un des deux graphes $H_{\{F, F'\}}$ et $\bar{H}_{\{\bar{F}, F'\}}$ ne contient pas de couplage parfait. Pour tout $F' \in \mathcal{E} \setminus \{F, \bar{F}\}$, soit l'ensemble $S_{F'} = \{F, F'\}$ soit l'ensemble $\bar{S}_{F'} = \{\bar{F}, F'\}$ est solution du $P(G, U_c, \pi)$, ce qui nous donne au moins $|\mathcal{E}| - 2$ solutions. D'après l'hypothèse 4.1, les graphes $H'_{\{F\}}$ et $\bar{H}'_{\{\bar{F}\}}$ ne contiennent pas de couplage parfait. Nous considérons aussi les ensembles $S_F = \{F\}$ et $S_{\bar{F}} = \{\bar{F}\}$. Nous obtenons ainsi $|\mathcal{E}|$ solutions dont leurs vecteurs d'incidence satisfont $x_F + x_{\bar{F}} = 1$ et sont affinement indépendants. \square

4.3.3 Inégalités de couplages

Nous discutons ici de conditions nécessaires et suffisantes pour que les contraintes de couplages (4.2) définissent des facettes de $P(G, U_c, \pi)$.

Proposition 4.5 *Soient M' et M^* deux couplages parfaits de G . Si $F_{M'} \subset F_{M^*}$, alors la contrainte de couplages associée à M' domine celle associée à M^* .*

Preuve. Les contraintes associées aux couplages M' et M^* s'écrivent :

$$\sum_{F \in F_{M'}} x_F \leq |F_{M'}| - 1, \quad (4.9)$$

$$\sum_{F \in F_{M'}} x_F + \sum_{F \in F_{M^*} \setminus F_{M'}} x_F \leq |F_{M'}| + |F_{M^*} \setminus F_{M'}| - 1. \quad (4.10)$$

La contrainte (4.10) est combinaison linéaire des contraintes (4.9) et $x_F \leq 1$, pour tout $F \in F_{M^*} \setminus F_{M'}$. \square

Par abus de langage, on dit qu'un couplage parfait M' domine un couplage parfait M^* si et seulement si la contrainte de couplages associée à M' domine la contrainte de couplages associée à M^* .

L'exemple suivant illustre le résultat de la proposition précédente. Considérons la contrainte de couplages

$$x_{E_{u_1}^t} + x_{E_{u_2}^t} + x_{E_{u_3}^t} \leq 2, \quad (4.11)$$

associée au couplage parfait M^* de la figure 4.2 (a). Cette contrainte est dominée par la contrainte de couplages

$$x_{E_{u_1}^t} + x_{E_{u_2}^t} \leq 1, \quad (4.12)$$

associée au couplage parfait M' de la figure 4.2 (b). En effet, la contrainte (4.11) est une combinaison linéaire de la contrainte (4.12) et de la contrainte triviale $x_{E_{u_3}^t} \leq 1$. Il est facile de voir que $F_{M'} \subset F_{M^*}$.

On appelle *couplage à dominance maximale* un couplage parfait M de G tel qu'il n'existe aucun couplage parfait M' de G avec $F_{M'} \subset F_M$. On appelle *inégalité de couplages à dominance maximale*, une inégalité de couplages (4.2) associée à un couplage à dominance maximale.

On note par \mathcal{M}_D l'ensemble des couplages à dominance maximale de G . Les contraintes de couplages à dominance maximale s'écrivent alors

$$\sum_{F \in F_M} x_F \leq |F_M| - 1, \quad \text{pour tout } M \in \mathcal{M}_D, \quad (4.13)$$

Ces contraintes constituent une sous-famille des contraintes de couplages. Nous allons dans la suite donner des conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes.

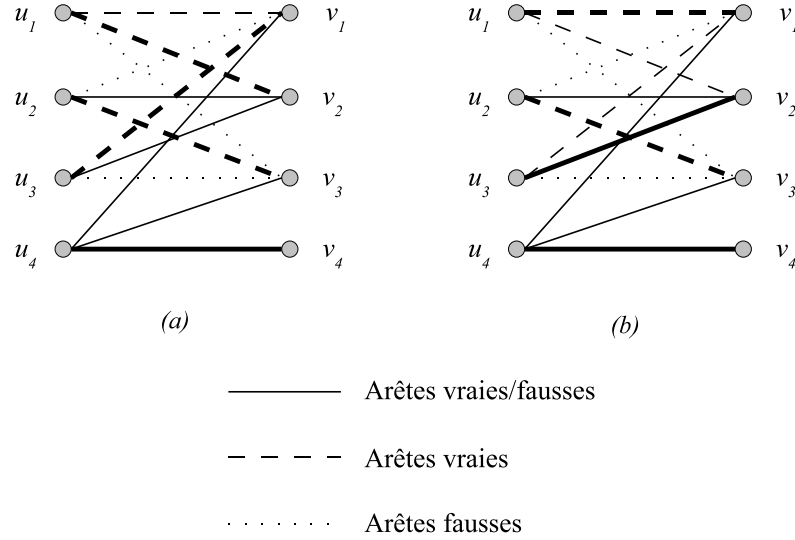


FIGURE 4.2 – Exemple d'un couplage parfait dominé par un autre couplage parfait.

Théorème 4.6 Soit M un couplage parfait à dominance maximale. L'inégalité (4.13) associée à M définit une facette de $P(G, U_c, \pi)$ si et seulement si les conditions suivantes sont vérifiées

- 1) pour tout $F \in F_M$, le graphe associé à $F_M \cup \{\bar{F}\} \setminus \{F\}$ ne contient pas de couplage parfait,
- 2) pour tout F tel que $F, \bar{F} \in \mathcal{E} \setminus F_M$, il existe $F' \in F_M$ tel que le graphe associé à $F_M \cup \{F\} \setminus \{F'\}$ ne contient pas de couplage parfait.

Preuve. (\Rightarrow) Supposons d'abord qu'il existe $F \in F_M$ tel que le graphe associé à $F_M \cup \{\bar{F}\} \setminus \{F\}$ contient un couplage parfait, disons M' . Comme M est un couplage à dominance maximale, on déduit que $F_{M'} = F_M \setminus \{F\} \cup \{\bar{F}\}$. En additionnant les contraintes de couplages (4.13) associées à M et M' et l'inégalité de satisfiabilités (4.1) associée à F et \bar{F} , en divisant par 2 et en arrondissant le second membre à l'entier inférieur, nous obtenons la contrainte

$$\sum_{F' \in F_M} x_{F'} + x_{\bar{F}} \leq |F_M| - 1. \quad (4.14)$$

Il est clair que cette contrainte domine la contrainte de couplage (4.13) associée à M et définit une face propre. Ceci implique que la contrainte (4.13) ne peut pas définir une facette de $P(G, U_c, \pi)$.

Supposons maintenant qu'il existe F , tel que $F, \bar{F} \notin F_M$ et pour tout $F' \in F_M$, le graphe associé à $F_M \cup \{F\} \setminus \{F'\}$ contient un couplage parfait, disons $M_{F'}$. Comme M est de dominance maximale, il s'ensuit que $F_{M_{F'}} = F_M \cup \{F\} \setminus \{F'\}$.

En additionnant les inégalités (4.13) associées aux couplages $M_{F'}$, pour tout $F' \in F_M$ et au couplage M , en divisant par $|F_M|$ et en arrondissant le second membre à l'entier inférieur, nous obtenons la contrainte

$$\sum_{F' \in F_M} x_{F'} + x_F \leq |F_M| - 1. \quad (4.15)$$

Comme cette contrainte domine la contrainte (4.13), cela implique que cette dernière ne définit pas une facette.

(\Leftarrow) Supposons maintenant que les conditions 1) et 2) sont vérifiées. Notons par $ax \leq \alpha$ la contrainte (4.13) associée à M . Soit $bx \leq \beta$ une contrainte définissant une facette de $P(G, U_c, \pi)$. Supposons que $\{x \in P(G, U_c, \pi) : ax = \alpha\} \subseteq \{x \in P(G, U_c, \pi) : bx = \beta\}$. Puisque $P(G, U_c, \pi)$ est de pleine dimension, il suffit de montrer qu'il existe $\rho > 0$ tel que $b = \rho a$.

Soient F_1 et F_2 deux éléments distincts de F_M . Considérons les ensembles $S_1 = F_M \setminus \{F_1\}$, et $S_2 = F_M \setminus \{F_2\}$.

D'après la condition 1), les graphes associés à $S_1 \cup \{\bar{F}_1\}$ et $S_2 \cup \{\bar{F}_2\}$ ne contiennent pas de couplage parfait, ce qui implique que les graphes H_{S_1} et H_{S_2} n'en contiennent pas non plus. Comme les vecteurs d'incidence de S_1 et S_2 vérifient $ax = \alpha$, il s'ensuit que $bx^{S_1} = bx^{S_2}$. Par conséquent, $b(F_1) = b(F_2)$. Comme F_1 et F_2 sont deux éléments quelconques de F_M on déduit que

$$b(F') = b(F^*) = \rho, \quad \text{pour tout } F', F^* \in F_M, \quad (4.16)$$

où ρ est un scalaire dans \mathbb{R} .

Considérons l'ensemble $S_3 = S_1 \setminus \{\bar{F}_1\}$. D'après la condition 1), le graphe H_{S_3} ne contient pas de couplage parfait. De plus, il est clair que S_3 vérifie la condition (i), ce qui implique que S_3 est une solution. Comme $ax^{S_1} = ax^{S_3}$, nous avons $bx^{S_1} = bx^{S_3}$, ce qui implique que $b(\bar{F}_1) = 0$. Comme F_1 est un élément quelconque de F_M on en déduit que

$$b(\bar{F}) = 0, \quad \text{pour tout } F \in F_M. \quad (4.17)$$

Soit F_3 un élément quelconque de $\mathcal{E} \setminus F_M$ tel que $\overline{F_3} \notin F_M$. D'après la condition 2), il existe un élément de F_M , disons F_4 , tel que le graphe associé à $F_M \cup \{F_3\} \setminus \{F_4\}$ ne contient pas de couplage parfait. Considérons les ensembles $S_4 = F_M \cup \{F_3\} \setminus \{F_4\}$ et $S_5 = F_M \setminus \{F_4\}$. Il est clair que S_4 et S_5 sont des solutions. Comme $ax^{S_4} = ax^{S_5}$ et donc $bx^{S_4} = bx^{S_5}$ il en résulte que $b(F_3) = 0$. Comme F_3 est un élément quelconque de $\mathcal{E} \setminus F_M$ on en déduit que

$$b(F) = 0, \quad \text{pour tout } F \notin F_M. \quad (4.18)$$

Par (4.16)-(4.18), il s'ensuit que $b = \rho a$. De plus, comme pour tout ensemble d'arêtes $F \in F_M$, il existe un sous-graphe H contenant F tel que la contrainte soit vérifiée à l'égalité, la face définie par $ax \leq \alpha$ est différente d'une face triviale $\{x \in P(G, U_c, \pi) : x_F = 0\}$. Cela implique que la face induite par $bx \leq \beta$ n'est pas contenue dans une face triviale. Il s'ensuit que $b(F) \geq 0$ pour tout $F \in F_M$. Comme $bx \leq \beta$ définit une facette de $P(G, U_c, \pi)$, il doit donc exister au moins un ensemble d'arêtes $F \in F_M$ tel que $b(F) > 0$. Par conséquent $\rho > 0$ et la preuve est terminée. \square

Nous pouvons remarquer que la condition 1) vérifie que la contrainte de couplages est à dominance maximale.

4.3.4 Inégalités de couplages proches

Dans cette section, nous présentons une nouvelle famille d'inégalités valides pour le PSGSCP. Ces inégalités sont définies par rapport à un couplage parfait et un élément particulier de \mathcal{E} .

Soient (G, U_c, π) une instance du PSGSCP, M' un couplage parfait à dominance maximale de G et F' un élément de $F_{M'}$. Un couplage parfait M^* de G est dit *proche* de M' par rapport à F' si $F_{M^*} \subseteq (F_{M'} \setminus \{F'\}) \cup \{\overline{F'}\}$. On remarque que $\overline{F'}$ appartient à F_{M^*} . En effet, puisque M' est à dominance maximale, il n'existe pas de couplage parfait M tel que $F_M \subset F_{M'}$.

Proposition 4.7 *Soient M' un couplage parfait à dominance maximale de G et $F' \in F_{M'}$. S'il existe un couplage parfait proche de M' par rapport à F' , alors la contrainte associée à M' et F' , donnée par*

$$\sum_{F \in F_{M'}} x_F + x_{\overline{F'}} \leq |F_{M'}| - 1, \quad (4.19)$$

est valide pour le PSGSCP.

Preuve. Notons par M^* le couplage proche de M' par rapport à F' . La somme des contraintes triviales $x_F \leq 1$ pour tout $F \in F_{M'} \setminus (F_{M^*} \cup \{F'\})$ donne la contrainte

$$\sum_{F \in F_{M'} \setminus (F_{M^*} \cup \{F'\})} x_F \leq |F_{M'} \setminus (F_{M^*} \cup \{F'\})|. \quad (4.20)$$

Comme $F_{M'} \cap (F_{M^*} \cup \{F'\}) = (F_{M^*} \setminus \{\bar{F}'\}) \cup \{F'\}$, le membre de droite de la contrainte (4.20) est égal à $|F_{M'}| - |F_{M^*}|$. La somme de la contrainte (4.20), des contraintes de couplages (4.13) associées à M' et M^* et de la contrainte de satisfiabilités associée à F' et \bar{F}' donne la contrainte :

$$2 \sum_{F \in F_{M'}} x_F + 2x_{\bar{F}'} \leq 2|F_{M'}| - 1. \quad (4.21)$$

En divisant cette contrainte par deux et en prenant la partie entière inférieure du membre de droite, on obtient la contrainte (4.19) associée à M' et F' . \square

Les contraintes (4.19) sont appelées *contraintes de couplages proches*. Nous donnons dans le théorème suivant les conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes.

Théorème 4.8 *Soient M' un couplage parfait à dominance maximale et F' un élément de $F_{M'}$. L'inégalité (4.19) associée à M' et F' définit une facette de $P(G, U_c, \pi)$ si et seulement si les conditions suivantes sont vérifiées :*

- 1) *il existe $F^* \in F_{M'} \setminus \{F'\}$ tel que le graphe associé à $F_{M'} \cup \{\bar{F}'\} \setminus \{F', F^*\}$ ne contient pas de couplage parfait,*
- 2) *pour tout $F \in F_{M'} \setminus \{F'\}$ au moins un des graphes associés aux ensembles $F_{M'} \cup \{\bar{F}\} \setminus \{F\}$ et $F_{M'} \cup \{\bar{F}, \bar{F}'\} \setminus \{F, F'\}$ ne possède pas de couplage parfait.*
- 3) *pour tout $F \in \mathcal{E} \setminus F_{M'}$ tel que $\bar{F} \notin F_{M'}$ et le graphe associé à $F_{M'} \cup \{F\} \setminus \{F'\}$ contient un couplage parfait, il existe $F^* \in F_{M'} \setminus \{F'\}$, tel que l'un des graphes associés à $F_{M'} \cup \{F\} \setminus \{F^*\}$ ou $F_{M'} \cup \{\bar{F}', F\} \setminus \{F^*, F'\}$ ne contient pas de couplage parfait.*

Preuve. (\Leftarrow) Supposons d'abord que pour tout $\tilde{F} \in F_{M'}$ différent de F' , le graphe associé à $S_{\tilde{F}} = (F_{M'} \cup \{\bar{F}'\}) \setminus \{F', \tilde{F}\}$ contient un couplage parfait, disons $N_{\tilde{F}}$. La somme de l'inégalité de couplages (4.13) associée au couplage $N_{\tilde{F}}$ et des contraintes triviales (4.4) associées à tout élément de $S_{\tilde{F}} \setminus F_{N_{\tilde{F}}}$, donne

$$\sum_{F \in S_{\tilde{F}}} x_F \leq |S_{\tilde{F}}| - 1. \quad \text{pour tout } \tilde{F} \in F_{M'} \setminus \{F'\} \quad (4.22)$$

En additionnant les contraintes (4.22) pour tout élément \tilde{F} de $F_{M'} \setminus \{F'\}$, $|F_{M'}| - 2$ fois l'inégalité de satisfiabilités (4.1) associée à F' et \overline{F}' , et l'inégalité de couplages proches (4.19) associée à M' et F' , on obtient, après division par $|F_{M'}| - 1$ et arrondi inférieur du membre de droite, la contrainte

$$\sum_{F \in F_{M'}} x_F + 2x_{\overline{F}'} \leq |F_{M'}| - 1. \quad (4.23)$$

Il est clair que la contrainte (4.23) domine la contrainte de couplages proches (4.19) associée à M' et F' et définit une face propre. Cela implique que cette dernière ne peut pas définir une facette.

Supposons maintenant qu'il existe un élément de $F_{M'} \setminus \{F'\}$, disons F_1 , tel que les graphes associés aux ensembles $S_1 = F_{M'} \cup \{\overline{F}_1\} \setminus \{F_1\}$ et $S_2 = F_{M'} \cup \{\overline{F}', \overline{F}_1\} \setminus \{F_1, F'\}$ contiennent un couplage parfait. Notons respectivement par N_1 et N_2 les couplages parfaits de H_{S_1} et H_{S_2} . La somme des contraintes de couplages (4.13) associées aux couplages N_1 et N_2 , des inégalités triviales (4.4) pour tout élément de l'ensemble $(S_1 \setminus F_{N_1}) \cup (S_2 \setminus F_{N_2})$ et de la contrainte de satisfiabilités (4.1) associée à F' et \overline{F}' donne, après division par deux et arrondi inférieur du membre de droite, la contrainte

$$\sum_{F \in F_{M'} \setminus \{F_1\}} x_F + x_{\overline{F}_1} + x_{\overline{F}'} \leq |F_{M'}| - 1. \quad (4.24)$$

En additionnant la contrainte (4.24), la contrainte de couplages proches (4.19) associée à M' et F' et la contrainte de satisfiabilités (4.1) associée à F_1 et \overline{F}_1 , on obtient alors, après division par deux et arrondi, la contrainte

$$\sum_{F \in F_{M'}} x_F + x_{\overline{F}_1} + x_{\overline{F}'} \leq |F_{M'}| - 1. \quad (4.25)$$

La contrainte (4.25) domine la contrainte de couplages proches (4.19) associée à M' et F' . De plus, elle définit une face propre, ce qui implique que cette dernière ne peut pas définir une facette.

Supposons que la condition (3) n'est pas vérifiée. Il existe donc un élément de $F_{M'} \setminus \mathcal{E}$ différent de \overline{F}' , disons F_2 , tel que \overline{F}_2 n'appartient pas à $F_{M'}$ et le graphe associé à $(F_{M'} \cup \{F_2\}) \setminus \{F'\}$ contient un couplage parfait. De plus, pour tout élément F^* de $F_{M'} \setminus \{F'\}$, les graphes associés à $(F_{M'} \cup \{F_2\}) \setminus \{F^*\}$ et $(F_{M'} \cup \{\overline{F}', F_2\}) \setminus \{F', F^*\}$ contiennent chacun un couplage parfait.

Soit $F_3 \in F_{M'} \setminus \{F'\}$. Notons respectivement par S_3^1 et S_3^2 les ensembles $(F_{M'} \cup \{F_2\}) \setminus \{F_3\}$ et $(F_{M'} \cup \{\overline{F}', F_2\}) \setminus \{F', F_3\}$. Par hypothèse, $H_{S_3^1}$ et $H_{S_3^2}$ contiennent

chacun un couplage parfait, disons N_3^1 et N_3^2 . Comme $|S_3^1| = |S_3^2| = |F_{M'}|$, la somme des contraintes de couplages (4.13) associées à N_3^1 et N_3^2 , des contraintes triviales (4.4) pour tout élément de l'ensemble $(S_3^1 \setminus F_{N_3^1}) \cup (S_3^2 \setminus F_{N_3^2})$ et de la contrainte de satisfiabilité (4.1) associée à F' et \overline{F}' donne, après division par deux et arrondi inférieur du membre de droite, la contrainte

$$\sum_{F \in F_{M'} \setminus \{F_3\}} x_F + x_{F_2} + x_{\overline{F}'} \leq |F_{M'}| - 1. \quad (4.26)$$

Notons par N_S un couplage parfait de H_S . En additionnant la contrainte de couplages (4.13) associée à N_S , les contraintes triviales (4.4) associées aux éléments de $S \setminus F_{N_S}$, les contraintes (4.26) associées à tout élément de $F_{M'} \setminus \{F'\}$ et la contrainte de couplages proches (4.19) associée à M' et F' , on obtient, après division par $|F_{M'}|$ et arrondi inférieur du membre de droite, la contrainte

$$\sum_{F \in F_{M'}} x_F + x_{F_2} + x_{\overline{F}'} \leq |F_{M'}| - 1. \quad (4.27)$$

Il est clair que la contrainte (4.27) domine la contrainte de couplages proches (4.19) associée à M' et F' et définit une face propre. Cela implique que cette dernière ne peut pas définir une facette.

(\Leftarrow) Supposons maintenant que les conditions 1) - 3) sont vérifiées. Notons par $ax \leq \alpha$ la contrainte (4.19) associée à M' et $F' \in F_{M'}$. Soit $bx \leq \beta$ une contrainte définissant une facette de $P(G, U_c, \pi)$. Supposons que $\{x \in P(G, U_c, \pi) : ax = \alpha\} \subseteq \{x \in P(G, U_c, \pi) : bx = \beta\}$. Puisque $P(G, U_c, \pi)$ est de pleine dimension, il suffit de montrer qu'il existe $\rho > 0$ tel que $b = \rho a$.

Comme la condition 1) est respectée, il existe $F^* \in F_{M'} \setminus \{F'\}$ tel que le graphe associé à $S^* = (F_{M'} \cup \{\overline{F}'\}) \setminus \{F', F^*\}$ ne contient pas de couplage parfait. Comme M' est un couplage parfait, $F_{M'}$ satisfait la condition (i) de la définition (3.9). D'après la définition de \overline{F}' par rapport à un élément $F \in \mathcal{E}$, il s'ensuit que S^* vérifie aussi la condition (i). S^* est donc une solution du PSGSCP. De plus, comme $|S^* \cap (F_{M'} \cup \{\overline{F}'\})| = |F_{M'}| - 1$, il s'ensuit que x^{S^*} vérifie $ax^{S^*} = \alpha$. L'ensemble $S'^* = F_{M'} \setminus \{F'\}$ est aussi une solution du PSGSCP puisque M' est un couplage à dominance maximale, ce qui implique que $H_{S'^*}$ ne contient pas de couplage parfait. De plus, comme $F' \in F_{M'}$, on a $ax^{S'^*} = \alpha$. On déduit alors que $bx^{S^*} = bx^{S'^*}$, ce qui implique que

$$b(F') = b(\overline{F}') = \rho, \quad (4.28)$$

où ρ est un scalaire dans \mathbb{R} .

Soit F_1 un élément de $F_{M'} \setminus \{F'\}$. Clairement, $S_1 = F_{M'} \setminus \{F_1\}$ correspond à une solution réalisable. De plus, elle vérifie $ax^{S_1} = \alpha$. On a donc $bx^{S'^*} = bx^{S_1}$, ce qui implique que $b(F_1) = b(F')$. Comme F_1 est un élément quelconque de $F_{M'} \setminus \{F'\}$, on déduit alors, avec l'équation (4.28), que

$$b(F) = \rho, \text{ pour tout } F \in F_{M'} \cup \{\overline{F'}\}. \quad (4.29)$$

Soit F_2 un élément de $\mathcal{E} \setminus F_{M'}$ tel que $\overline{F_2} \notin F_{M'}$. Clairement, F_2 est différent de $\overline{F'}$. Considérons l'ensemble $S_2 = (F_{M'} \cup \{F_2\}) \setminus \{F'\}$. Supposons que H_{S_2} ne contient pas de couplage parfait. Dans ce cas, comme $\overline{F_2} \notin F_{M'}$, S_2 est solution du PSGSCP. Comme $F' \in F_{M'}$, $ax^{S_2} = \alpha$. On a alors $bx^{S_2} = bx^{S'^*}$, ce qui implique que $b(F_2) = 0$. Supposons maintenant que le graphe H_{S_2} contient un couplage parfait. La condition 3) du théorème implique alors qu'il existe un élément de $F_{M'} \setminus \{F'\}$, disons F_3 , tel que le graphe associé à $S_3 = (F_{M'} \cup \{F_2\}) \setminus \{F_3\}$ ou le graphe associé à $S'_3 = (F_{M'} \cup \{F_2, \overline{F'}\}) \setminus \{F_3, F'\}$ ne contient pas de couplage parfait. Comme $\overline{F_2} \notin F_{M'}$, il est clair que S_3 et S'_3 vérifient la condition (i) de la définition (3.9). De plus, on a $|S_3 \cap (F_{M'} \cup \{\overline{F'}\})| = |S'_3 \cap (F_{M'} \cup \{\overline{F'}\})| = |F_{M'}| - 1$. Il s'ensuit que S_3 et/ou S'_3 est solution du PSGSCP. Supposons que S_3 est solution. Dans ce cas, comme on a $ax^{S_3} = \alpha$, il s'ensuit que $bx^{S_3} = bx^{S'^*}$, ce qui implique que $b(F') + b(F_2) = b(F_3)$. D'après l'équation (4.29) il s'ensuit que $b(F_2) = 0$. Supposons maintenant que S'_3 est une solution. Comme on a $bx^{S'_3} = bx^{S'^*}$, on déduit que $b(F_2) + b(\overline{F'}) = b(F_3)$, ce qui implique, d'après l'équation (4.29), que $b(F_2) = 0$. Comme F_2 est un élément quelconque de $\mathcal{E} \setminus F_{M'}$ tel que $\overline{F_2} \notin F_{M'}$, on déduit que

$$b(F) = 0 \text{ pour tout } F \in \mathcal{E} \setminus F_{M'} \text{ tel que } \overline{F} \notin F_{M'}. \quad (4.30)$$

Considérons finalement un élément de $\mathcal{E} \setminus F_{M'}$ différent de $\overline{F'}$, disons F_4 , tel que $\overline{F_4} \in F_{M'}$. Considérons les ensembles $S_4 = F_{M'} \cup \{F_4\} \setminus \{\overline{F_4}\}$ et $S'_4 = F_{M'} \cup \{F_4, \overline{F'}\} \setminus \{\overline{F_4}, F'\}$. D'après la condition 2) le graphe H_{S_4} et/ou $H_{S'_4}$ ne contient pas de couplage parfait. Il est clair que S_4 et S'_4 vérifient la condition (i) de la définition (3.9). De plus, on a $|S_4 \cap (F_{M'} \cup \{\overline{F'}\})| = |S'_4 \cap (F_{M'} \cup \{\overline{F'}\})| = |F_{M'}| - 1$. Il s'ensuit que S_4 et/ou S'_4 est solution du PSGSCP. Supposons que S_4 est solution. Dans ce cas, comme on a $ax^{S_4} = \alpha$, il s'ensuit que $bx^{S_4} = bx^{S'^*}$, ce qui implique que $b(F_4) + b(\overline{F}) = b(\overline{F_4})$. D'après l'équation (4.29), il s'ensuit que $b(F_4) = 0$. Supposons maintenant que S'_4 est une solution. Comme on a $bx^{S'_4} = bx^{S'^*}$, on déduit que $b(F_4) + b(\overline{F'}) = b(\overline{F_4})$, ce qui implique, d'après l'équation (4.29), que $b(F_4) = 0$. Comme F_4 est un élément quelconque de $\mathcal{E} \setminus F_{M'}$ différent de $\overline{F'}$ tel que $\overline{F_4} \in F_{M'}$, on déduit que

$$b(F) = 0 \text{ pour tout } F \in \mathcal{E} \setminus F_{M'}, F \neq \overline{F'} \text{ tel que } \overline{F} \in F_{M'}. \quad (4.31)$$

D'après les équations (4.28)-(4.31), il s'ensuit que $b = \rho a$. De plus, comme pour tout ensemble d'arêtes $F \in F_{M'} \cup \{\overline{F'}\}$, on a exhibé une solution contenant F et dont le vecteur d'incidence vérifie la contrainte à l'égalité. Il s'ensuit que la face définie par $ax \leq \alpha$ est différente d'une face triviale $\{x \in P(G, U_c, \pi) : x_F = 0\}$. Cela implique que la face induite par $bx \leq \beta$ n'est pas contenue dans une face triviale. On déduit que $b(F) \geq 0$ pour tout $F \in F_{M'} \cup \{\overline{F'}\}$. Comme $bx \leq \beta$ définit une facette de $P(G, U_c, \pi)$, il doit donc exister au moins un ensemble d'arêtes $F \in F_{M'} \cup \{\overline{F'}\}$ tel que $b(F) > 0$. Par conséquent, $\rho > 0$ et la preuve est terminée. \square

4.3.5 Inégalités de k -couplages

Dans ce qui suit, on note par Θ l'ensemble des sous-ensembles de \mathcal{E} différents de l'ensemble vide vérifiant la condition (i) de la définition (3.9). Autrement dit, un sous-ensemble non-vide $\theta \subseteq \mathcal{E}$ appartient à Θ s'il n'existe pas un sommet $u \in U_c$ tel que $\{E_u^t, E_u^f\} \subseteq \theta$. Soient $\theta \in \Theta$, $\theta' \subseteq \theta$ et k un entier appartenant à $\{1, \dots, |\theta'| + 1\}$. Le graphe H_θ est dit k -couplages par rapport à θ' si après la suppression de n'importe quels $k - 1$ ensembles d'arêtes de θ' dans H_θ , le graphe résultant contient toujours un couplage parfait.

Proposition 4.9 *Soient $\theta \in \Theta$, $\theta' \subseteq \theta$ et $k \in \{1, \dots, |\theta'| + 1\}$. Si le graphe H_θ est k -couplages par rapport à θ' alors la contrainte associée à θ' , θ et k , donnée par*

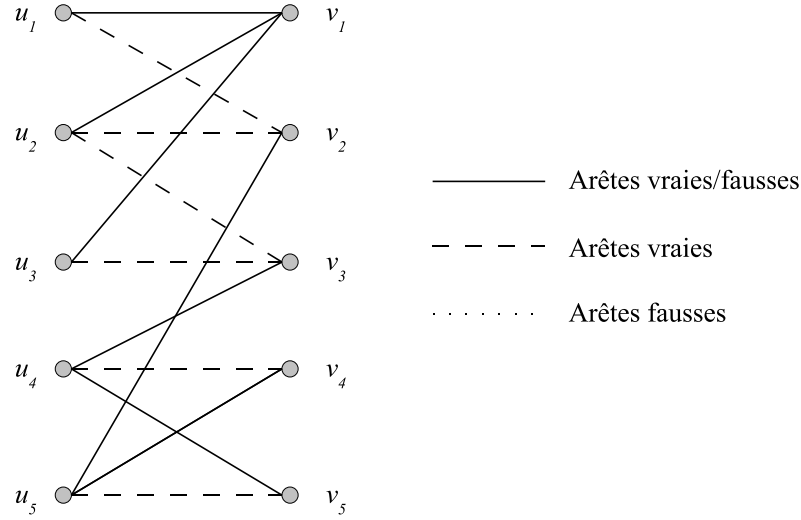
$$\sum_{F \in \theta'} x_F + k \sum_{F \in \theta \setminus \theta'} x_F \leq |\theta'| + k(|\theta \setminus \theta'| - 1), \quad (4.32)$$

est valide pour $P(G, U_c, \pi)$.

Preuve. Supposons qu'il existe une solution S de $P(G, U_c, \pi)$ telle que le vecteur d'incidence x^S ne satisfait pas (4.32), c'est-à-dire x^S vérifie

$$\sum_{F \in \theta'} x_F^S + k \sum_{F \in \theta \setminus \theta'} x_F^S > |\theta'| + k(|\theta \setminus \theta'| - 1) \quad (4.33)$$

Notons par θ^* les éléments de θ' n'appartenant pas à S . Comme $x^S \in \{0, 1\}^\mathcal{E}$, on en déduit que $\sum_{F \in \theta'} x_F^S \leq |\theta'|$ et $k \sum_{F \in \theta \setminus \theta'} x_F^S \geq k(|\theta \setminus \theta'| - 1) + 1$. Par conséquent $\sum_{F \in \theta \setminus \theta'} x_F^S = |\theta \setminus \theta'|$. On en déduit que $\sum_{F \in \theta'} x_F^S \geq |\theta'| - k + 1$. Cela implique que $|\theta^*| \leq k - 1$. Par définition des graphes k -coupages, le graphe $H_{\theta \setminus \theta^*}$ contient un couplage parfait. Comme $\theta \setminus \theta^* \subseteq S$, H_S contient un couplage parfait nous avons donc une contradiction. \square

FIGURE 4.3 – Graphe H_θ .

Nous appelons les contraintes (4.32) *inégalités de k -couplages*.

Par exemple, considérons le graphe G de la figure 4.3. Posons $\theta = \{E_{u_1}^t, E_{u_2}^t, E_{u_3}^t, E_{u_4}^t, E_{u_5}^t\}$ et $\theta' = \{E_{u_1}^t, E_{u_2}^t, E_{u_3}^t\}$. Le graphe H_θ est 2-couplages par rapport à θ' . En effet, le graphe H_θ possède un couplage parfait même si on supprime $E_{u_1}^t, E_{u_2}^t$ ou $E_{u_3}^t$. On en déduit que la contrainte de 2-couplages

$$x_{E_{u_1}^t} + x_{E_{u_2}^t} + x_{E_{u_3}^t} + 2x_{E_{u_4}^t} + 2x_{E_{u_5}^t} \leq 5$$

est valide.

Nous donnons dans la suite des conditions nécessaires pour que les inégalités de k -couplages définissent des facettes.

Proposition 4.10 Soient $\theta \in \Theta$, $\theta' \subseteq \theta$ et $k \in \{1, \dots, |\theta'| + 1\}$ tels que H_θ est k -couplages par rapport à θ' . La contrainte de k -couplages associée à θ' , θ et k ne définit pas une facette si une des conditions est vérifiée :

- 1) H_θ est k' -couplages par rapport à θ' , avec $k' > k$,
- 2) il existe $F \in \theta \setminus \theta'$ tel que H_θ est k -couplages par rapport à $\theta' \cup \{F\}$,

3) il existe $\tilde{\theta} \subset \theta$ et $\tilde{\theta}' \subseteq \tilde{\theta}$ tels que $\tilde{\theta} \setminus \tilde{\theta}' \subseteq \theta \setminus \theta'$, $\tilde{\theta}' \subseteq \theta'$ et $H_{\tilde{\theta}}$ est k -couplages par rapport à $\tilde{\theta}'$.

Preuve. Supposons que la condition 1) est vérifiée. Dans ce cas, la contrainte de k -couplages associée à θ' , θ et k' est valide pour le PSGSCP. Cela implique que les vecteurs d'incidence des solutions appartenant à la face induite par la contrainte de k -couplages associées à θ' , θ et k , disons F_k , vérifient alors $\sum_{F \in \theta'} x_F = |\theta'|$ et $\sum_{F \in \theta \setminus \theta'} x_F = |\theta \setminus \theta'| - 1$. Ils vérifient donc à l'égalité la contrainte de k -couplages associées à θ' , θ et k' , ce qui implique qu'ils appartiennent à la face induite par cette contrainte disons $F_{k'}$. De plus, il est clair que $F_{k'}$ est différent de $P(G, U_c, \pi)$. Comme les contraintes associées à θ' , θ et k et θ' , θ et k' , ne sont pas équivalentes, F_k ne peut pas définir une facette.

Supposons que la condition 2) est vérifiée, c'est-à-dire qu'il existe $F \in \theta \setminus \theta'$ tel que H_{θ} est k -couplages par rapport à $\theta' \cup \{F\}$. La contrainte de k -couplages (4.32) associée à θ' , θ et k est combinaison linéaire de la contrainte de k -couplages associée à $\theta' \cup \{F\}$, θ et k et de $k-1$ fois la contrainte triviale (4.4) associée à F . La contrainte (4.32) ne peut pas définir une facette de $P(G, U_c, \pi)$.

Supposons que la condition 3) est respectée, alors il existe $\tilde{\theta} \subset \theta$ et $\tilde{\theta}' \subseteq \tilde{\theta}$ tels que $\tilde{\theta} \setminus \tilde{\theta}' \subseteq \theta \setminus \theta'$, $\tilde{\theta}' \subseteq \theta'$ et $H_{\tilde{\theta}}$ est k -couplages par rapport à $\tilde{\theta}'$. La contrainte de k -couplages (4.32) associée à θ' , θ et k est combinaison linéaire de la contrainte de k -couplages associée à $\tilde{\theta}'$, $\tilde{\theta}$ et k , des contraintes triviales (4.4) associées aux éléments de $\theta' \setminus \tilde{\theta}'$ et de k fois les contraintes triviales (4.4) associées aux éléments de $(\theta \setminus \theta') \setminus (\tilde{\theta} \setminus \tilde{\theta}')$. La contrainte (4.32) ne peut pas définir une facette de $P(G, U_c, \pi)$. \square

On considère maintenant le cas où $\theta' = \theta$ et $k = 2$. Une contrainte de k -couplages associée à θ , $\theta' = \theta$ et $k = 2$ est appelée contrainte de 2-couplages associée à θ . Elle correspond donc à

$$\sum_{F \in \theta} x_F \leq |\theta| - 2, \quad (4.34)$$

Proposition 4.11 *Étant donné $\theta \in \Theta$ tel que H_{θ} est 2-couplages par rapport à θ , la contrainte de 2-couplages (4.34) associée à θ ne définit pas une facette de $P(G, U_c, \pi)$ si une des conditions suivantes est vérifiée :*

1) *il existe une partition $\{\theta^1, \theta^2, \theta^3\}$ de θ telle que les graphes $H_{\theta^1 \cup \theta^2}$ et $H_{\theta^1 \cup \theta^3}$ sont 2-couplages par rapport à θ^1 ,*

2) il existe une partition $\{\theta^1, \theta^2\}$ de θ telle que le graphe $H_{\theta^1 \cup \theta^2}$ est 3-couplages par rapport à θ^1 et le graphe induit par θ_1 contient un couplage parfait.

Preuve. Supposons que la condition 1) est respectée, alors il existe une partition $\{\theta^1, \theta^2, \theta^3\}$ de θ telle que les graphes $H_{\theta^1 \cup \theta^2}$ et $H_{\theta^1 \cup \theta^3}$ sont 2-couplages par rapport θ^1 . La contrainte 2-couplages (4.34) associée à θ est combinaison linéaire des contraintes de k -couplages associées à θ^1 , $\theta^1 \cup \theta^2$, $k = 2$ et θ^1 , $\theta^1 \cup \theta^3$ et $k = 2$. La contrainte (4.34) ne peut pas définir une facette de $P(G, U_c, \pi)$.

Supposons que la condition 2) est respectée, alors il existe une partition $\{\theta^1, \theta^2\}$ de θ telle que le graphe $H_{\theta^1 \cup \theta^2}$ est 3-couplages par rapport à θ^1 et le graphe induit par θ_1 contient un couplage parfait, disons M_1 . La somme de la contrainte de couplages (4.13) associée à M_1 , des contraintes (4.4) associées à chaque élément de $\theta^1 \setminus F_{M_1}$ et de deux fois la contrainte de k -couplages (4.32) associée à θ^1 , $\theta^1 \cup \theta^2$ et $k = 3$ donne, après division par trois et arrondi inférieur du membre de droite, la contrainte

$$\sum_{F \in \theta^1} x_F + 2 \sum_{F \in \theta^2} x_F \leq |\theta^1| + 2|\theta^2| - 3 \quad (4.35)$$

La contrainte 2-couplages (4.34) associée à θ est combinaison linéaire de la contrainte (4.35), de la contrainte de couplages (4.13) associée à M_1 et des contraintes triviales (4.4) associées aux éléments $\theta^1 \setminus F_{M_1}$. La contrainte de 2-couplages (4.34) ne peut pas définir une facette de $P(G, U_c, \pi)$. \square

Remarquons qu'il est possible d'étendre la condition 1) de la proposition 4.11 pour les contraintes de k -couplages associées à θ , $\theta' = \theta$ et k , avec $2 < k < |\theta| + 1$. Considérons une partition de θ en $k + 1$ sous-ensembles. Notons ces ensembles θ^i , $i \in \{0, \dots, k\}$. Si les graphes $H_{\theta^0 \cup \theta^j}$, $j = 1, \dots, k$, sont k -couplages par rapport à θ^0 , alors la contrainte de k -couplages associées à θ , $\theta' = \theta$ et k est combinaison linéaire des contraintes de k -couplages (4.34) associées à θ^0 , θ^j et k , pour tout $j = 1, \dots, k$.

4.3.6 Inégalités de sous-graphe régulier

La famille de contraintes que nous allons présenter dans la suite est définie par deux sous-graphes particuliers. Considérons dans un premier temps l'ensemble \mathcal{F} représentant un sous-ensemble de \mathcal{E} restreint sur un ensemble de sommets $U'_c \subseteq U_c$. Soient $U' \subseteq U$, $V' \subseteq V$, $U'_c = U_c \cap U'$ et $\mathcal{F}_{U'_c} \subseteq \{E_u^t, E_u^f : u \in U'_c\}$, tels que $|U'| = |V'|$ et $|\mathcal{F}_{U'_c} \cap \{E_u^t, E_u^f\}| = 1$ pour tout $u \in U'_c$. On note par $\Gamma_{U'_c}$ l'ensemble des sous-ensembles

$\mathcal{F}_{U'_c}$. On dit qu'un sous graphe $G' = G(U' \cup V')$ est un *sous-graphe régulier* si pour tout $\gamma \in \Gamma_{U'_c}$ le graphe $H_\gamma(U' \cup V')$ possède un couplage parfait. On appelle $G^* = G(U^* \cup V^*)$ où $U^* = U \setminus U'$ et $V^* = V \setminus V'$ le graphe complémentaire de G' .

Remarquons que le sous-système associé au sous-graphe G' est régulier.

Proposition 4.12 *Soient une instance du PSGSCP (G, U_c, π) , G' un sous-graphe régulier, G^* le graphe complémentaire de G' et M un couplage parfait de G^* . Alors la contrainte*

$$\sum_{u \in U'_c} (x_{E_u^t} + x_{E_u^f}) + \sum_{F \in F_M} x_F \leq |U'_c| + |F_M| - 1 \quad (4.36)$$

est valide.

Preuve. Supposons qu'il existe une solution S de $P(G, U_c, \pi)$ telle que le vecteur d'incidence x^S ne satisfait pas (4.36), c.a.d. x^S vérifie

$$\sum_{u \in U'_c} (x_{E_u^t}^S + x_{E_u^f}^S) + \sum_{F \in F_M} x_F^S > |U'_c| + |F_M| - 1 \quad (4.37)$$

Comme $x^S \in \{0, 1\}^\mathcal{E}$ on en déduit que $\sum_{u \in U'_c} (x_{E_u^t}^S + x_{E_u^f}^S) \leq |U'_c|$ et $\sum_{F \in F_M} x_F^S \geq |F_M|$. Par conséquent $\sum_{F \in F_M} x_F^S = |F_M|$. On déduit que $\sum_{u \in U'_c} (x_{E_u^t}^S + x_{E_u^f}^S) = |U'_c|$. Par définition les graphes $H_\gamma(U' \cup V')$, pour tout $\gamma \in \Gamma_{U'_c}$, possèdent un couplage parfait. Comme M est un couplage parfait de G^* , tous les sous-graphes $H_{\gamma \cup F_M}$, avec $\gamma \in \Gamma_{U'_c}$, possèdent un couplage parfait. Nous avons donc une contradiction. \square

Par exemple considérons le graphe G de la figure (4.4). Le sous-graphe G' est régulier, de plus le graphe G^* possède un couplage parfait, donné par les arêtes en gras. On en déduit la contrainte valide

$$x_{E_{u_1}^t} + x_{E_{u_1}^f} + x_{E_{u_2}^t} + x_{E_{u_2}^f} + x_{E_{u_3}^t} + x_{E_{u_3}^f} + x_{E_{u_4}^t} + x_{E_{u_5}^t} \leq 4. \quad (4.38)$$

4.4 Conclusion

Dans ce chapitre, nous avons donné les conditions nécessaires et suffisantes pour que les contraintes du programme linéaire en nombres entiers (P) définissent des facettes

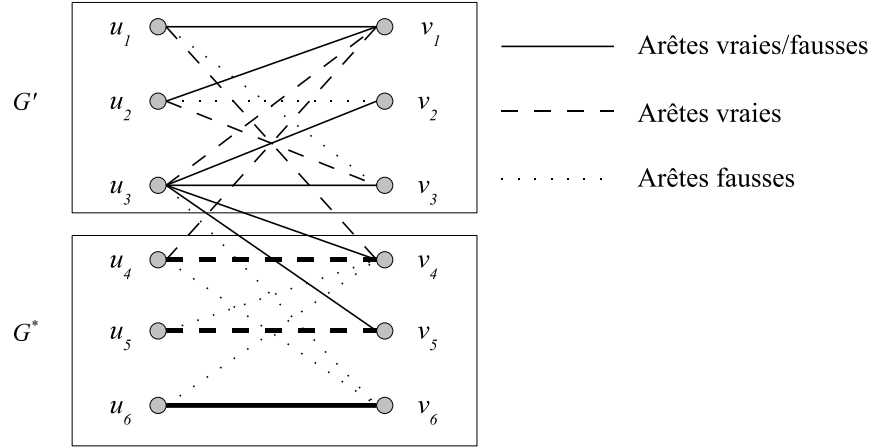


FIGURE 4.4 – Exemple de graphe G avec G' régulier et G^* le graphe complémentaire de G' .

du polytope des solutions. Nous avons présenté les contraintes de couplages à dominance maximale définissant une sous-famille des contraintes de couplages. Nous avons ensuite donné les conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes. Nous avons ensuite donné de nouvelles contraintes valides. Nous avons présenté les contraintes de couplages proches pour lesquelles nous avons donné les conditions nécessaires et suffisantes pour qu'elles définissent des facettes du polytope. Nous avons ensuite discuté des contraintes de k -couplages pour lesquelles nous avons donné plusieurs conditions nécessaires pour qu'elles définissent des facettes du polytope. Enfin nous avons présenté les contraintes de sous-graphe régulier. Ce travail a donné lieu à la publication [52].

Dans le chapitre suivant, nous présentons les méthodes permettant de séparer certaines de ces contraintes et nous développons un algorithme de coupes et branchements basé sur ces méthodes de séparation.

Chapitre 5

Algorithme de coupes et branchements pour le PSGSCP

Ce chapitre présente un algorithme de coupes et branchements pour résoudre le problème du sous-graphe sans couplage parfait. Cet algorithme est basé sur les résultats polyédraux du chapitre précédent. Ce travail a pour objectif de montrer l'intérêt de l'étude polyédrale réalisée, l'efficacité des contraintes trouvées lors de cette étude et d'obtenir un algorithme exacte pour le problème de l'analyse structurelle des systèmes algébro-différentiels conditionnels.

Les algorithmes présentés dans ce chapitre s'appliquent à une instance du PSGSCP (G, U_c, π) , avec $G = (U \cup V, E)$ un graphe biparti, $U_c \subseteq U$ et $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U\}$ une partition de E . Rappelons que nous notons par $E^{tf} = \cup_{u \in U} E_u^{tf}$ et par \mathcal{E} le sous-ensemble de π donné par $\mathcal{E} = \{E_u^t, E_u^f : u \in U_c\}$. De plus, étant donné un couplage parfait M de G , on note par F_M l'ensemble des éléments de \mathcal{E} contenant une arête de M , c'est-à-dire $F_M = \{F \in \mathcal{E} : F \cap M \neq \emptyset\}$.

Dans la section 1, nous présentons différents algorithmes permettant de séparer les contraintes valides pour le PSGSCP présentées dans les chapitres précédents. Nous présentons ensuite dans la section 2 l'algorithme de coupes et branchements développé pour résoudre le problème. Nous terminons par la présentation de résultats expérimentaux obtenus pour différentes variantes de l'algorithme de coupes et branchements.

5.1 Séparation des contraintes

Dans cette section, nous proposons des algorithmes de séparation pour les inégalités de couplages (3.11), les inégalités de couplages à dominance maximale (4.13), les inégalités de couplages proches (4.19) et les inégalités de k -couplages (4.32), lorsque $k = 2$.

Dans les algorithmes de séparation, on notera par $x^* \in \mathbb{R}_+^{|\mathcal{E}|}$ le vecteur sur le quel s'applique la séparation.

5.1.1 Séparation des inégalités de couplages

Comme les contraintes de couplages appartiennent à la formulation, il est nécessaire de les séparer de manière exacte afin de s'assurer de la réalisabilité des solutions. En effet, une solution entière est réalisable seulement si elle satisfait toutes les contraintes de couplages.

Théorème 5.1 *Les inégalités de couplages (3.11) peuvent être séparées en $O(nm \log(n))$.*

Preuve. Posons $x' = 1 - x^*$. Les contraintes de couplages (3.11) pour x^* sont donc équivalentes à

$$\sum_{F \in \mathcal{F}_M} x'_F \geq 1. \quad (5.1)$$

Considérons le vecteur de poids $w \in \mathbb{R}_+^{|E|}$ associé aux arêtes du graphe G défini par :

$$w_{uv} = \begin{cases} x'_F & \text{si } uv \in F, \text{ avec } F \in \mathcal{E}, \\ 0 & \text{sinon,} \end{cases} \quad \text{pour tout } uv \in E.$$

Comme π est une partition de E , alors w est défini de manière unique et $w_{uv} = 0$ pour tout arête de E^{tf} . Soit $w(M)$ la valeur du couplage parfait M dans G de poids minimum par rapport à w . Si $w(M)$ est inférieur à 1, alors la contrainte (3.11) associée au couplage parfait M est violée. Autrement, si $w(M)$ est supérieur ou égal à 1, comme

M est le couplage parfait de poids minimum, on en déduit que toutes les contraintes de couplages (3.11) sont satisfaites. D'après le théorème 1.4, le problème de couplages parfaits de poids minimum peut être résolu en $O(nm \log(n))$. Les inégalités de couplages (3.11) peuvent donc être séparées en $O(nm \log(n))$. \square

Nous donnons maintenant un exemple de séparation des contraintes de couplages (3.11). Considérons l'instance (G, U_c, π) du PSGSCP où $G = (U \cup V, E)$ est le graphe donné par la figure 5.1, $U_c = \{u_1, u_2, u_3, u_4\}$ et π se déduit des labels vrais et faux sur les arêtes indiquées sur la figure 5.1. Soit $x^* \in [0, 1]^8$ une solution fractionnaire définie par $x_{E_{u_1}^t}^* = 1$, $x_{E_{u_2}^t}^* = 1$, $x_{E_{u_3}^t}^* = 0.6$, $x_{E_{u_4}^t}^* = 1$, $x_{E_{u_1}^f}^* = 0$, $x_{E_{u_2}^f}^* = 0$, $x_{E_{u_3}^f}^* = 0.4$ et $x_{E_{u_4}^f}^* = 0$. Le problème de séparation des contraintes de couplages se ramène alors à déterminer dans G un couplage parfait de poids minimum par rapport au vecteur w défini par :

$$w_{uv} = \begin{cases} 0 & \text{si } uv \in \{u_1v_1, u_1v_2, u_2v_1, u_2v_2, u_4v_4, u_5v_3, u_5v_4, u_5v_5\}, \\ 0.4 & \text{si } uv \in \{u_3v_3\}, \\ 0.6 & \text{si } uv \in \{u_3v_2, u_3v_5\}, \\ 1 & \text{sinon.} \end{cases}$$

Un couplage parfait de poids minimum par rapport à w est $M' = \{u_1v_2, u_2v_1, u_3v_3, u_4v_4, u_5v_5\}$. De plus, on a $w(M') = 0.4$, ce qui implique que $w(M') < 1$. La contrainte de couplages associée à M' est donc violée par x^* . Comme $F_{M'} = \{E_{u_1}^t, E_{u_2}^t, E_{u_3}^t, E_{u_4}^t\}$, la contrainte de couplages associée à M' est donnée par

$$x_{E_{u_1}^t} + x_{E_{u_2}^t} + x_{E_{u_3}^t} + x_{E_{u_4}^t} \leq 3. \quad (5.2)$$

5.1.2 Séparation des contraintes de couplages à dominance maximale

Nous proposons ici une amélioration de la séparation des contraintes de couplages (3.11), permettant d'assurer que les contraintes séparées correspondent à des couplages à dominance maximale (4.13). En effet, l'algorithme de séparation présenté dans la section précédente détermine, si elle existe, une contrainte de couplages (3.11) violée par x^* . Cependant, cette contrainte peut ne pas être à dominance maximale. Dans

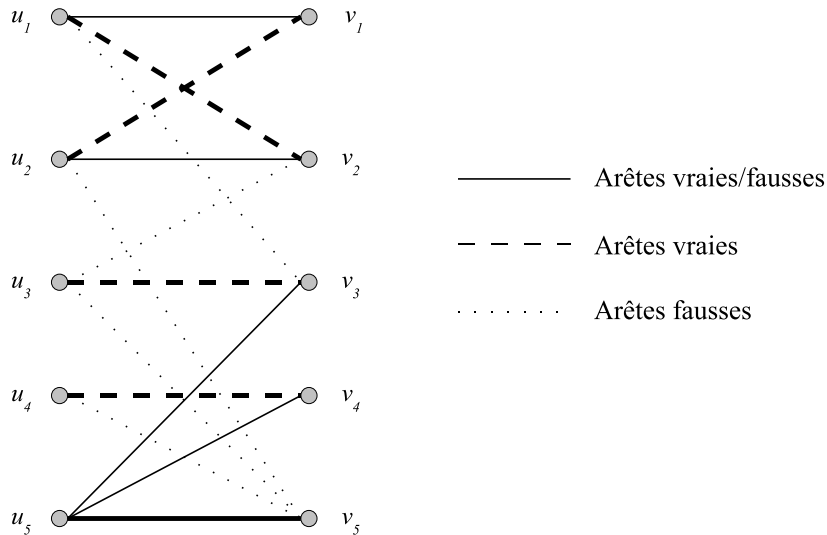


FIGURE 5.1 – Graphe G associé à x^* et couplage parfait de poids minimum.

ce cas, la proposition 4.5 implique que cette contrainte est dominée par une autre contrainte de couplages associée à un couplage parfait M' de G tel que $F_{M'} \subset F_M$. Il est donc préférable, afin de renforcer la relaxation linéaire courante, de n'ajouter que des contraintes de couplages à dominance maximale. Le problème de séparation de ces contraintes consiste alors, étant donnée une contrainte de couplages (3.11) associée à un couplage parfait M' dans G et violée par x^* , à déterminer un couplage parfait M'' de G tel que $F_{M''} \subseteq F_{M'}$ et $|F_{M''}|$ minimum.

Le problème défini précédemment correspond bien au problème de séparation des contraintes de couplages à dominance maximale (4.13). En effet, par définition des couplages à dominance maximale, le couplage parfait M'' est bien un couplage à dominance maximale. De plus, la contrainte (4.13) associée à M'' est violée par x^* car cette contrainte domine la contrainte de couplages (3.11). Par ailleurs, si aucune contrainte de couplages (3.11) n'est violée par x^* , alors aucune contrainte (4.13) n'est violée par x^* puisque les contraintes de couplages à dominance maximale sont des contraintes de couplages (3.11).

Le problème de séparation des contraintes (4.13) peut être résolu de manière exacte grâce à l'algorithme défini comme suit. Considérons le graphe biparti $H^{M'} = (U \cup V, E')$, avec $E' = (E^{tf} \cup (\cup_{F \in F_{M'}} F))$. Tout couplage parfait M de $H^{M'}$ est un couplage

parfait vérifiant $F_M \subseteq F_{M'}$. Le problème de séparation consiste alors à déterminer un couplage parfait M'' de $H^{M'}$ contenant un minimum d'arêtes intersectant $F_{M'}$, ce qui est équivalent à maximiser le nombre d'arêtes de E^{tf} . Le problème de séparation revient alors à déterminer dans $H^{M'}$ un couplage parfait de poids maximum par rapport au vecteur coût $w \in \mathbb{R}_+^{|E'|}$ défini par :

$$w_{uv} = \begin{cases} 1 & \text{si } uv \in E^{tf}, \\ 0 & \text{sinon,} \end{cases} \quad \text{pour tout } uv \in E'.$$

L'algorithme de séparation des contraintes de couplages à dominance maximale peut alors être résolu en $O(nm \log(n))$.

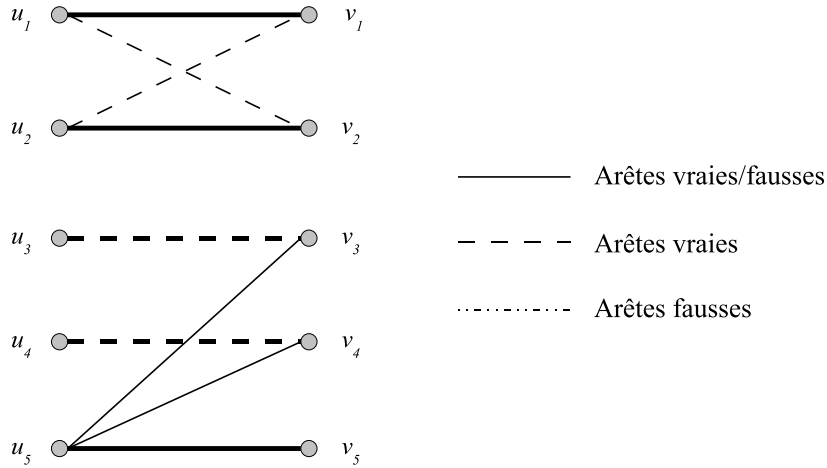
Nous donnons maintenant un exemple de séparation des contraintes de couplages à dominance maximale (4.13). Considérons de nouveau l'instance (G, U_c, π) du PSGSCP où $G = (U \cup V, E)$ est le graphe donné par la figure 5.1, $U_c = \{u_1, u_2, u_3, u_4\}$ et π se déduit des labels vrais et faux sur les arêtes indiqués sur la figure 5.1. La contrainte de couplages (3.11) violée par x^* trouvée dans l'exemple précédent est la contrainte associée au couplage parfait $M' = \{u_1v_2, u_2v_1, u_3v_3, u_4v_4, u_5v_5\}$. Le sous-graphe $H^{M'} = (U \cup V, E')$, avec $E' = (E^{tf} \cup E_{u_1}^t \cup E_{u_2}^t \cup E_{u_3}^t \cup E_{u_4}^t)$ associé au couplage parfait M' est donné par la figure 5.2. Le problème de séparation des contraintes de couplages à dominance maximale (4.13) se ramène alors à déterminer dans $H^{M'}$ un couplage parfait de poids maximum par rapport au vecteur w défini par :

$$w_{uv} = \begin{cases} 1 & \text{si } uv \in \{u_1v_1, u_2v_2, u_5v_3, u_5v_4, u_5v_5\}, \\ 0 & \text{sinon.} \end{cases}$$

Un couplage parfait de poids maximum par rapport à w est donné par $M'' = \{u_1v_1, u_2v_2, u_3v_3, u_4v_4, u_5v_5\}$. Comme la contrainte de couplages associée à M' est violée par x^* on en déduit que la contrainte de couplages associée à M'' est aussi violée par x^* . La contrainte de couplages associée à M'' est donnée par

$$x_{E_{u_3}^t} + x_{E_{u_4}^t} \leq 1. \quad (5.3)$$

De plus, il est clair que l'inégalité (5.3) domine l'inégalité (5.2) car l'inégalité (5.2) est obtenue par combinaison linéaire de l'inégalité (5.3) et des inégalités triviales $x_{E_{u_1}^t} \leq 1$ et $x_{E_{u_2}^t} \leq 1$.

FIGURE 5.2 – Graphe H^M et couplage parfait à dominance maximale.

5.1.3 Séparation des contraintes de couplages proches

On rappelle que si F est un élément de \mathcal{E} , alors \bar{F} correspond à l'élément de $\mathcal{E} \setminus \{F\}$ couvrant le même sommet de U_c que l'ensemble F . Autrement dit, si $F = E_u^t$ (resp. E_u^f) pour un $u \in U_c$ donné, alors $\bar{F} = E_u^f$ (resp. $\bar{F} = E_u^t$). Nous donnons dans cette section une séparation heuristique des contraintes de couplages proches (4.19). Cette heuristique s'appuie sur la séparation des contraintes de couplages à dominance maximale définie dans la section précédente. Notons par M' le couplage à dominance maximale associé à la contrainte de couplages à dominance maximale (4.13) trouvée par l'algorithme de séparation précédent. L'heuristique cherche alors un couplage parfait M et un élément $F \in F_{M'}$ tel que M est proche de M' par rapport à F et la contrainte de couplages proches associée à M' et F est violée par x^* .

Cet algorithme n'est donc pas exact car il peut exister un couplage parfait M_1 différent de M' dans G et un élément F_1^* appartenant à F_{M_1} tels qu'il existe un couplage parfait proche de M_1 par rapport à F_1^* et la contrainte (4.19) associée à M_1 et F_1^* soit violée par x^* . Cette contrainte ne sera pas détectée par l'algorithme. La complexité du problème de séparation exact des contraintes de couplages proches est une question ouverte.

L'heuristique, décrite par l'algorithme 1, peut être présentée de la manière suivante. Soit M' un couplage à dominance maximale tel que la contrainte de couplages à dominance maximale (4.13) associée à M' est violée par x^* . Pour chaque élément $F \in F_{M'}$, on considère le sous-graphe $H_F = (U \cup V, E')$ de G , avec $E' = (E^{tf} \cup \overline{F} \cup (\cup_{F' \in F_M \setminus \{F\}} F'))$. Supposons qu'il existe un couplage parfait, disons M'' , dans H_F . Alors $F_{M''} \subseteq (F_{M'} \cup \{\overline{F}\} \setminus \{F\})$. Le couplage parfait M'' est donc proche de M' par rapport à F . Comme M' est un couplage à dominance maximale, alors la contrainte de couplages proches (4.19) associée à M' et F est valide pour le PSGSCP. Comme elle domine la contrainte (4.13) associée à M' , elle est violée par x^* .

Notre séparation des inégalités de couplages proches utilise l'algorithme des chaînes augmentantes. La recherche d'un couplage parfait dans H_F peut être fait en $O(m)$. En effet, la restriction de M' sur H_F correspond à un couplage de cardinalité $n - 1$. S'il existe une M' -chaîne augmentante dans H_F , alors le nouveau couplage sera un couplage parfait de H_F . La recherche d'un couplage parfait dans H_F se fait donc en déterminant une M' -chaîne augmentante. La recherche d'une chaîne augmentante dans un graphe se fait en $O(m)$. Nous obtenons donc une heuristique s'exécutant en $O(|F_{M'}|m)$.

Algorithme 1: Heuristique de séparation des contraintes (4.19)

Entrées : - Vecteur x^*

- Un couplage parfait à dominance maximale M' de G tel que la contrainte de couplages associée à M' est violée par x^*

Sorties : - Ensemble S de contraintes de couplages proches (4.19) violées par x^*

début

$S = \emptyset;$

$E_{M'} = \cup_{F \in F_{M'}} F \cup E^{tf};$

pour chaque $F \in F_{M'}$ **faire**

 Construire le graphe $H_F = (U \cup V, E')$, avec $E' = (E_{M'} \cup \overline{F}) \setminus F;$

si il existe une chaîne $(M' \setminus F)$ -augmentante dans H_F **alors**

 Ajouter dans S la contrainte de couplages proches (4.19) associée à M' et $F;$

fin

fin

retourner $S;$

fin

Nous donnons maintenant un exemple de séparation des contraintes de couplages proches (4.19). Considérons de nouveau l'instance (G, U_c, π) du PSGSCP où $G = (U \cup V, E)$ est le graphe donné par la figure 5.1, $U_c = \{u_1, u_2, u_3, u_4\}$ et π se déduit

des labels vrais et faux sur les arêtes indiqués par la figure 5.1. Le couplage parfait à dominance maximale associée à la contrainte violée par x^* trouvée par le problème de séparation des contraintes de couplages à dominance maximale est donné par le couplage parfait $M = \{u_1v_1, u_2v_2, u_3v_3, u_4v_4, u_5v_5\}$. Les sous-graphes $H_{E_{u_3}^t} = (U \cup V, E')$, avec $E' = (E^{tf} \cup E_{u_3}^f \cup E_{u_4}^t)$ et $H_{E_{u_4}^t} = (U \cup V, E'')$, avec $E'' = (E^{tf} \cup E_{u_3}^t \cup E_{u_4}^f)$ sont donnés par les figures 5.3(a) et 5.3(b). Le problème de séparation des contraintes de couplages se ramène à déterminer dans $H_{E_{u_3}^t}$ et dans $H_{E_{u_4}^t}$ s'il existe un couplage parfait. Comme les sous-graphes $H_{E_{u_3}^t}$ et $H_{E_{u_4}^t}$ possèdent un couplage parfait, l'heuristique génère les contraintes de couplages proches

$$x_{E_{u_3}^t} + x_{E_{u_3}^f} + x_{E_{u_4}^t} \leq 1 \tag{5.4}$$

$$x_{E_{u_3}^t} + x_{E_{u_4}^f} + x_{E_{u_4}^t} \leq 1 \tag{5.5}$$

violée par x^* .

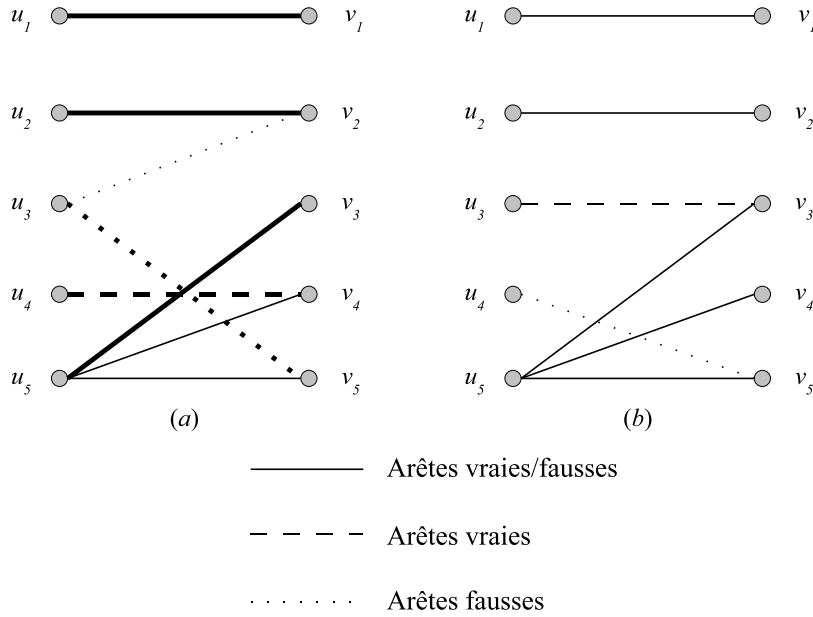


FIGURE 5.3 – Graphes $H_{E_{u_3}^t}$ et $H_{E_{u_4}^t}$.

5.1.4 Séparation des contraintes de 2-couplages

Dans cette section, nous présentons des algorithmes de séparation des contraintes de k -couplages (4.32) lorsque $k = 2$. Nous présentons d'abord un algorithme de séparation

exact correspondant à la résolution d'un programme linéaire en nombres entiers. La complexité de cet algorithme est une question ouverte. Pour cette raison nous considérons ensuite une heuristique permettant de résoudre une version relaxée de ce problème de séparation. Nous montrons que cette heuristique correspond à un algorithme de flot de coût minimum qui peut donc être résolu en temps polynomial.

Rappelons que l'on note par Θ l'ensemble des sous-ensembles de \mathcal{E} différents de l'ensemble vide vérifiant la condition (i) de la définition (3.9). Autrement dit, un sous-ensemble non-vide $\theta \subseteq \mathcal{E}$ appartient à Θ s'il n'existe pas un sommet $u \in U_c$ tel que $\{E_u^t, E_u^f\} \subseteq \theta$.

Soient $\theta \in \Theta$ et $\theta' \subseteq \theta$. La contrainte de 2-couplages associée à θ et θ' s'écrit :

$$\sum_{F \in \theta'} x_F + 2 \sum_{F \in \theta \setminus \theta'} x_F \leq |\theta'| + 2(|\theta \setminus \theta'| - 1). \quad (5.6)$$

On remarque que dans le chapitre 4, on a appelé contrainte de 2-couplages les contraintes de k -couplages lorsque $k = 2$ et $\theta' = \theta$. Ici on appelle contrainte de 2-couplages les contraintes associées à $\theta \in \Theta$, $\theta' \subseteq \theta$ et $k = 2$.

5.1.4.1 Séparation exacte des contraintes de 2-couplages

Les contraintes de 2-couplages sont en nombre exponentiel. Afin de pouvoir les utiliser dans un algorithme de coupes et branchements, il est nécessaire de s'intéresser au problème de séparation de ces contraintes. Ce problème peut être défini comme suit. Étant donné un vecteur x^* , existe-t-il $\theta \in \Theta$, $\theta' \subseteq \theta$ tels que le sous-graphe H_θ de G est 2-couplages par rapport à θ' et la contrainte de 2-couplages (5.6) associée à θ et θ' est violée par x^* ?

Posons $x' = 1 - x^*$. Les contraintes de 2-couplages (5.6) pour x^* sont donc équivalentes à

$$\sum_{F \in \theta'} x'_F + 2 \sum_{F \in \theta \setminus \theta'} x'_F \geq 2. \quad (5.7)$$

Notons par $D(G, \theta, \theta')$ l'ensemble des couples (θ, θ') tels que $\theta \in \Theta$, $\theta' \subseteq \theta$ et le graphe H_θ est 2-couplages par rapport à θ' . Étant donné $(\theta, \theta') \in D(G, \theta, \theta')$, on note par $w(\theta, \theta')$ la valeur du membre de gauche de la contrainte (5.7) associée à θ et θ' donnée pour x^* . On a alors

$$w(\theta, \theta') = \sum_{F \in \theta'} x'_F + 2 \sum_{F \in \theta \setminus \theta'} x'_F. \quad (5.8)$$

Le problème de séparation des contraintes de 2-couplages (5.6) associé à x^* revient alors à déterminer

$$w^* = w(\theta^*, \theta'^*) = \min_{(\theta, \theta') \in D(G, \theta, \theta')} w(\theta, \theta') \quad (5.9)$$

Si $w^* < 2$, alors la contrainte de 2-couplages associée à θ^* et θ'^* est violée par x^* . Autrement, il n'existe pas de contrainte de 2-couplages (5.6) violée par x^* .

Nous allons montrer dans un premier temps que le problème de séparation est équivalent à un problème secondaire. Nous montrons ensuite que ce problème secondaire est équivalent à un problème de flot particulier.

Soient M_1, M_2, \dots, M_n n couplages parfaits de G . Notons par \mathcal{M} la famille d'arêtes correspondant à l'union des couplages parfaits M_i , $i = 1, 2, \dots, n$ en prenant en compte les multiplicités des arêtes. Autrement dit, si une arête e appartient à k couplages parfaits, alors elle apparaît k fois dans \mathcal{M} . On a alors $|\mathcal{M}| = n|U|$. La famille \mathcal{M} est appelée *n -couplages parfaits vrai/faux (n -CPVF)* si elle vérifie :

$$E_u^t \cap \mathcal{M} = \emptyset \text{ ou } E_u^f \cap \mathcal{M} = \emptyset \text{ pour tout } u \in U_c. \quad (5.10)$$

Étant donné un n -CPVF \mathcal{M} , on associe un coût, disons $C(\mathcal{M})$, égal à $\sum_{u \in U_c} c_u$ où

$$c_u = \begin{cases} 2x'_{E_u^t} & \text{si } |E_u^t \cap \mathcal{M}| = n, \\ x'_{E_u^t} & \text{si } 1 \leq |E_u^t \cap \mathcal{M}| < n, \\ 2x'_{E_u^f} & \text{si } |E_u^f \cap \mathcal{M}| = n, \\ x'_{E_u^f} & \text{si } 1 \leq |E_u^f \cap \mathcal{M}| < n, \\ 0 & \text{sinon.} \end{cases} \quad (5.11)$$

On montre avec les deux propositions suivantes que le problème de séparation des contraintes de 2-couplages se ramène à chercher un n -CPVF de poids minimum.

Lemme 5.2 *Soit \mathcal{M} un n -CPVF. Alors il existe $(\theta, \theta') \in D(G, \theta, \theta')$ tel que $w(\theta, \theta') = C(\mathcal{M})$.*

Preuve. Nous construisons θ, θ' à partir de \mathcal{M} de la manière suivante. Considérons un sommet $u \in U$. Comme \mathcal{M} est un n -CPVF, alors on a $|\delta(u) \cap (\mathcal{M} \setminus E_u^t)| = n$ et/ou $|\delta(u) \cap (\mathcal{M} \setminus E_u^f)| = n$. Supposons, sans perte de généralité, que $|\delta(u) \cap (\mathcal{M} \setminus E_u^f)| = n$. Si $|E_u^t \cap \mathcal{M}| = 0$, E_u^t n'appartient pas à θ . Si $|E_u^t \cap \mathcal{M}| = n$, alors on ajoute E_u^t dans θ . Autrement, on ajoute E_u^t dans θ' et θ . Il est facile de voir que par construction, on a $w(\theta, \theta') = C(\mathcal{M})$.

Montrons maintenant que le sous-graphe H_θ de G est 2-couplages par rapport à θ' . Considérons un élément F de θ' . Par construction, comme $F \in \theta'$, cela implique que $1 \leq |F \cap \mathcal{M}| < n$. Comme \mathcal{M} est l'union de n -couplages parfaits, il existe au moins un couplage parfait de H_θ , disons M_1 , tel que $\delta(u) \cap E^{tf} \cap M_1 \neq \emptyset$. Il s'ensuit que M_1 est un couplage parfait de $H_{\theta \setminus \{F\}}$. Comme F est un élément quelconque de θ' , on déduit alors que H_θ contient un couplage parfait même après la suppression d'un ensemble de θ' , ce qui implique que H_θ est 2-couplage. \square

Lemme 5.3 *Soit $(\theta, \theta') \in D(G, \theta, \theta')$. Alors il existe un n -CPVF \mathcal{M} tel que $C(\mathcal{M}) \leq w(\theta, \theta')$.*

Preuve. On construit, à partir de θ et θ' , n couplages parfaits dont l'union forme un n -CPVF. On a $\theta' = \{F_1, F_2, \dots, F_{|\theta'|}\}$. Pour tout $F \in \theta'$, il existe dans $H_\theta \setminus F$ un couplage parfait puisque H_θ est 2-couplages par rapport à θ' . Notons par $M_1, M_2, \dots, M_{|\theta'|}$ les couplages parfaits obtenus dans $H_\theta \setminus F_i$, $i = 1, 2, \dots, |\theta'|$. Considérons $n - |\theta'|$ copies additionnelles de M_1 et considérons le n -couplages parfaits, disons \mathcal{M} , obtenu par l'union (en prenant en compte les multiplicités des arêtes) des n couplages parfaits ainsi obtenus. Supposons maintenant que \mathcal{M} ne vérifie pas (5.10), c'est-à-dire qu'il existe $u \in U_c$ tel que $\mathcal{M} \cap E_u^t \neq \emptyset$ et $\mathcal{M} \cap E_u^f \neq \emptyset$. Comme \mathcal{M} est l'union de couplage parfait de H_θ , cela implique que θ contient E_u^t et E_u^f , une contradiction avec le fait que $\theta \in \Theta$. \mathcal{M} est donc un n -CPVF. Considérons maintenant le coût associé à \mathcal{M} . Soit u un sommet quelconque de U . Si $\{E_u^t, E_u^f\} \cap \theta = \emptyset$, alors $\mathcal{M} \cap \{E_u^t, E_u^f\} = \emptyset$, ce qui implique que $c_u = 0$. Supposons maintenant que $\{E_u^t, E_u^f\} \cap \theta \neq \emptyset$. Comme $\theta \in \Theta$, on a $|\{E_u^t, E_u^f\} \cap \theta| = 1$. Notons par F l'élément de $\{E_u^t, E_u^f\}$ appartenant à θ . Si $F \in \theta'$, alors il existe $i \in \{1, \dots, |\theta'|\}$ tel que M_i est un couplage parfait de $H_{\theta \setminus \{F_i\}}$. On déduit alors que $|\mathcal{M} \cap F| < n$, ce qui implique que $c_u \leq x_F^*$. Si $F \notin \theta'$, alors $|\mathcal{M} \cap F| \leq n$, ce qui implique que $c_u \leq 2x_F^*$. Il est alors facile de voir que $c(\mathcal{M}) \leq w(\theta, \theta')$. \square

Théorème 5.4 *Le problème de séparation des contraintes de 2-couplages (5.6) associé à x^* revient à chercher un n -CPVF de coût minimum.*

Preuve. Soit \mathcal{M} un n -CPVF de coût minimum. Si $c(\mathcal{M}) < 2$, alors par le lemme 5.2, il existe $(\theta, \theta') \in D(G, \theta, \theta')$ tel que $w(\theta, \theta') = c(\mathcal{M}) < 2$. La contrainte de 2-couplages associée à θ et θ' est alors violée par x^* .

Supposons maintenant que $c(\mathcal{M}) \geq 2$. Par le lemme 5.3, on déduit que pour tout $(\theta, \theta') \in D(G, \theta, \theta')$, on a $w(\theta, \theta') \geq 2$, ce qui implique qu'aucune contrainte de 2-couplages (5.6) n'est violée par x^* . \square

Nous formulons maintenant la recherche d'un n -CPVF de coût minimum à l'aide d'un programme linéaire en nombres entiers. Ce programme s'appuie sur un graphe orienté $D = (V_D, A_D)$ contenant des arcs multiples et construit à partir de $G = (U \cup V, E)$ de la manière suivante.

Pour chaque sommet $u \in U_c$, on ajoute dans V_D les sommets $\{w_u^1, \dots, w_u^9\}$. On ajoute également dans V_D les sommets $u \in U \setminus U_c$, les sommets $v \in V$ et deux sommets s et t . L'ensemble V_D contient alors $8|U_c| + |U| + |V| + 2$ sommets. Notons $n = |U_c|$. L'ensemble A_D est construit de la manière suivante. Pour chaque arc de $a \in A_D$ on associe également un coût $p(a)$ et une capacité $c(a)$. Pour tout $u \in U_c$, on ajoute dans A_D les arcs

- 1) $(w_u^2, w_u^1), (w_u^6, w_u^1), (w_u^7, w_u^6), (w_u^3, w_u^2)$, avec un coût de 0 et une capacité de n ,
- 2) (w_u^5, w_u^2) , avec un coût de $x'_{E_u^t}$ et une capacité de n ,
- 3) (w_u^9, w_u^6) , avec un coût de $x'_{E_u^f}$ et une capacité de n ,
- 4) (w_u^4, w_u^5) , avec un coût de $x'_{E_u^t}$ et une capacité de 1,
- 5) (w_u^4, w_u^5) , avec un coût de 0 et une capacité de $n - 1$,
- 6) (w_u^8, w_u^9) , avec un coût de $x'_{E_u^f}$ et une capacité de 1,
- 7) (w_u^8, w_u^9) , avec un coût de 0 et une capacité de $n - 1$,

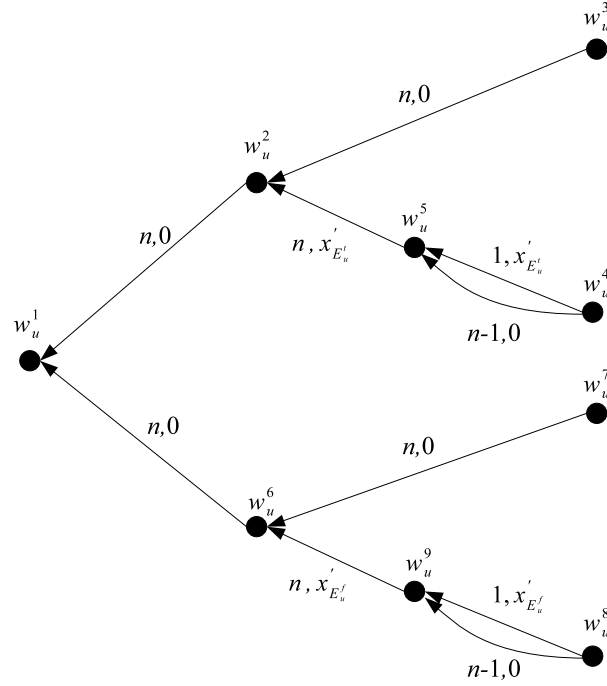
La figure 5.4 donne l'ensemble d'arêtes ajoutées pour un sommet u donné. Les deux nombres sur chaque arc a correspondent respectivement à $c(a)$ et $p(a)$.

Pour tout $E_u^t \in \mathcal{E}$ (resp. $E_u^f \in \mathcal{E}$) et pour tout $uv \in E_u^t$ (resp. $uv \in E_u^f$) on ajoute l'arc (v, w_u^4) (resp. (v, w_u^8)), avec un coût de 0 et une capacité de n . Pour tout $u \in U_c$ et pour tout $uv \in E^{tf}$ on ajoute les arcs (v, w_u^3) et (v, w_u^7) , avec un coût de 0 et une capacité de n . Pour tout $u \in U \setminus U_c$ et pour tout $uv \in E^{tf}$ on ajoute l'arc (v, u) , avec un coût de 0 et une capacité de n . Finalement, pour tout $v \in V$, $u \in U \setminus U_c$ et $u' \in U_c$, on ajoute les arcs (s, v) , (u, t) et $(w_{u'}^1, t)$, avec un coût de 0 et une capacité de n .

Nous allons voir que la recherche d'un n -CPVF de coût minimum dans G revient à trouver un st -flot particulier de valeur $|U||U_c|$ et de coût minimum.

Considérons un flot f tel que

- (i) $f(a) \leq c(a)$ pour tout $a \in A_D$,
- (ii) pour tout $u \in V_D \setminus \{s, t\}$, $\sum_{(u,v) \in A_D} f((u, v)) = \sum_{(v,u) \in A_D} f((v, u))$, c'est-à-dire que le flot est conservé,

FIGURE 5.4 – Gadget de remplacement d'un sommet $u \in U_c$.

$$(iii) \sum_{(s,v) \in A_D} f((s,v)) = \sum_{(v,t) \in A_D} f((v,t)) = |U||U_c|,$$

$$(iv) \text{ pour tout } u \in U_c, f((w_u^2, w_u^1)) = 0 \text{ ou } f((w_u^6, w_u^1)) = 0,$$

$$(v) \text{ minimiser } \sum_{a \in A'} p(a) \text{ tel que } A' = \{a \in A : f(a) > 0\}.$$

Ce problème de flot minimise la somme des coûts des arcs traversés par le flot et pas le coût du flot.

Proposition 5.5 *Le problème de séparation des contraintes de 2-couplages (5.6) associé à x^* revient à trouver un flot f vérifiant les conditions (i)-(v).*

Preuve. Considérons un n -CPVF, disons \mathcal{M} , de poids minimum dans G . Nous allons définir dans D un flot f correspondant à \mathcal{M} .

- $f((s, v)) = n$, pour tout $v \in V$, $f((u, t)) = n$, pour tout $u \in U \setminus U_c$, et $f((w_u^1, t)) = n$, pour tout $u \in U_c$.
- $f((v, u)) = k$ où $u \in U \setminus U_c$, $v \in V$ et k est égal au nombre d'occurrences de l'arête uv dans \mathcal{M} .
- $f((v, w_u^4)) = k$ où $u \in U_c$, $v \in V$ tel que $uv \in E_u^t$ et k est égal au nombre d'occurrences de l'arête uv dans \mathcal{M} .
- $f((v, w_u^8)) = k$ où $u \in U_c$, $v \in V$ tel que $uv \in E_u^f$ et k est égal au nombre d'occurrences de l'arête uv dans \mathcal{M} .
- $f((v, w_u^3)) = k$ (resp. $f((v, w_u^7)) = k$), avec $u \in U_c$ et $v \in V$ et k égal au nombre d'occurrences de l'arête uv dans \mathcal{M} , tel que $uv \in E_u^{tf}$.

Pour tout $u \in U_c$:

- si $|E_u^t \cap \mathcal{M}| \leq n$ (resp. $|E_u^f \cap \mathcal{M}| \leq n$), alors $f((w_u^4, w_u^5)) = f((w_u^5, w_u^2)) = |E_u^t \cap \mathcal{M}|$ (resp. $f((w_u^8, w_u^9)) = f((w_u^9, w_u^6)) = |E_u^f \cap \mathcal{M}|$), $f((w_u^3, w_u^2)) = n - |E_u^t \cap \mathcal{M}|$ (resp. $f((w_u^7, w_u^6)) = n - |E_u^f \cap \mathcal{M}|$), $f((w_u^2, w_u^1)) = n$ (resp. $f((w_u^6, w_u^1)) = n$).
- si $|E_u^{tf} \cap \mathcal{M}| = n$, alors $f((w_u^3, w_u^2)) = n$, $f((w_u^2, w_u^1)) = n$ choisis arbitrairement.

Comme le n -CPVF vérifie la condition (5.10), on en déduit que le flot vérifie donc la condition (iv) et, par construction, le flot vérifie les conditions (i)-(v) est bien un flot de coût minimum $C(\mathcal{M})$.

Considérons maintenant dans D un st-flot de valeur $|U||U_c|$ et de coût minimum vérifiant les conditions (i)-(v). Considérons l'ensemble d'arêtes \mathcal{M} de G où pour tout $u \in U$ et pour tout $v \in V$, on ajoute $f((v, w_u^4)) + f((v, w_u^8)) + f((v, w_u^3)) + f((v, w_u^7)) + f((v, u))$ fois l'arête $uv \in \mathcal{M}$. Il est clair que chaque sommet de $U \cup V$ possède exactement n arêtes incidentes. Cela implique que \mathcal{M} correspond à un n -CPVF. De plus comme le flot vérifie la condition (iv), on en déduit que l'ensemble \mathcal{M} vérifie la condition (5.10). On en déduit que \mathcal{M} est un n -CPVF. De plus, il est clair que le poids de \mathcal{M} est égal au coût du flot. \square

Pour résoudre ce problème de flot nous proposons, dans la suite, une formulation à l'aide d'un programme linéaire en nombres entiers. Soient $f \in \{0, 1, \dots, n\}^{|A_D|}$ tel que, pour tout $(u, v) \in A_D$, f_{uv} correspond au flot de u à v , et $y \in \{0, 1\}^{|A_D|}$ tel que, pour tout $(u, v) \in A_D$, $y_{uv} = 1$ si $f((u, v)) > 0$ et $y_{uv} = 0$ sinon.

Soit $z \in \{0, 1\}^{|U_c|}$ tel que,

$$z_u = \begin{cases} 1 & \text{si } f((w_u^6, w_u^1)) = 0, \\ 0 & \text{si } f((w_u^2, w_u^1)) = 0, \end{cases} \text{ pour tout } u \in U_c.$$

Le problème de séparation des contraintes de 2-couplages est équivalent au programme linéaire en nombres entiers suivant :

$$\begin{aligned} \min \quad & \sum_{(u,v) \in A_D} p((u,v)) y_{uv} \\ & \sum_{(v,u) \in \delta^{in}(u)} f_{vu} - \sum_{(u,v) \in \delta^{out}(u)} f_{uv} = b_v \quad \text{pour tout } u \in V_D, \end{aligned} \quad (5.12)$$

$$f_{w_u^2 w_u^1} \leq n z_u \quad \text{pour tout } u \in U_c, \quad (5.13)$$

$$f_{w_u^6 w_u^1} \leq n(1 - z_u) \quad \text{pour tout } u \in U_c, \quad (5.14)$$

$$f_{uv} \leq c((u,v)) y_{uv} \quad \text{pour tout } (u,v) \in A_D, \quad (5.15)$$

$$z_u \in \{0, 1\} \quad \text{pour tout } u \in U_c, \quad (5.16)$$

$$f_{uv} \in \{0, \dots, n\} \quad \text{pour tout } (u,v) \in A_D, \quad (5.17)$$

$$y_{uv} \in \{0, 1\} \quad \text{pour tout } (u,v) \in A_D, \quad (5.18)$$

où

$$b_v = \begin{cases} |U||U_c| & \text{si } v = s, \\ -|U||U_c| & \text{si } v = t, \\ 0 & \text{sinon,} \end{cases}$$

et $c((u,v))$ correspond à la capacité de l'arc (u,v) .

Les contraintes (5.12) sont les contraintes de conservation de flots assurant que nous avons bien un flot de $|U||U_c|$ allant de s à t . Les contraintes (5.13) et (5.14) représentent la condition (iv). Les contraintes (5.15) permettent de borner la valeur du flot passant sur chaque arc en fonction de sa capacité. Les contraintes (5.16) et (5.17) sont les contraintes d'intégrités.

Comme le calcul de flot de coût minimum respectant la condition (v) est un problème NP-difficile[32], nous conjecturons que ce problème est lui aussi NP-difficile. Nous présentons, dans la prochaine section, une heuristique de séparation des contraintes de 2-couplages.

5.1.4.2 Séparation heuristique des contraintes de 2-couplages

Nous présentons une heuristique de séparation des contraintes de 2-couplages. Cette heuristique consiste d'abord à déterminer un ensemble $\omega \in \Theta$ tel que $|\{E_u^t, E_u^f\} \cap \omega| = 1$

pour tout $u \in U_c$. L'heuristique détermine ensuite s'il existe $\theta \subseteq \omega$ et $\theta' \subseteq \theta$ tel que la contrainte de 2-couplages (5.6) associée à θ et θ' est violée par x^* . Cette séparation est donc clairement une heuristique puisque pour être exacte, il faudrait appliquer la séparation pour tout ensemble de Θ vérifiant $|\{E_u^t, E_u^f\} \cap \omega| = 1$ pour tout $u \in U_c$.

L'ensemble ω est déterminé de la manière suivante. Pour tout sommet $u \in U_c$, on ajoute E_u^t dans ω si $x_{E_u^t}^* \geq 0.5$, E_u^f sinon.

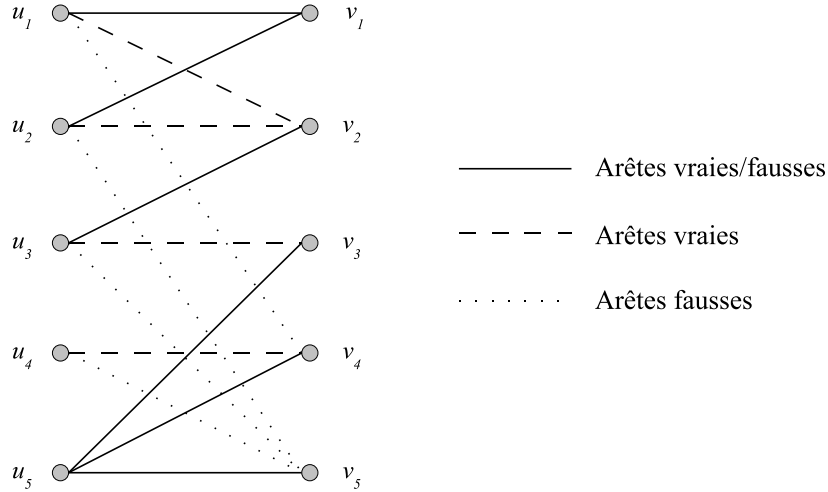
Séparer les contraintes de 2-couplages par rapport à ω revient à déterminer un n -CPVF \mathcal{M} de G de coût minimum tel que $\mathcal{M} \cap F = \emptyset$ pour tout $F \in \mathcal{E} \setminus \omega$. En termes de flot, cela revient à déterminer dans D un flot respectant les conditions (i)-(v) tel que $f((w_u^6, w_u^1)) = 0$ si $E_u^t \in \omega$ et $f((w_u^2, w_u^1)) = 0$ si $E_u^f \in \omega$ pour tout $u \in U_c$. Nous calculons un flot f de valeur $|U|/|U_c|$ de coût minimum dans le graphe $D(W)$, où W correspond à l'ensemble V_D dans lequel les sommets $\{w_u^2, w_u^3, w_u^4, w_u^5\}$ pour tout $u \in U_c$ tel que $E_u^f \in \omega$ et les sommets $\{w_u^6, w_u^7, w_u^8, w_u^9\}$ pour tout $u \in U_c$ tel que $E_u^t \in \omega$ sont supprimés. Nous proposons de relâcher la condition (v). Soient $z' = \sum_{a \in A'} p(a)$ tel que $A' = \{a \in A : f(a) > 0\}$. Notons que $z' = w(\theta, \theta')$ où θ (resp. θ') correspond à l'ensemble des ensembles $F_u \in \omega$, $u \in U_c$, tel que $f((w_u^5, w_u^2))$ ou $f((w_u^9, w_u^6))$ est égal à $|U_c|$ (resp. est compris entre 1 et $|U_c| - 1$). Si $z' < 2$ alors la contrainte de 2-couplages associée à θ et θ' est violée par x^* . Comme ce problème de flot peut se résoudre en temps polynomial, notre heuristique peut elle aussi se résoudre en temps polynomial.

Nous donnons maintenant un exemple de séparation heuristique des contraintes de 2-couplages (5.6). Considérons l'instance (G, U_c, π) du PSGSCP où $G = (U \cup V, E)$ est le graphe donné par la figure 5.5, $U_c = \{u_1, u_2, u_3, u_4\}$ et π se déduit des labels vrais et faux sur les arêtes indiqués sur la figure 5.1. Soit la solution courante à séparer $x_{E_{u_1}^t}^* = 1$, $x_{E_{u_2}^t}^* = 1$, $x_{E_{u_3}^t}^* = 0.6$, $x_{E_{u_4}^t}^* = 1$, $x_{E_{u_1}^f}^* = 0$, $x_{E_{u_2}^f}^* = 0$, $x_{E_{u_3}^f}^* = 0.4$, $x_{E_{u_4}^f}^* = 0$, les valeurs x' associées sont $x'_{E_{u_1}^t} = 0$, $x'_{E_{u_2}^t} = 0$, $x'_{E_{u_3}^t} = 0.4$, $x'_{E_{u_4}^t} = 0$, $x'_{E_{u_1}^f} = 1$, $x'_{E_{u_2}^f} = 1$, $x'_{E_{u_3}^f} = 0.6$, $x'_{E_{u_4}^f} = 1$. L'ensemble ω correspond alors à $\{E_u^t : u \in U_c\}$. L'ensemble W correspond alors à $V_D \setminus \{\{w_u^6, w_u^7, w_u^8, w_u^9\} : u \in U_c\}$. Le graphe $D(W)$ donné dans la figure 5.6. Le flot passe par les arêtes en gras. Nous avons $z' = 0.8$, on en déduit $\theta = \{E_{u_1}^t, E_{u_2}^t, E_{u_3}^t, E_{u_4}^t\}$ et $\theta' = \{E_{u_1}^t, E_{u_2}^t\}$

La contrainte de 2-couplages

$$x_{E_{u_1}^t} + x_{E_{u_2}^t} + 2x_{E_{u_3}^t} + 2x_{E_{u_4}^t} \leq 4$$

est donc valide et violée par x^* . Nous avons présenté des algorithmes de séparation associé aux contraintes de couplages, couplages à dominance maximale, couplages proches

FIGURE 5.5 – Graphe G .

et 2-couplages. Nous allons voir dans la prochaine section comment nous avons intégré ces algorithmes de séparation dans notre algorithme de coupes et branchements.

5.2 Algorithme de coupes et branchements

Dans cette section, nous présentons un algorithme de coupes et branchements pour le problème du sous-graphe sans couplage parfait. La formulation (P) contient une famille de contraintes en nombre exponentiel, à savoir les contraintes de couplages (3.11), les autres apparaissant en nombre polynomial. Cette famille d'inégalités sera donc ajoutée dynamiquement au programme linéaire courant. Notre algorithme commence par résoudre le programme linéaire formé par les contraintes de (P) apparaissant en nombre polynomial, c'est-à-dire, le *programme linéaire initial* donné par

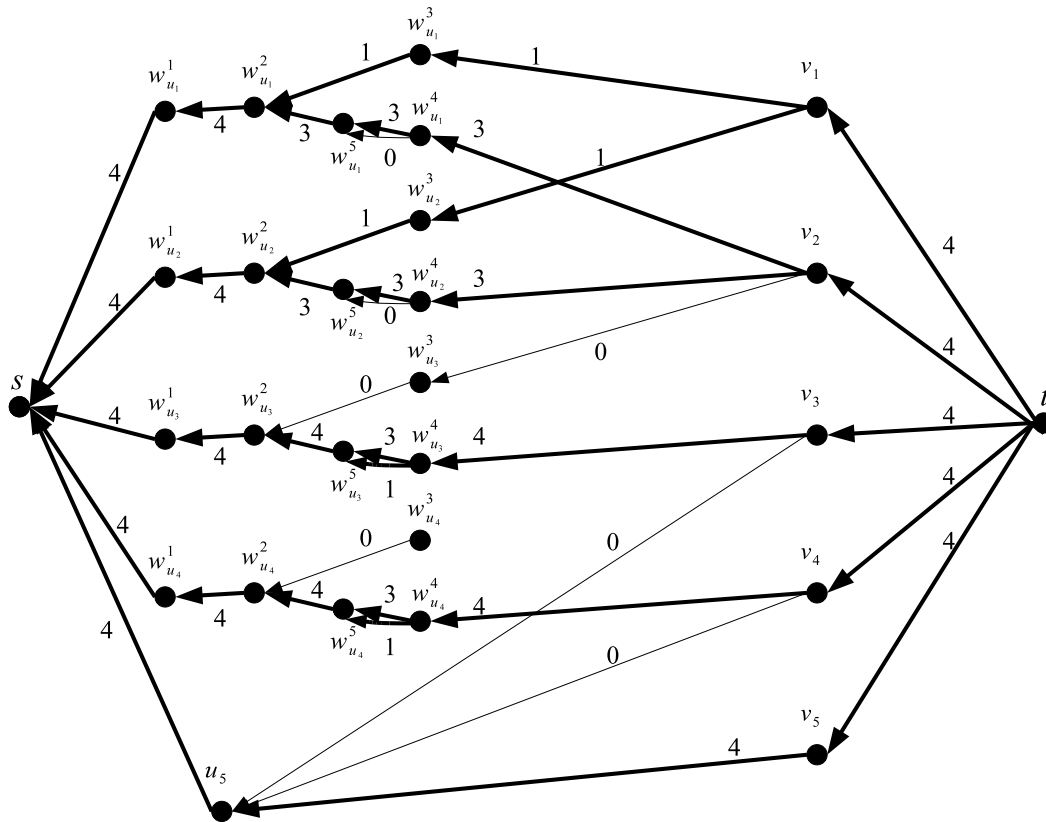


FIGURE 5.6 – Flot de coût minimum.

$$\begin{aligned}
 & \max \sum_{F \in \mathcal{E}} x_F \\
 & x_{E_u^t} + x_{E_u^f} \leq 1 && \forall u \in U_c, \\
 & x_F \geq 0 && \forall F \in \mathcal{E}, \\
 & x_F \leq 1 && \forall F \in \mathcal{E}.
 \end{aligned}$$

La solution optimale $x^* \in \mathbb{R}^{|\mathcal{E}|}$ de cette relaxation est réalisable pour le problème si x^* est un vecteur entier qui satisfait les contraintes de couplages (3.11). En général, la solution x^* n'est pas réalisable. Dans ce cas, à chaque itération, l'algorithme génère des inégalités valides pour le problème mais pas pour la solution optimale courante. Les inégalités ainsi trouvées sont ajoutées au programme linéaire courant et le nouveau programme est résolu. Ce processus est repris jusqu'à ce qu'aucune inégalité violée n'est détectée. Si la solution n'est pas encore réalisable, l'algorithme procède alors

à un branchement. L'algorithme 2 ci-dessous illustre les principales phases de notre algorithme de coupes et branchements.

Algorithme 2: Algorithme de coupes et branchements

Entrées : un graphe biparti $G = (U \cup V, E)$, un sous-ensemble $U_c \in U$ et une partition π de E

Sorties : Solution optimale de (P)

début

1 : PL \leftarrow Programme linéaire initial

2 : Résoudre le programme linéaire PL.

Soit x la solution optimale de PL.

3 : **si** x est réalisable pour $P(G, U_c, \pi)$ **alors**

| x est une solution optimale. STOP

fin

4 : **si** des contraintes violées (couplages à dominance maximale, couplages proches, 2-couplages) par x sont trouvées **alors**

| Les ajouter à PL.

| Aller en 2.

fin

5 : **sinon**

| Brancher sur une variable fractionnaire.

fin

6 : Prendre la meilleure solution de tous les sous-problèmes.

fin

La séparation des inégalités est réalisée dans l'ordre suivant. Dans un premier temps, on sépare les contraintes de couplages. Ensuite, on améliore, si c'est possible, la contrainte de couplages en contrainte de couplages à dominance maximale. Puis on génère l'ensemble des contraintes de couplages proches associées à ce couplage. Enfin on sépare les contraintes de 2-couplages. Les algorithmes de séparation utilisés pour ces inégalités ont été donnés dans la section précédente. Les contraintes de 2-couplages sont séparées uniquement en utilisant l'heuristique définie dans la section 5.1.4.2.

On peut remarquer que toutes les inégalités sont globales (*i.e.* valides dans tout l'arbre de branchements). Par ailleurs, plusieurs contraintes peuvent être ajoutées à chaque itération.

Lorsque des contraintes (3.11), (4.13), (4.19) et (5.6) du programme linéaire ne sont plus serrées par la solution x pendant plusieurs itérations, ces dernières sont supprimées du programme linéaire afin de ne pas alourdir inutilement la taille de ce dernier. Cependant, il est possible que ces contraintes soient par la suite de nouveau violées par

la solution courante. Afin de ne pas générer plusieurs fois la même contrainte, chaque contrainte violée trouvée est ajoutée dans un pool. La séparation de chaque famille d'inégalités consiste alors à vérifier d'abord s'il existe des contraintes de cette famille dans le pool qui sont violées par la solution courante. Si de telles contraintes existent, alors elles sont ajoutées au programme linéaire courant et la séparation s'arrête. Dans le cas contraire, les procédures de séparation définies précédemment sont exécutées pour trouver de nouvelles inégalités violées qui n'ont jamais été générées par l'algorithme.

Par ailleurs, on remarque qu'il n'est pas nécessaire de brancher si nous obtenons une solution fractionnaire strictement inférieure à $|U_c|$. En effet, le système est singulier si et seulement si le programme linéaire en nombres entiers possède une solution égale à $|U_c|$. Par conséquent, si la solution optimale du programme où plusieurs contraintes d'intégrité sont relaxées est strictement inférieure à $|U_c|$ alors il n'existe pas de solution entière égale à $|U_c|$. Le système est donc régulier si la solution optimale fractionnaire est strictement inférieure à $|U_c|$. Il n'est donc pas nécessaire de brancher à partir d'un nœud de l'arbre de branchements si la solution fractionnaire associée est inférieure à $|U_c|$. Par contre, il est nécessaire d'explorer les autres nœuds de l'arbre de branchements.

5.3 Résultats expérimentaux

5.3.1 Contexte informatique

Avant de présenter les différents résultats expérimentaux que nous avons obtenus, nous donnons un bref descriptif des logiciels et du matériel informatique que nous avons utilisés.

L'algorithme de coupes et branchements décrit précédemment a été programmé en C++ en utilisant la librairie ABACUS [1], une librairie qui gère l'arbre de branchements. Pour la résolution des programmes linéaires, ABACUS fait appel à la librairie CPLEX 11.0 [41]. Nous avons utilisé l'algorithme de Galil, Micali, et Gabow [28] présenté dans la section 1.6 et implémenté dans la librairie COIN-OR LEMON Graph [4] afin de résoudre les problèmes de couplages parfaits de poids minimum. Nous avons aussi utilisé la librairie COIN-OR LEMON Graph pour résoudre les problèmes de flots de coût minimum, selon l'algorithme du network simplex [7, 16, 48]. Notre algorithme de coupes et branchements a été testé sur un Pentium core 2 duo 2,66 Ghz avec 2Go de mémoire vive, sous système Linux.

Dans un premier temps, nous allons montrer l'efficacité des contraintes que nous

avons proposées. Pour cela, nous testons un ensemble d'instances en utilisant différentes variantes de notre algorithme de coupes et branchements. Nous étudions ensuite la difficulté de résolution des instances. Dans cette étude, nous avons pu tester notre algorithme sur des instances de grande taille.

5.3.2 Analyse d'efficacité des contraintes valides

Tout d'abord, nous allons décrire le générateur d'instances aléatoires que nous avons utilisé pour comparer l'efficacité des différentes contraintes valides. Nous présentons ensuite les résultats obtenus par les différentes variantes de notre algorithme de coupes et branchements.

5.3.2.1 Description des instances testées

Nous avons généré des instances aléatoires afin de tester notre algorithme de coupes et branchements. On rappelle qu'une instance du PSGSCP est définie par le triplet (G, U_c, π) où $G = (U \cup V, E)$ est un graphe biparti, $U_c \subseteq U$ et $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U\}$ est une partition de E . Par ailleurs, les instances aléatoires sont générées de manière à respecter les conditions suivantes :

- (i) les sommets de V sont couverts par au moins une arête vraie/fausse,
- (ii) les sommets $u \in U_c$ sont couverts par au moins une arête vraie et une arête fausse,
- (iii) pour tout $F \in \mathcal{E}$, le graphe associé à $\{F\}$ ne contient pas de couplage parfait.

Si la condition (i) n'est pas vérifiée, alors nous sommes dans le cas polynomial présenté dans le chapitre 3. Si la condition (ii) ou (iii) n'est pas vérifiée, il est alors possible de transformer le graphe comme vu dans la remarque 4.1. Dans ce cas, le nombre de sommets représentant des équations conditionnelles est strictement inférieur à $|U_c|$.

La taille d'une instance aléatoire est définie par le nombre de sommets du graphe biparti ainsi que le nombre moyen d'arêtes incidentes à chaque sommet de l'ensemble U . Les ensembles U_c contiennent entre 10 et 40 sommets. Le nombre de sommets de $U \setminus U_c$ (correspondant aux équations non conditionnelles) est toujours égal à 5. Le nombre total de sommets de chaque graphe est donc compris entre 30 et 90. Par ailleurs, le nombre d'arêtes incidentes à chaque sommet de U est un entier aléatoire appartenant à l'ensemble $\{k-2, \dots, k+2\}$, où k est un entier fixé. Nous avons considéré des valeurs de $k = 5, 7, 9, 11, 13$, car il est assez rare d'avoir plus de 13 variables dans une équation algèbro-différentielle. Pour chaque taille d'instances, nous avons testé cinq instances. Nous considérons les moyennes des résultats obtenus pour ces cinq instances.

Nous avons proposé un générateur d'instances aléatoires permettant de vérifier les conditions (i)-(iii). Dans un premier temps, notre générateur construit un graphe biparti $G = (U \cup V, E)$ ne contenant pas d'arête $|E| = 0$. On définit ensuite, en fonction du paramètre k , le nombre d'arêtes incidentes à chaque sommet $u \in U$. On note $I(u)$ le nombre d'arêtes incidentes au sommet $u \in U$. Pour chaque sommet $v \in V$ on ajoute une arête vraie/fausse $uv \in E^{tf}$ où $u \in U$ est défini de manière aléatoire, respectant la borne $I(u)$ si $u \in U \setminus U_c$ et $I(u) - 2$ si $u \in U_c$. Ceci permet d'assurer que la condition (i) est respectée. Ensuite, pour chaque sommet $u \in U_c$, on ajoute une arête vraie $uv \in E_u^t$ et une arête fausse $uv' \in E_u^f$, où les sommets v et v' sont générés de manières aléatoires. On ajoute, de manière aléatoire, des arêtes vraies, fausses ou vraies/fausses, afin qu'il y est $I(u)$ arêtes incidentes pour chaque sommet $u \in U_c$. Nous finissons par ajouter, de manière aléatoire, des arêtes vraies/fausses afin qu'il y est $I(u)$ arêtes incidentes aux sommets $u \in U \setminus U_c$. Nous vérifions enfin la condition (iii). Si elle est vérifiée, alors l'instance générée est une instance valide, autrement on génère une autre instance. L'algorithme 3 ci-dessous illustre notre générateur d'instances.

5.3.2.2 Résultats obtenus

Nous avons testé plusieurs variantes de notre algorithme de coupes et branchements. Dans un premier temps, nous présentons les résultats si nous séparons uniquement les contraintes de couplages. Nous donnons ensuite les résultats quand nous améliorons la séparation des contraintes de couplages avec les contraintes de couplages à dominance maximale. Nous intégrons, dans une troisième variante, les contraintes de couplages proches. Enfin, dans une quatrième variante, nous intégrons les contraintes de 2-couplages. Nous avons fixé le temps maximum de résolution à une heure.

Nos résultats expérimentaux sont reportés dans les tables suivantes. Les différentes colonnes de ces tables représentent :

Algorithme 3: Générateur d'instance aléatoire

Entrées : - la taille des ensembles $|U| = |V| = n$, le nombre d'arêtes incidentes en moyenne k et le nombre de sommet de $|U_c| = z$

Sorties : - Un graphe biparti $G = (U \cup V, E)$ vérifiant les conditions (i)-(iii).

début

Construire le graphe $G = (U \cup V, E)$, avec $|E| = 0$, $|U| = |V| = n$;

pour chaque $v \in V$ **faire**

 Soit un booléen b égal à vrai;

tant que b est vrai **faire**

 Soit $u \in U$ un sommet défini de manière aléatoire;

si $u \in U \setminus U_c$ et $\delta(u) < I(u)$ **alors**

 | b égal faux

fin

si $u \in U_c$ et $\delta(u) < I(u) - 2$ **alors**

 | b égal faux

fin

fin

 Ajouter $uv \in E^{tf}$;

fin

pour chaque $u \in U_c$ **faire**

 Soient $v, v' \in V$ deux sommets définis de manière aléatoire;

 Ajouter $uv \in E_u^t$ et $uv' \in E_u^f$;

fin

pour chaque $u \in U$ **faire**

tant que $\delta(u) < I(u)$ **faire**

 Soient $v \in V$ un sommet défini de manière aléatoire et F un ensemble défini de manière aléatoire dans $\{E_u^t, E_u^f, E_u^{tf}\}$;

 Ajouter $uv \in F$;

fin

fin

pour chaque $u \in U \setminus U_c$ **faire**

tant que $\delta(u) < I(u)$ **faire**

 Soit $v \in V$ un sommet défini de manière aléatoire;

 Ajouter $uv \in E_u^{tf}$;

fin

fin

Construire $E = \cup_{u \in U} (E_u^{tf} \cup E_u^t \cup E_u^f)$ et $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U\}$;

si G ne vérifie pas la condition (iii) **alors**

 | relancer l'algorithme;

fin

retourner (G, U_c, π) ;

fin

n	: nombre de sommets correspondant aux équations conditionnelles de l'instance,
k	: entier définissant le degrés de chaque sommet de l'ensemble U compris entre $k - 2$ et $k + 2$
N_C	: nombre de contraintes de couplages générées,
N_{CDM}	: nombre de contraintes de couplages à dominance maximale générées,
N_{CP}	: nombre de contraintes de couplages proches générées,
N_{2CP}	: nombre de contraintes de 2-couplages générées,
T_{Csep}	: temps de séparation des contraintes de couplages, couplages à dominance maximale et de couplages proches en secondes,
T_{2CP}	: temps de séparation des contraintes de 2-couplages en secondes,
No	: nombre de nœuds dans l'arbre de branchements,
o/p	: nombre de problèmes résolus à l'optimum sur le nombre d'instances testées,
CPU	: temps de résolution en secondes.

Le temps T_{Csep} regroupe les temps de séparations des contraintes de couplages, couplages à dominance maximale et de couplages proches. En effet, suivant la variante utilisé nous essayons améliorer la contrainte il est donc nécessaire de considérer le temps global.

Notre première série d'expérimentations concerne la variante de notre algorithme de coupes et branchements dans laquelle seules les contraintes de couplages (3.11) sont séparées. Cette variante consiste à résoudre la formulation (P) sans ajouter de contraintes pour renforcer la relaxation linéaire. Nous considérons des instances contenant entre 10 et 40 équations conditionnelles avec $k \in \{5, 7, 9, 11, 13\}$. La table 5.1 rapporte la moyenne des résultats obtenus pour ces instances. On remarque que pour les instances contenant moins de 20 équations conditionnelles, toutes les instances ont été résolues à l'optimum durant le temps imparti. Par contre, pour les instances contenant au moins 25 équations conditionnelles, quasiment aucune instance n'a pu être résolue.

On peut également remarquer que pour la plupart des instances, un très grand nombre de contraintes de couplages (3.11) ont été générées. La séparation des contraintes est relativement rapide au vu du nombre de contraintes de couplages violées trouvées. Cependant, comme le nombre de contraintes de couplages (3.11) est très grand, le temps de séparation représente entre 5 et 10% du temps de résolution. De plus, on peut voir que le nombre de nœuds dans l'arbre de branchements est relativement grand pour les instances contenant plus de 15 équations conditionnelles, de l'ordre de plusieurs milliers.

n	k	N_C	T_{Csep}	No	o/p	CPU
10	5	48,2	0	7,6	5/5	0
10	7	69,6	0,2	6,8	5/5	0,4
10	9	62,4	0	5,6	5/5	0,4
10	11	61	0,2	6	5/5	0,6
15	5	238,2	0,4	21,2	5/5	1,6
15	7	625,2	1	58,8	5/5	5,8
15	9	386,2	0,4	35,2	5/5	3,2
15	11	548	1	51,6	5/5	5,6
15	13	600,6	1,8	55,6	5/5	6,4
20	5	3554	7	406	5/5	44,8
20	7	18086,8	55	2163,6	4/5	891,6
20	9	14864,2	55	1662,4	0/5	489,2
20	11	12487,8	58,6	1448,8	0/5	465,2
20	13	16405	82,6	1885,2	0/5	643,4
25	5	37441,8	121,4	4488,4	2/5	2320,4
25	7	56409,8	243,2	7366,8	0/5	>1h
25	9	57494,8	251,4	7434	0/5	>1h
25	11	56116,6	296,2	7324	0/5	>1h
25	13	54428,6	321,8	7462,4	0/5	>1h
30	5	61900,6	169,6	6317,2	1/5	3406,8
30	7	60309,2	225,6	6744,8	0/5	>1h
30	9	62662,8	263,4	6502,4	0/5	>1h
30	11	60208	297,2	6571,2	0/5	>1h
30	13	60655,6	333,6	6406,8	0/5	>1h
35	5	64168,4	196	6762,4	0/5	>1h
35	7	62128,4	230,2	6156	0/5	>1h
35	9	60982	301	5987,2	0/5	>1h
35	11	60283,6	315,2	5970,4	0/5	>1h
35	13	59724,2	343,8	5853,2	0/5	>1h
40	5	47083	206,8	4923,6	0/5	>1h
40	7	57308,8	344,2	5656,4	0/5	>1h
40	9	55737,6	400,2	5512,4	0/5	>1h
40	11	54478	422,2	5420,4	0/5	>1h
40	13	53479,4	479,2	5346,8	0/5	>1h

TABLE 5.1 – Résultat pour l'algorithme utilisant les contraintes de couplages.

Nous allons désormais voir l'impact des contraintes de couplages à dominance maximale.

La table 5.2 donne les résultats expérimentaux obtenus par l'algorithme de coupes et branchements lorsque les contraintes de couplages (3.11) violées ajoutées à la relaxation linéaire courante sont uniquement des contraintes de couplages à dominance maximale (4.13).

L'algorithme utilise alors l'algorithme de séparation donné dans la section 5.1.2. Les instances testées sont les mêmes que celles utilisées pour la première variante de l'algorithme de coupes et branchements. On peut noter que toutes les instances sauf six ont été résolues à l'optimum. En effet, deux instances avec $n = 30$, une instance avec $n = 35$ et trois instances avec $n = 40$, pour lesquelles chaque équation contient à peu près 5 variables, n'ont pu être résolues à l'optimum.

On remarque que pour toutes les instances résolues à l'optimum, un faible nombre de contraintes de couplages à dominance maximale (4.13) ont été générées, excepté pour les instances contenant 35 équations conditionnelles et dont chaque équation contient à peu près 7 variables, pour lesquelles environ 1000 contraintes (4.13) violées sont générées en moyenne. On peut également noter que le temps passé dans la séparation des contraintes de couplages à dominance maximale est relativement faible par rapport au nombre de contraintes violées générées. Finalement, on peut remarquer que pour les instances résolues à l'optimum, le nombre de nœuds dans l'arbre de branchements est relativement faible. Pour la majorité de ces instances, ce nombre est inférieur à 10.

La comparaison des résultats expérimentaux donnés par les tables 5.1 et 5.2 permet de mettre en valeur l'intérêt des contraintes de couplages à dominance maximale (4.13). En effet, grâce à ces contraintes, beaucoup d'instances qui n'avaient pu être résolues durant le temps imparti sont maintenant résolues en moins d'une seconde. De plus, quasiment toutes les instances ont pu être résolues. Finalement, le nombre de nœuds dans l'arbre de branchements est beaucoup plus restreint. Ces résultats montrent l'intérêt de l'étude polyédrale développée au chapitre 4. La condition nécessaire de dominance maximale donnée dans cette étude permet de limiter la recherche de contraintes de couplages aux seules contraintes intéressantes. Grâce à cette dominance maximale, un nombre réduit de contraintes violées est ajouté à la relaxation linéaire et ces contraintes sont les contraintes de couplages les plus fortes.

Nous allons voir l'efficacité des contraintes de couplages proches.

La table 5.3 rapporte la moyenne des résultats obtenus lorsque les contraintes de cou-

n	k	N_{CDM}	T_{Csep}	No	o/p	CPU
10	5	10,8	0	4,8	5/5	0
10	7	7	0	3,6	5/5	0
10	9	4,4	0	2	5/5	0
10	11	4,6	0	2	5/5	0
15	5	10,2	0	4,4	5/5	0
15	7	8,8	0	2,8	5/5	0
15	9	4	0	2	5/5	0,2
15	11	4	0	2	5/5	0
15	13	4	0	2	5/5	0
20	5	328,2	1,2	122,4	5/5	3,8
20	7	127	1,2	28,8	5/5	2
20	9	15,2	0	5,6	5/5	0,2
20	11	7,4	0	2,4	5/5	0
20	13	5	0,2	2,8	5/5	0,2
25	5	86,4	0,6	38,4	5/5	1,8
25	7	36,4	0,2	11,2	5/5	1
25	9	9,2	0,2	3,6	5/5	1
25	11	5	0	2,8	5/5	0,6
25	13	6,2	0	2,8	5/5	0,8
30	5	7661,4	38	2786	3/5	152,2
30	7	89	0,4	20,4	5/5	1,6
30	9	22,4	0	6,4	5/5	0,4
30	11	6,6	0,2	2,8	5/5	0
30	13	4	0,2	2	5/5	0,2
35	5	8422,2	50,8	2118	4/5	211
35	7	1020,4	7,8	262,4	5/5	21,8
35	9	13,8	0,4	7,2	5/5	0,8
35	11	11	0,2	4	5/5	0,4
35	13	5,2	0,2	2,8	5/5	0,4
40	5	29212,6	272,2	9307,6	2/5	1625,4
40	7	23,4	0,6	14,8	5/5	1,4
40	9	8	0	2,8	5/5	1
40	11	5,8	0	2,8	5/5	1
40	13	4	0	2	5/5	0,6

TABLE 5.2 – Résultat pour l'algorithme utilisant les contraintes de couplages à dominance maximale.

plages proches (4.19) sont utilisées pour renforcer la relaxation linéaire. Les instances testées sont les mêmes que celles testées précédemment. Comme pour la précédente variante, les contraintes de couplages considérées sont toutes des contraintes de couplages à dominance maximale. On note que toutes les instances sauf deux ont été résolues à l'optimum en moins d'une heure. Seules deux instances contenant respectivement 35 et 40 équations conditionnelles et $k = 5$ n'ont pu être résolues.

Le nombre de contraintes de couplages proches (4.19) générées est généralement faible. Cependant, pour certaines instances, notamment celles qui n'ont pas pu être résolues à l'optimum, un très grand nombre de ces contraintes ont été générées. Le nombre de contraintes de couplages à dominance maximale (4.13) violées générées est très faible. En effet, au plus une contrainte (4.13) en moyenne est ajoutée à la relaxation linéaire courante. La plupart du temps, aucune contrainte de couplages à dominance maximale n'est ajoutée. Ceci s'explique par le fait que presque toutes les contraintes de couplages à dominance maximale peuvent être étendues en une (ou plusieurs) contraintes de couplages proches à l'aide de la séparation donnée dans la section 5.1.3.

Le temps nécessaire à la résolution des problèmes de séparation des contraintes est très faible. Finalement, on remarque que pour les instances résolues à l'optimum durant le temps imparti, le nombre de nœuds dans l'arbre de branchements est généralement très faible. Ce nombre est corrélé au nombre moyen de variables apparaissant dans les équations. Plus ce dernier est élevé, plus le nombre de nœuds est petit.

Les résultats expérimentaux donnés par les tables 5.2 et 5.3 montrent l'importance des contraintes de couplages proches (4.19). En effet, les résultats de la table 5.3 montrent que quasiment toute contrainte de couplages à dominance maximale peut être étendue en une ou plusieurs contraintes de couplages proches. Cela implique que les contraintes (4.13) trouvées ne définissent quasiment jamais des facettes. De plus, la prise en compte des contraintes de couplages proches (4.19) diminue systématiquement le temps de résolution sauf pour les instances à 15 et 35 équations conditionnelles avec $k = 5$. De même, le nombre de nœuds dans l'arbre de branchements est quasiment toujours plus faible. Les contraintes de couplages proches renforcent donc considérablement la relaxation linéaire.

Enfin nous allons présenter les résultats obtenus quand nous ajoutons les contraintes de 2-couplages.

La table 5.4 rapporte la moyenne des résultats obtenus lorsque toutes les contraintes, couplages à dominance maximale, couplages proches et 2-couplages, sont utilisées pour renforcer la relaxation linéaire. Les instances testées sont les mêmes que celles tes-

n	k	N_{CDM}	N_{CP}	T_{Csep}	No	o/p	CPU
10	5	0,4	8,2	0	0,4	5/5	0
10	7	0	8,8	0	1,6	5/5	0
10	9	0	4	0	0	5/5	0
10	11	0	4	0	0	5/5	0
15	5	0,4	15	0,2	1,6	5/5	0,2
15	7	0	5,8	0	0	5/5	0
15	9	0	4	0	0	5/5	0
15	11	0	4	0	0	5/5	0
15	13	0	4	0	0	5/5	0
20	5	0	103,6	0,2	11,6	5/5	0,6
20	7	0,2	131	0,4	22,8	5/5	1,2
20	9	0	18,4	0	10	5/5	0,2
20	11	0	5	0	0	5/5	0
20	13	0	5,6	0	0,4	5/5	0
25	5	0	66,4	0	10	5/5	0,2
25	7	0	37,8	0,2	8,8	5/5	0,2
25	9	0	9,4	0	0,8	5/5	0
25	11	0	5,6	0,2	0,4	5/5	0,2
25	13	0	5,6	0	0,4	5/5	0
30	5	0,6	2103,8	9,8	150,8	5/5	19,2
30	7	0	54	0,6	6	5/5	0,8
30	9	1	13,4	0	3,2	5/5	0,4
30	11	0	5,6	0	0,4	5/5	0
30	13	0	4	0	0	5/5	0
35	5	0	34937,2	129	3736	4/5	606,2
35	7	0	1733,4	7	229,2	5/5	20,4
35	9	0	15,2	0	1,6	5/5	0
35	11	0	7,6	0	0,4	5/5	0,2
35	13	0	5,6	0,2	0,4	5/5	0,2
40	5	0	15109,6	97,6	926,4	4/5	237,2
40	7	0	68,2	0,4	12,4	5/5	1,4
40	9	0	5,8	0,2	0,4	5/5	0,2
40	11	0	5,6	0	0,4	5/5	0
40	13	0	4	0,2	0	5/5	0,6

TABLE 5.3 – Résultat pour l’algorithme utilisant les contraintes de couplages proches.

tées précédemment. Comme pour la précédente variante, les contraintes de couplages considérées sont toutes des contraintes de couplages à dominance maximale. On note que toutes les instances ont été résolues à l'optimum et en moyenne en moins de 16 secondes.

Les résultats expérimentaux donnés par les tables 5.3 et 5.4 montrent l'importance des contraintes de 2-couplages (5.6). En effet, les résultats de la table 5.4 montrent qu'avec en moyenne moins de 7 contraintes de 2-couplages, les temps de calcul diminuent de manière significative. De même, le nombre de noeuds dans l'arbre de branchement est très faible. De plus, 69% des instances ont pu être résolues à la racine. Les contraintes de 2-couplages renforcent considérablement la relaxation linéaire.

Nous allons voir, dans la prochaine section, que ce n'est pas seulement la densité ou la taille du graphe qui influence les performances mais aussi la cardinalité du couplage maximum dans le graphe composé uniquement des arêtes vraies/fausses $G' = (U \cup V, E^{tf})$.

5.3.3 Analyse des paramètres des instances

Nous allons voir quelques paramètres permettant de définir les instances difficiles à résoudre. Dans un premier temps nous allons voir l'impact du paramètre k , puis nous verrons l'impact de la cardinalité du couplage maximum du graphe composé uniquement des arêtes vraies/fausses.

5.3.3.1 Paramètre k

Afin d'analyser la difficulté de résolution des instances, nous avons dans un premier temps essayé notre algorithme sur des instances de plus grande taille. Nous avons donc généré plusieurs instances en utilisant le même générateur d'instances. Les ensembles U_c contiennent entre 200 et 300 sommets. Le nombre de sommets de $U \setminus U_c$ (correspondant aux équations non-conditionnelles) est toujours égal à 20. Nous obtenons des graphes ayant entre 440 et 640 sommets. Nous avons considéré les valeurs 5,9 et 13 pour k . Pour chaque taille d'instance, nous avons testé cinq instances. Nous considérons les moyennes des résultats obtenus pour ces cinq instances. Afin de tester la difficulté de résolution des instances nous avons augmenté le temps maximum d'exécution à cinq heures.

n	k	N_{CDM}	N_{2CP}	N_{CP}	T_{Csep}	T_{2CP}	No	o/p	CPU
10	5	0,4	0,8	6,2	0,2	0	0,8	5/5	0,2
10	7	0	1,2	8,8	0	0	1,6	5/5	0
10	9	0	1	4	0	0	0	5/5	0,2
10	11	0	1,2	4	0	0	0	5/5	0
15	5	0,8	0,8	11,6	0	0	1,6	5/5	0
15	7	0	1,2	5,8	0	0	0	5/5	0
15	9	0	1	4	0	0	0	5/5	0
15	11	0	1	4	0	0	0	5/5	0
15	13	0	0,8	4	0	0	0	5/5	0
20	5	0	6,8	198,4	0,6	0	34	5/5	1,4
20	7	0,8	4,2	278	1,2	0	55,2	5/5	3
20	9	0	2,4	12,4	0	0	1,6	5/5	0,2
20	11	0	1,8	5	0,2	0	0	5/5	0,2
20	13	0	1	5,6	0	0	0,4	5/5	0,2
25	5	0	1,4	49	0,2	0	6	5/5	0,2
25	7	0	2,8	33,4	0	0	6,8	5/5	0,4
25	9	0	1,4	9,4	0	0	0,8	5/5	0,2
25	11	0	1	4	0	0	0	5/5	0
25	13	0	1	5,6	0	0	0,4	5/5	0
30	5	0	1,8	29,4	0	0	2	5/5	0,2
30	7	0,2	4,8	71,8	0,2	0	11,6	5/5	1
30	9	1,2	4	13,8	0,2	0	2,4	5/5	0,6
30	11	0	1	5,6	0	0	0,4	5/5	0,2
30	13	0	1	4	0	0	0	5/5	0,2
35	5	2	6,2	1381,2	7,2	0	147,2	5/5	13,2
35	7	0	4	1550,6	6,8	0	198,8	5/5	16
35	9	0	1	15,2	0	0	1,6	5/5	0,4
35	11	0	3,2	9,6	0,2	0	0,4	5/5	0,6
35	13	0	1	4	0	0	0	5/5	0,2
40	5	0	2,8	180,6	1,8	0	35,2	5/5	2,4
40	7	0	1,2	68,2	0,8	0	12,4	5/5	1,2
40	9	0	1,2	5,8	0	0	0,4	5/5	0,4
40	11	0	1	5,6	0	0	0,4	5/5	0,4
40	13	0	1	4	0	0	0	5/5	0,2

TABLE 5.4 – Résultat pour l'algorithme utilisant toutes les contraintes.

n	k	N_{CDM}	N_{2CP}	N_{CP}	T_{Csep}	T_{2CP}	CPU	No	o/p
200	5	0	0,8	28,8	2,09	1,6	4	0	5/5
200	9	0	1	149159,2	6564,31	11213,4	18123,2	21584,4	1/5
200	13	0	1	31453,6	1313,09	3145,6	4527,8	2977,6	5/5
225	5	0	1	250,4	4,18	16,4	20,8	0,8	5/5
225	9	0	2	124606,8	5074,52	11085,4	16427	14967,2	3/5
225	13	0	1	40070,6	1732,61	4861,4	6685,2	3762,8	5/5
250	5	0	2,2	2130,2	211,28	153,4	375,8	86	5/5
250	9	0	2,2	104283,8	6191,53	10567,6	17085	18835,6	1/5
250	13	0	1	11468,8	377,53	1724,8	2122,2	1022	5/5
275	5	0	1,25	136,75	7,6	9	17,5	0	5/5
275	9	0	3	94259,2	4456,83	10598,2	15289,6	14334	4/5
275	13	0	1	7577,6	265,24	1341,2	1620,4	714,8	5/5
300	5	0	2	623,3	16,15	55	72	0,7	5/5
300	9	0	1,6	78971,4	4143,9	10179	14541	8996	5/5
300	13	0	1	17203,2	940,12	3355	4344,6	1431,6	5/5

TABLE 5.5 – Résultat pour l'algorithme afin de tester le paramètre k .

La table 5.5 nous indique que toutes les instances avec $k = 5$ ont pu être résolues en quelques minutes, alors que les instances avec $k = 9$ sont rarement résolues en moins de cinq heures par notre algorithme pour les instances possédant entre 200 et 250 équations conditionnelles. On remarque que toutes les instances possédant 300 équations conditionnelles ont été résolues à l'optimum. Nous pouvons en déduire que la densité joue un rôle dans la difficulté de résolution des instances. De plus, nous déduisons que le nombre d'équations conditionnelles n'est pas un critère suffisant pour définir la difficulté de résolution des instances.

Nous avons cherché un autre paramètre permettant d'expliquer ces résultats. Pour cela, nous avons utilisé un second générateur d'instances aléatoires. Ce générateur permet de faire varier un paramètre différent de la taille de l'instance et de la densité du graphe.

5.3.3.2 Paramètre $\nu(H_\emptyset)$

Nous allons définir dans cette section un autre paramètre permettant de donner des informations sur la difficulté de résolution des instances. Ce paramètre correspond à la

cardinalité du couplage de taille maximum dans le graphe $G' = (U \cup V, E^{tf})$. G' est le graphe associé à l'ensemble vide. Cette cardinalité correspond alors à $\nu(H_\emptyset)$.

Description des instances testées

Nous avons généré d'autres instances aléatoires afin de tester notre algorithme de coupes et branchements. Ces instances aléatoires du PSGSCP sont définies par le triplet (G, U_c, π) où $G = (U \cup V, E)$ est un graphe biparti, $U_c \subseteq U$ et $\pi = \{E_u^t, E_u^f, E_u^{tf} : u \in U\}$ est une partition de E . Le générateur que nous allons présenter génère des instances qui respectent toujours les conditions (i)-(iii). Nous respectons aussi une autre condition :

(iv) les sommets de U sont couverts par au moins une arête vraie/fausse.

Cette condition nous permet de générer plus facilement nos graphes.

Nous avons proposé un générateur d'instances aléatoires permettant de vérifier les conditions (i)-(iv). Notre générateur construit un graphe biparti $G = (U \cup V, E)$ possédant le bon nombre de sommets et contenant les arêtes E^{tf} vérifiant les conditions (i) et (iv) et tel que pour tout sommet $u \in U$, $\delta(u) = 5$. Ensuite, pour chaque sommet $u \in U_c$, on ajoute une arête vraie $uv \in E_u^t$ et une arête fausse $uv' \in E_u^f$, où les sommets v et v' sont générés de manière aléatoire et vérifiant la condition (iii). Enfin, on ajoute les arêtes vraies ou fausses, afin qu'il y ait 10 arêtes incidentes pour chaque sommet $u \in U_c$, tout en vérifiant toujours la condition (iii).

Une instance aléatoire est définie par le nombre de sommets du graphe biparti ainsi que la cardinalité du couplage maximum dans le sous-graphe couvrant composé uniquement des arêtes vraies/fausses $G' = (U \cup V, E^{tf})$. Dans notre générateur, nous avons fixé le nombre d'arêtes incidentes à chaque sommet de l'ensemble U , afin de ne pas perturber notre analyse. Nous considérons 5 arêtes vraies/fausses et 5 arêtes vraies ou fausses incidentes à chaque sommet $u \in U_c$ et uniquement 5 arêtes vraies/fausses à chaque sommet $u \in U \setminus U_c$. Les graphes bipartis générés contiennent entre 150 et 1300 sommets appartenant à U_c (correspondant aux équations conditionnelles). Le nombre de sommets appartenant à $U \setminus U_c$ (correspondant aux équations non conditionnelles) est toujours égal à 5. Nous obtenons donc des graphes contenant entre 310 et 2610 sommets. Pour chaque taille d'instances, nous avons testé dix instances. Nous considérons les moyennes des résultats obtenus pour ces dix instances.

Nous utilisons la variante de notre algorithme de coupes et branchements qui utilise les contraintes de couplages à dominance maximale, couplage proches et 2-couplages.

n	N_{CDM}	N_{2CP}	N_{CP}	T_{2CP}	T_{Csep}	No	o/p	CPU
150	0	16,8	404,8	2,1	12,6	4,6	10/10	45
175	0	23,1	520,9	3,2	20,8	7	10/10	82,3
200	0	46,8	1111,8	12	65,5	18,4	10/10	251,3
225	0	10,4	211,6	2,2	15	2,8	10/10	63,9
250	0	30,6	774,4	9,2	48,5	8,8	10/10	239,6
275	0	25,9	519,1	8,2	41,8	6,8	10/10	231
300	0	23,7	633,4	9	52,7	7,6	10/10	264,9
325	0	12,7	264,5	3,8	25,2	2,8	10/10	169,2
350	0	41,8	930,1	15,6	110	11,2	10/10	636,6
375	0	32,9	775	17,1	92,8	9,8	10/10	558,6
400	0	7,8	195,6	4,7	27,4	2	10/10	175,6
425	0	22,8	509,8	11,3	74,6	5,8	10/10	530,9
450	0	15,9	413	9,1	62,4	4,4	10/10	423,4
475	0	23,75	463	13,75	90	5,5	10/10	698
900	0	22,8	630,5	2606,7	312,3	6,8	10/10	2934,9
1000	0	18,1	540,7	2658,7	334,7	5,8	10/10	3012,9
1100	0	42,1	1277,8	8740,1	967,3	13,6	9/10	9738,3
1200	0	37,3	962,2	8911,1	889,2	8,4	8/10	9829,4
1300	0	19,5	661,2	12089,9	702,9	4,8	9/10	12818,5

TABLE 5.6 – Résultat pour l'algorithme de coupes et branchements avec $\nu(H_\emptyset) = |U_c| - 10$

Résultats obtenus

Nous avons testé notre algorithme sur trois séries d'instances telles que $\nu(H_\emptyset) = |U_c| - 10$, $\nu(H_\emptyset) = |U_c| - 20$ et $\nu(H_\emptyset) = |U_c| - 25$, où M^{tf} représente le couplage maximum contenant uniquement des arêtes vraies/fausses.

Nos résultats expérimentaux sont reportés dans les tables suivantes. Les différentes colonnes de ces tables sont données dans la section 5.3.2.2.

Pour chaque taille d'instances, nous avons testé dix instances. Nous considérons les moyennes des résultats obtenus pour ces dix instances. Nous avons fixé le temps maximum de résolution à cinq heures.

n	N_{CDM}	N_{2CP}	N_{CP}	T_{2CP}	T_{Csep}	No	o/p	CPU
150	0	257,2	31695,1	1270,72	1135,4	504,9	10/10	2723,8
175	0	343,8	40432,6	2238,96	1795,4	598,9	10/10	4594,1
200	0	268,3	32209,7	1996,72	1661,7	533,1	9/10	4157,6
225	0	335,9	37197,8	3217,68	2292,8	583,6	10/10	6314,9
250	0	292,2	34417,6	3107,76	2457,2	544,9	10/10	6341,9
275	0	226,8	26346,7	2892,4	2129,9	414	10/10	5745,4
300	0	214	26722,2	3342,96	2463,9	421,6	10/10	6642,6
325	0	245,8	31104,3	4353,28	3157,3	506,7	10/10	8598,9
350	0	273,6	35031,2	5722,72	3962,7	583,8	10/10	11116,1
375	0	291	34898,2	6586,4	4337,6	553,1	9/10	12570,6
400	0	223,7	27581,7	5543,28	3620,1	398,3	10/10	10549,2
425	0	239,4	30826,6	7282,4	4396,3	476,2	8/10	13499,3
450	0	209,6	26194,7	6476,7	4037,8	398,8	8/10	12133,7
475	0	263,1	29077,8	8848	4889,2	427,6	4/10	15949,2

TABLE 5.7 – Résultat pour l’algorithme de coupes et branchements avec $\nu(H_\emptyset) = |U_c| - 20$

La table 5.6 rapporte les résultats obtenus lorsque les instances ont été générées avec $\nu(H_\emptyset) = |U_c| - 10$. Les résultats expérimentaux donnés par la table 5.6 montrent que ces instances sont faciles à résoudre. En effet, en moyenne, le nombre de nœuds dans l’arbre de branchements est inférieur à 20. De plus, l’algorithme génère relativement peu de contraintes pour trouver l’optimum.

La table 5.7 rapporte les résultats obtenus lorsque les instances ont été générées avec $\nu(H_\emptyset) = |U_c| - 20$. Les résultats expérimentaux donnés par la table 5.7 montrent que notre algorithme résout la plupart des instances contenant au plus 450 équations conditionnelles énormément. Par contre, seules les instances avec $n = 150$ sont résolues en moins d’une heure. On remarque que nous avons en moyenne 500 nœuds dans l’arbre de branchement quelle que soit les tailles d’instances que nous avons testées. De même, le nombre de contraintes générées est en moyenne à peu près le même quelle que soit l’instance testée considérée. On peut en déduire que la difficulté pour résoudre un système ne dépend pas seulement de la taille de l’instance.

n	N_{CDM}	N_{2CP}	N_{CP}	T_{2CP}	T_{Csep}	No	o/p	CPU
150	0	191,4	215364	985,2	5645,4	3767,2	0/10	>5h
175	0	145,3	153122,6	876,2	4980,7	2685,6	2/10	14821,3
200	0	173	171514,5	1190,75	6625,5	2926	0/10	>5h

TABLE 5.8 – Résultat pour l'algorithme de coupes et branchements avec $\nu(H_\theta) = |U_c| - 25$

La table 5.8 rapporte les résultats obtenus lorsque les instances ont été générées avec $\nu(H_\theta) = |U_c| - 25$. Les résultats expérimentaux donnés par la table 5.8 montrent qu'il est très difficile de résoudre ces instances. Ceci confirme que la difficulté pour résoudre une instance dépend aussi de $\nu(H_\theta)$.

5.4 Conclusion

Nous avons présenté dans ce chapitre des algorithmes exacts et heuristiques pour la séparation de la plupart des contraintes présentées dans les chapitres précédents. Grâce à ces algorithmes, nous avons développé un algorithme de coupes et branchements permettant de résoudre le PSGSCP. Nous avons testé l'efficacité des différentes familles d'inégalités utilisées pour renforcer la relaxation linéaire. Des instances contenant jusqu'à 1300 équations différentielles conditionnelles ont ainsi pu être résolues grâce à cet algorithme de coupes et branchements.

Une étude approfondie des instances difficiles à résoudre pourrait nous permettre de trouver de nouvelles contraintes nous permettant de résoudre plus facilement ces instances.

Chapitre 6

Analyse structurelle des systèmes algébro-différentiels conditionnels imbriqués

Dans ce chapitre, nous présentons d'abord une extension des SADC lorsqu'une équation différentielle conditionnelle est associée à plusieurs conditions. Nous présentons ensuite le problème de l'analyse structurelle pour de tels systèmes et donnons une formulation de ce problème en termes de graphes. Nous présentons finalement une formulation du problème à l'aide d'un programme linéaire en nombres entiers.

6.1 Systèmes algébro-différentiels conditionnels imbriqués et analyse structurelle

Dans cette section, nous présentons les systèmes algébro-différentiels conditionnels imbriqués. Ces systèmes sont une généralisation des SADC. Ils permettent de prendre en compte des systèmes plus complexes où plusieurs conditions peuvent être associées à une équation conditionnelle et certaines implications logiques entre les conditions sont considérées.

Une *équation différentielle conditionnelle imbriquée* (EDCI) est une équation dont la valeur dépend d'une ou plusieurs conditions. Un exemple d'EDCI est donné par l'équation suivante :

$$\begin{aligned}
 eq_1 : & \text{ if } a > 0 & (6.1) \\
 & \text{ then} \\
 & \quad \text{if } b > 0 \\
 & \quad \text{ then} & 0 = \dot{x}_2, \\
 & \quad \text{ else} & 0 = 3x_1, \\
 & \text{ else} \\
 & \quad \text{if } c > 0 \\
 & \quad \text{ then} & 0 = 3x_3 + \dot{x}_2, \\
 & \quad \text{ else} & 0 = x_4.
 \end{aligned}$$

L'équation (6.1) est associée aux 3 conditions $a > 0$, $b > 0$ et $c > 0$. Suivant les valeurs (vraies ou fausses) de ces conditions, l'équation (6.1) génère différentes équations non-conditionnelles. Ainsi, si les conditions $a > 0$ et $b > 0$ sont vraies, l'équation (6.1) correspond alors à l'équation

$$0 = \dot{x}_2.$$

Si la condition $a > 0$ est fausse et la condition $c > 0$ est vraie, alors (6.1) est équivalente à

$$0 = 3x_3 + \dot{x}_2.$$

Les équations différentielles conditionnelles imbriquées généralisent donc les EDC. Étant donnée une EDCI, on appelle *affectation d'équation* toute valeur vraie ou fausse d'une partie (ou la totalité) des conditions associées à l'équation permettant de définir la valeur de cette équation. Ainsi, pour l'équation (6.1), l'affectation $a > 0$ vrai et $b > 0$ faux est une affectation d'équation. Une affectation des conditions est représentée par un ensemble de conditions permettant de définir si les conditions sont vraies ou fausses. Ainsi, $\{a > 0, b \leq 0\}$ correspond à l'affectation d'équation $a > 0$ vrai et $b > 0$ faux.

Sans perte de généralité, on suppose que chaque affectation d'une EDCI génère une unique équation non-conditionnelle. Dans le cas contraire, il est possible de réécrire

l'EDCI en plusieurs EDCI, chacune générant une unique équation. Par exemple, l'EDCI

$$\begin{aligned}
 eq_2 : & \text{ if } d > 0 & (6.2) \\
 & \text{ then} \\
 & \quad \text{if } e > 0 \\
 & \quad \text{ then} & 0 = 4x_2 + \dot{x}_1, \\
 & \quad \text{ else} & 0 = x_3, \\
 & \quad \text{if } f > 0 \\
 & \quad \text{ then} & 0 = x_1, \\
 & \quad \text{ else} & 0 = x_4 + x_3, \\
 & \text{ else} \\
 & & 0 = x_1 + \dot{x}_4 + 1, \\
 & & 0 = x_4 + \dot{x}_2 + 1.
 \end{aligned}$$

peut être remplacée par les 2 EDCI

$$\begin{aligned}
 eq'_2 : & \text{ if } d > 0 & (6.3) \\
 & \text{ then} \\
 & \quad \text{if } e > 0 \\
 & \quad \text{ then} & 0 = 4x_2 + \dot{x}_1, \\
 & \quad \text{ else} & 0 = x_3, \\
 & \text{ else} \\
 & & 0 = x_1 + \dot{x}_4 + 1,
 \end{aligned}$$

$$\begin{aligned}
 eq''_2 : & \text{ if } d > 0 & (6.4) \\
 & \text{ then} \\
 & \quad \text{if } f > 0 \\
 & \quad \text{ then} & 0 = x_1, \\
 & \quad \text{ else} & 0 = x_4 + x_3, \\
 & \text{ else} \\
 & & 0 = x_4 + \dot{x}_2 + 1.
 \end{aligned}$$

Un *système algébro-différentiel conditionnel imbriqué* (SADCI) est un SAD comportant des EDCI. Le système donné par les équations (6.1), (6.3), (6.4) et

$$eq_3 : \text{if } a > 0 \tag{6.5}$$

$$\text{then } 0 = 6\dot{x}_1 + 2,$$

$$\text{else } 0 = 3\dot{x}_2 + x_3 + 3 + x_4,$$

$$eq_4 : 0 = 6\dot{x}_5 + x_1 + 1, \tag{6.6}$$

est un exemple de SADCI. Notons (S_5) ce système composé de 4 EDCI et d'une équation non conditionnelle. Les SADCI généralisent les SADC présentés dans le chapitre 2 puisque ces derniers sont des SADCI dans lesquels les EDCI sont associées à une unique condition et chaque condition apparaît dans une seule équation.

Étant donné un SADCI, on note par $\mathcal{C} = \{C_1, \dots, C_m\}$ l'ensemble des affectations d'équations des différentes EDCI du système. Cet ensemble est appelé *ensemble des affectations du système*. On remarque que la cardinalité de cet ensemble est polynomiale par rapport à la taille de l'instance puisqu'elle est inférieure ou égale au nombre d'équations non-conditionnelles pouvant être générées par les EDCI du système. En effet, il n'y a pas obligatoirement égalité puisque 2 EDCI peuvent avoir une même affectation de conditions.

Pour le système (S_5) défini précédemment, l'ensemble des affectations du système est donné par : $C_1 = \{a > 0, b > 0\}$, $C_2 = \{a > 0, b \leq 0\}$, $C_3 = \{a \leq 0, c > 0\}$, $C_4 = \{a \leq 0, c \leq 0\}$, $C_5 = \{d > 0, e > 0\}$, $C_6 = \{d > 0, e \leq 0\}$, $C_7 = \{d > 0, f > 0\}$, $C_8 = \{d > 0, f \leq 0\}$, $C_9 = \{d \leq 0\}$, $C_{10} = \{a > 0\}$ et $C_{11} = \{a \leq 0\}$. Remarquons que l'affectation d'équations C_9 génère 2 équations non-conditionnelles. Ces deux équations sont données par les EDCI (6.3) et (6.4).

On suppose que les différentes conditions du système sont indépendantes. Ainsi, pour le système (S_5) , les valeurs des conditions $a > 0$, $b > 0$, $c > 0$, $d > 0$, $e > 0$ et $f > 0$ sont indépendantes. Si cela peut sembler restrictif, il faut remarquer que les conditions peuvent être très complexes et dépendre de nombreux paramètres. Ainsi, déterminer les implications logiques entre les différentes conditions est un problème difficile qui n'entre pas dans le cadre du travail de cette thèse.

Bien que les conditions du système soient indépendantes, il existe cependant des incompatibilités logiques entre les différentes affectations du système. À titre d'exemple, si l'on considère les affectations $C_1 = \{a > 0, b > 0\}$ et $C_{11} = \{a \leq 0\}$, celles-ci impliquent que la condition $a > 0$ soit à la fois vraie et fausse, ce qui nous donne une contradiction. Deux affectations sont dites *incompatibles* si elles contiennent deux valeurs différentes pour une même condition. Dans le cas contraire, elles sont dites

compatibles. Il existe aussi des implications logiques entre les différentes affectations du système. À titre d'exemple, si l'on considère les affectations $C_1 = \{a > 0, b > 0\}$ et $C_{10} = \{a > 0\}$, si C_1 est vérifiée alors C_{10} l'est aussi.

Étant donné un SADCI, nous appelons *état* du SADCI une valeur (vraie ou fausse) d'une partie ou totalité des conditions permettant de définir la valeur du système. Pour le système (S_5), un état est par exemple donné par $a > 0, b > 0, d > 0, e > 0$ et $f > 0$ vraies. Le système induit par cet état est

$$eq_1 : 0 = \dot{x}_2, \tag{6.7}$$

$$eq'_2 : 0 = 4x_2 + \dot{x}_1, \tag{6.8}$$

$$eq''_2 : 0 = x_1, \tag{6.9}$$

$$eq_3 : 0 = 6\dot{x}_1 + 2, \tag{6.10}$$

$$eq_4 : 0 = 6\dot{x}_5 + x_1 + 1. \tag{6.11}$$

On remarque que la valeur de la condition $c > 0$ n'est pas définie. Ceci s'explique par le fait que le système induit par l'état défini précédemment ne dépend pas de la valeur de cette condition. En effet, la valeur de la condition $c > 0$ n'a d'influence que si la condition $a > 0$ est fausse. Contrairement au cas non imbriqué, certains systèmes ne peuvent exister puisqu'une même condition peut apparaître dans plusieurs équations. Ainsi, le système

$$eq_1 : 0 = \dot{x}_2, \tag{6.12}$$

$$eq'_2 : 0 = 4x_2 + \dot{x}_1, \tag{6.13}$$

$$eq''_2 : 0 = x_3, \tag{6.14}$$

$$eq_3 : 0 = 3\dot{x}_2 + x_3 + 3 + x_4, \tag{6.15}$$

$$eq_4 : 0 = 6\dot{x}_5 + x_1 + 1. \tag{6.16}$$

n'est pas possible car l'équation (6.12) implique que la condition $a > 0$ soit vraie alors que l'équation (6.15) implique que l'équation $a > 0$ soit fausse.

Remarquons que les conditions qui définissent un état du SADCI sont compatibles car nous avons indiqué que nous considérons les conditions du système comme indépendantes.

Le problème de l'analyse structurelle associé aux systèmes algébro-différentiels conditionnels imbriqués, noté par la suite *Problème de l'Analyse Structurelle Conditionnelle Imbriquée* (PASCII), consiste alors à vérifier si les systèmes induits par tous les états du SADCI sont structurellement réguliers ou à trouver un état dont le système induit est structurellement singulier.

On remarque que les états d'un SADCI peuvent être définis avec des affectations du système. Il y a équivalence entre un état du système et un sous-ensemble $C^e \subseteq \mathcal{C}$ d'affectations du système compatibles et contenant une affectation d'équation pour toute EDCI du SADCI. En effet, considérons un état du système. L'ensemble C^e , obtenu en considérant l'ensemble des affectations de \mathcal{C} , vérifiant les valeurs donnés par l'état, contient uniquement des affectations compatibles. De plus, l'état définit pour chaque EDCI une unique valeur. Les valeurs des conditions donnant cette valeur correspondent à une affectation d'équation.

Considérons maintenant un sous-ensemble $C^e \subseteq \mathcal{C}$ d'affectations du système compatibles et contenant une affectation d'équation pour toute EDCI du SADCI. Comme les affectations de C^e sont compatibles, il n'existe pas de condition ayant pour valeur vrai et faux dans deux affectations de C^e . Il est donc possible, à partir de C^e , de définir des valeurs pour un ensemble de conditions du système. De plus, comme pour chaque EDCI du SADCI, il existe dans C^e une affectation de cette équation, la valeur de l'équation conditionnelle est donc définie, ce qui implique que l'ensemble de valeurs des conditions correspond à un état du système. À titre d'exemple, l'état donné précédemment pour le système (S_5) correspond au sous-ensemble d'affectations $\{C_3, C_9, C_{11}\}$.

6.2 Formulation du problème en termes de graphes

Dans cette section, nous formulons le problème de l'analyse structurelle des SADCI comme un problème de graphes. Une instance de ce dernier est alors définie par un graphe biparti $G = (U \cup V, E)$, un sous-ensemble U_c de sommets de U , une partition π de l'ensemble d'arête E et un graphe G_I représentant les incompatibilités entre les affectations des conditions du système. Cette instance est construite à partir d'un SADCI. Nous montrons ensuite que le PASCI est équivalent à rechercher un sous-graphe de $G = (U \cup V, E)$ sans couplage parfait et vérifiant certaines propriétés définies par U_c , π et G_I .

Nous présentons maintenant la construction du graphe biparti $G = (U \cup V, E)$, du sous-ensemble $U_c \in U$, de la partition π de E et du graphe G_I à partir d'un SADCI. Cette construction est similaire à celle donnée pour un SADC non imbriqué (cf Chapitre 3). Cependant, dans le cas des SADCI, le graphe G peut contenir des arêtes multiples. De plus, un sommet $u \in U$ peut être couvert par plus de 3 ensembles d'arêtes de π . Considérons un SADCI. À chaque variable du SADCI, on associe un sommet. On note par V l'ensemble de ces sommets. Pour chaque équation, on associe un sommet et on note par U cet ensemble. On note par U_c l'ensemble des sommets de U associés

aux équations conditionnelles imbriquées. Soit u un sommet quelconque de U . Si u est associé à une équation non-conditionnelle, alors pour tout $v \in V$ on ajoute dans E une arête uv si la variable associée au sommet v apparaît dans l'équation associée au sommet u . Si u est associé à une équation conditionnelle imbriquée, alors pour tout $v \in V$, on ajoute dans E une arête uv si la variable associée au sommet v apparaît dans toutes les équations générées par les différentes affectations d'équations de l'EDCI associées au sommet u . Notons E_u^{tf} l'ensemble d'arête ainsi obtenue. On note $E^{tf} = \cup_{u \in U} E_u^{tf}$. Pour toute affectation d'équations $C_i \in \mathcal{C}$, on ajoute dans E un ensemble d'arêtes E_i défini comme suit. Une arête $uv \in E_i$ pour tout $u \in U_c$ et pour tout $v \in V$, si $uv \notin E^{tf}$ et la variable associée au sommet v apparaît dans l'équation associée au sommet u quand l'affectation C_i est vérifiée. On remarque qu'entre un sommet $u \in U_c$ et un sommet $v \in V$, il peut exister dans G plusieurs arêtes multiples uv chacune appartenant à un ensemble $E_i, i = 1, \dots, m$ différent. Par la suite, on notera par \mathcal{E} l'ensemble des ensembles d'arêtes $E_i, i = 1, \dots, m$. On a alors $\mathcal{E} = \{E_i : i = 1, \dots, m\}$. La partition π de E est alors égale à $\pi = \mathcal{E} \cup \{E^{tf}\}$.

Nous représentons les incompatibilités entre les affectations du système à l'aide du graphe $G_I = (V_{\mathcal{E}}, E_T)$ défini comme suit. L'ensemble de sommets $V_{\mathcal{E}}$ est égal à $V_{\mathcal{E}} = \{v_{E_1}, \dots, v_{E_m}\}$. Chaque sommet $v_{E_i}, i = 1, \dots, m$ est associé à l'ensemble d'arêtes E_i . Pour i, j appartenant à $\{1, \dots, m\}, i \neq j$, l'arête $v_{E_i}v_{E_j}$ appartient à E_T si les affectations C_i et C_j sont incompatibles. On remarque qu'il existe une arête $v_{E_i}v_{E_j}$ si C_i et C_j sont associées à la même EDCI puisque dans ce cas, elles sont incompatibles.

Une instance du PASCII en termes de graphes est alors donnée par (G, U_c, π, G_I) où $G = (U \cup V, E)$ est un graphe biparti, $\pi = \mathcal{E} \cup \{E^{tf}\}$ est une partition de E et G_I est le graphe d'incompatibilités défini sur l'ensemble des éléments de \mathcal{E} .

Nous introduisons maintenant la notion de graphe associé à un sous-ensemble de l'ensemble \mathcal{E} . Étant donné $\mathcal{F} \subseteq \mathcal{E}$, on appelle graphe associé à \mathcal{F} le sous-graphe de G dont l'ensemble des sommets est $U \cup V$ et l'ensemble des arêtes est constitué des arêtes vraies/fausses de G ainsi que des arêtes appartenant à un élément de \mathcal{F} . Le graphe associé à $\mathcal{F} \subseteq \mathcal{E}$ sera par la suite noté $H_{\mathcal{F}} = (U \cup V, E_{\mathcal{F}})$, où $E_{\mathcal{F}} = E^{tf} \cup (\cup_{F \in \mathcal{F}} F)$.

Nous définissons maintenant le problème dit problème généralisé PSGSCP, correspondant à la formulation du PASCII en termes de graphes.

Définition 6.1 *Étant donné une instance (G, U_c, π, G_I) où $G = (U \cup V, E)$, $U_c \subseteq U$ et π est une partition de E , le problème généralisé PSGSCP (G -PSGSCP) consiste à déterminer s'il existe un sous-ensemble $\mathcal{F} \subseteq \mathcal{E}$ tel que*

- (i) *pour tout $v_{E_i}v_{E_j} \in G_I$, $E_i \notin \mathcal{F}$ et/ou $E_j \notin \mathcal{F}$,*

- (ii) le graphe $H_{\mathcal{F}}$ ne possède pas de couplage parfait,
- (iii) le nombre de sommets de U_c couverts par les arêtes des éléments de \mathcal{F} est maximum.

Il y a équivalence entre PASCI et le G-PSGSCP. De manière plus précise, Un SADCI est structurellement singulier si et seulement si une solution optimale \mathcal{F} de G-PSGSCP pour l'instance construite à partir du SADCI est telle que les arêtes des éléments de \mathcal{F} couvrent tous les sommets de U_c . En effet, supposons qu'il existe une solution \mathcal{F} satisfaisant les conditions (i)-(iii) telle que tous les sommets de U_c sont couverts par les arêtes des éléments de \mathcal{F} . Considérons alors le sous-ensemble d'affectations du système $C' \subset \mathcal{C}$ tel que C_j appartient à C' si et seulement si $E_j \in \mathcal{F}$, pour $j \in \{1, \dots, m\}$. Comme il existe dans le graphe G_I une arête uv entre deux sommets si les deux affectations associées à ces deux sommets sont incompatibles, C' contient uniquement des affectations compatibles puisque \mathcal{F} vérifie la condition (i). Comme chaque sommet u de U_c est couvert par les arêtes d'un élément $E_i \in \mathcal{F}$, il existe une affectation d'équation $C_i \in \mathcal{C}$ pour l'équation conditionnelle associée à u . D'après l'équivalence, donnée dans la section précédente, entre un sous-ensemble d'affectations compatibles contenant une affectation d'équation pour chaque EDCI du système et un état du SADCI, cela implique qu'il existe un état du SADCI tel que le graphe d'incidence du système induit par cet état correspond au graphe $H_{\mathcal{F}}$. Comme ce dernier ne contient pas de couplage, le système induit par cet état est structurellement singulier.

De manière similaire, supposons qu'il existe un état du SADCI dont le système induit est structurellement singulier. L'état implique qu'il existe un sous-ensemble C^e d'affectations compatibles contenant une affectation d'équation pour chaque EDCI du système. Considérons l'ensemble \mathcal{F} tel que $E_i \in \mathcal{F}$ si $C_i \in C^e$. Comme C^e contient une affectation d'équation pour chaque équation conditionnelle du SADCI, disons C_i , alors le sommet u associé à cette équation conditionnelle est couvert dans $H_{\mathcal{F}}$ par une arête de l'ensemble E_i associé à C_i . Comme le système induit par l'état est structurellement singulier, ceci implique que son graphe d'incidence, correspondant à $H_{\mathcal{F}}$, ne contient pas de couplage parfait. L'ensemble \mathcal{F} vérifie alors les propriétés (i)-(iii). De plus, tous les sommets de U_c sont couverts par des arêtes des éléments de \mathcal{F} .

Remarquons que les implications sont respectées. En effet, soient $E_j \in \mathcal{F}$ et un ensemble d'arêtes quelconque E_i , tel que $C_i \subset C_j$. Notons par \mathcal{C}_i les affectations d'équations, tel que leurs ensembles d'arêtes associés et l'ensemble d'arêtes E_i sont incidents au même sommet. Comme C_i est incompatible avec les éléments de \mathcal{C}_i , on en déduit que C_j l'est aussi. Comme la solution couvre un maximum de sommets de U_c , on en déduit que l'ensemble E_i appartient à la solution.

Pour le système (S_5) , les ensembles d'arêtes E_k correspondant aux affectations C_1, \dots, C_{11} sont $E_1 = \{u_1v_2\}$, $E_2 = \{u_1v_1\}$, $E_3 = \{u_1v_3, u_1v_2\}$, $E_4 = \{u_1v_4\}$, $E_5 = \{u_2v_2, u_2v_1\}$, $E_6 = \{u_2v_3\}$, $E_7 = \{u_3v_1\}$, $E_8 = \{u_3v_4, u_3v_3\}$, $E_9 = \{u_2v_1, u_2v_4, u_3v_4, u_3v_2\}$, $E_{10} = \{u_4v_1\}$ et $E_{11} = \{u_4v_2, u_4v_3, u_4v_4\}$. Nous avons aussi l'ensemble d'arêtes $E^{tf} = \{u_5v_1, u_5v_5\}$. Le graphe associé au système (S_5) est donné par la figure 6.1. Notons que les sommets u_1, u_2, u_3, u_4 et u_5 correspondent respectivement aux équations $eq_1, eq'_2, eq''_2, eq_3$ et eq_4 . Notons aussi que les sommets v_1, v_2, v_3, v_4 et v_5 correspondent respectivement aux variables x_1, x_2, x_3, x_4 et x_5 . Le nombre k sur chaque arête correspond à l'indice de l'ensemble E_k auquel appartient l'arête.

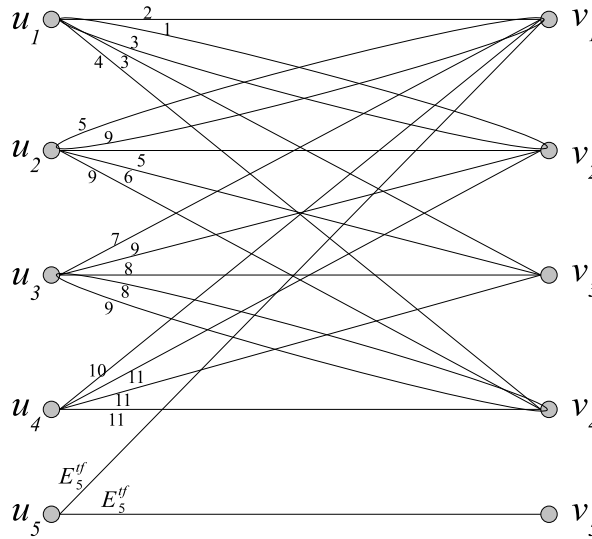


FIGURE 6.1 – Graphe G correspondant au système (S_5) .

Il est facile de voir que pour l'exemple donné par la figure 6.1, la solution $\mathcal{F} = \{E_1, E_6, E_7, E_{10}\}$ correspond à une solution du G-PSGSCP. Comme tous les sommets de U_c sont couverts par les arêtes des éléments de \mathcal{F} , l'état correspondant au sous-ensemble d'affectations $\{C_1, C_6, C_7, C_{10}\}$ induit un système structurellement singulier.

6.3 Formulation sous forme d'un programme linéaire en nombres entiers

Nous donnons ici une formulation mathématique du G-PSGSCP. Soit $(G, U_c, \mathcal{E}, G_I)$ une instance du G-PSGSCP. Soit $x \in \{0, 1\}^{\mathcal{E}}$ tel que

$$x_F = \begin{cases} 1 & \text{si } F \in \mathcal{F}, \\ 0 & \text{sinon,} \end{cases} \text{ pour tout } F \in \mathcal{E}.$$

Par abus de langage, on appelle graphe associé à x le graphe associé à l'ensemble de \mathcal{F} induit par x . La valeur $x_{E_i} = 1$ implique que l'affectation d'équations C_i est vérifiée et donc que l'ensemble d'arêtes E_i est présent dans le graphe associé à x .

La condition (i) de la définition 6.1 indique que si $v_{E_i}v_{E_j} \in G_I$, alors $E_i \notin \mathcal{F}$ et/ou $E_j \notin \mathcal{F}$. Nous en déduisons la contrainte

$$x_{E_i} + x_{E_j} \leq 1, \quad v_{E_i}v_{E_j} \in G_I.$$

Considérons un couplage parfait M de G . Soit $x \in \{0, 1\}^{\mathcal{E}}$ satisfaisant $x_F = 1$ si $M \cap F \neq \emptyset$ pour tout $F \in \mathcal{E}$. Alors le graphe associé à x contient un couplage parfait. Notons $F_M = \{F : F \in \mathcal{E} \text{ et } F \cap M \neq \emptyset\}$. Nous pouvons en déduire la contrainte de couplages suivante

$$\sum_{F \in F_M} x_F \leq |F_M| - 1.$$

Pour tout $F \in \mathcal{E}$, on note par w_F le nombre de sommets de U_c couverts par l'ensemble d'arêtes F . On a donc $w_F = |\{u \in U_c : \delta(u) \cap F \neq \emptyset\}|$. w_F correspond au nombre d'équations ayant comme état l'affectation d'équations associée à F . Ainsi, comme C_9 est une des affectations d'équations associées aux équations eq'_2 et eq''_2 , on en déduit que $w_{E_9} = 2$.

Le G-PSGSCP est alors équivalent au programme linéaire en nombres entiers (\widehat{P}) suivant.

$$\max \sum_{F \in \mathcal{E}} w_F x_F \quad (6.17)$$

$$\sum_{F \in F_M} x_F \leq |F_M| - 1, \quad \text{pour tout } M \in \mathcal{M}, \quad (6.18)$$

$$x_{E_i} + x_{E_j} \leq 1, \quad \text{pour tout } v_{E_i} v_{E_j} \in G_I, \quad (6.19)$$

$$x_F \geq 0, \quad \text{pour tout } F \in \mathcal{E}, \quad (6.20)$$

$$x_F \leq 1, \quad \text{pour tout } F \in \mathcal{E}, \quad (6.21)$$

$$x_F \text{ entiers}, \quad \text{pour tout } F \in \mathcal{E}, \quad (6.22)$$

où \mathcal{M} est l'ensemble des couplages parfaits.

Les contraintes (6.18) sont les *contraintes de couplages*. Les inégalités (6.19) sont les *contraintes d'incompatibilités*. Les contraintes (6.22) sont les *contraintes d'intégrité*.

Les conditions (i) et (ii) de la définition 6.1 sont vérifiées grâce aux contraintes (6.18) et (6.19). Pour montrer que le G-PSGSCP est équivalent au programme (\hat{P}) il nous suffit de montrer que la condition (iii) est respectée. Pour cela, il suffit de voir que la valeur de la fonction objectif correspond au nombre de sommets de U_c couverts par des arêtes des éléments de \mathcal{F} . En effet, par définition, deux affectations d'équation d'une même EDCI sont incompatibles, ce qui implique que deux ensembles de \mathcal{E} ne peuvent pas couvrir un même sommet de U_c . La définition des poids w_F , $F \in \mathcal{E}$, implique la correspondance.

Un SADCI est structurellement singulier si la solution optimale x^* possède un poids de $|U_c|$, sinon le système est structurellement régulier.

La formulation (\hat{P}) est une généralisation de la formulation (P) du chapitre 3. Remarquons que les contraintes de satisfiabilités de la formulation (P) sont un cas particulier des contraintes (6.19). Elles correspondent au cas particulier où les arêtes du graphe G_I forment un couplage parfait, les extrémités de chaque arête $v_{E_i} v_{E_j}$ du couplage parfait correspondant aux sommets E_u^t et E_u^f pour tout $u \in U_c$. Finalement, dans la formulation (P) , les coefficients de la fonction objectif sont tous égaux à un car les ensembles d'arêtes appartenant à \mathcal{E} ne couvrent qu'un sommet de U_c .

Soit $P(G, U_c, \pi, G_I)$ l'enveloppe convexe des solutions du programme (\hat{P}) , c'est-à-dire,

$$P(G, U_c, \pi, G_I) = \text{conv}(\{x \in \{0, 1\}^{\mathcal{E}} \mid x \text{ satisfait (6.18), (6.19)}\}).$$

Nous allons étudier dans la prochaine section l'enveloppe convexe $P(G, U_c, \pi, G_I)$.

6.4 Étude polyédrale

Dans cette section, on étudie l'enveloppe convexe des solutions du G-PSGSCP. On caractérise la dimension du polytope des solutions. On donne, par la suite, une extension des contraintes que nous avons présentées dans le chapitre 4 et on étudie également leur aspect facial.

Dans la suite du travail, on considérera que l'instance (G, U_c, π, G_I) vérifie l'hypothèse suivante

Hypothèse 6.2 *Les graphes associés à $\{F\}$ avec $F \in \mathcal{E}$, ne contiennent pas de couplage parfait.*

En effet, soit $F \in \mathcal{E}$ si le graphe associé à $\{F\}$ contient un couplage parfait alors il ne peut pas exister de solution $\mathcal{F} \subseteq \mathcal{E}$ contenant F car le graphe associé à \mathcal{F} contient lui aussi un couplage parfait. Nous pouvons donc transformer cette instance en supprimant F . Il est clair que ces transformations peuvent être faites en temps polynomial.

Étant donné une clique K de G_I , on note par \mathcal{E}_K les éléments de \mathcal{E} associés aux sommets de la clique K , c'est-à-dire $\mathcal{E}_K = \{F \in \mathcal{E} : v_F \in K\}$.

6.4.1 Dimension

Dans cette section, on caractérise la dimension de $P(G, U_c, \pi, G_I)$.

Théorème 6.3 *$P(G, U_c, \pi, G_I)$ est de pleine dimension.*

Preuve. Il suffit d'exhiber $|\mathcal{E}| + 1$ sous-ensembles de \mathcal{E} vérifiant les conditions (i) et (ii) de la définition 6.1 et dont les vecteurs d'incidence sont affinement indépendants. Considérons les ensembles $\mathcal{F}_0 = \emptyset$ et $\mathcal{F}_F = \{F\}$ pour tout $F \in \mathcal{E}$. Les ensembles contiennent au plus un élément de \mathcal{E} , ce qui implique que la condition (i) est vérifiée. Par hypothèse, les graphes associés à ces ensembles ne contiennent pas de couplages parfaits. Cela implique que les ensembles correspondent à des solutions du G-PSGSCP. De plus, il est clair que les vecteurs d'incidences de ces solutions sont affinement indépendants. \square

6.4.2 Inégalités valides et facettes

Nous allons voir dans cette section des contraintes valides pour le $P(G, U_c, \pi, G_I)$. Certaines de ces contraintes sont des extensions des contraintes présentées dans le chapitre 4.

6.4.2.1 Inégalités triviales

On remarque que les inégalités triviales (6.21) sont dominées par les contraintes de d'incompatibilités (6.19). En effet, pour tout $F \in \mathcal{E}$, il existe au moins un élément de \mathcal{E} incompatible avec F car une équation conditionnelle est associée à plus d'une affectation. On montre maintenant que les inégalités (6.20) définissent des facettes.

Théorème 6.4 *Les inégalités triviales (6.20) définissent des facettes de $P(G, U_c, \pi)$.*

Preuve. Considérons l'inégalité (6.20) associée à l'élément $F' \in \mathcal{E}$. Comme le polytope $P(G, U_c, \pi, G_I)$ est de pleine dimension, il suffit d'exhiber $|\mathcal{E}|$ sous-ensembles de \mathcal{E} vérifiant les conditions (i) et (ii) de la définition 6.1 et dont les vecteurs d'incidence sont affinement indépendants. Considérons les ensembles $\mathcal{F}_0 = \emptyset$ et $\mathcal{F}_F = \{F\}$ pour tout $F \in \mathcal{E} \setminus \{F'\}$. Ces ensembles contiennent au plus un élément de \mathcal{E} ce qui implique que la condition (i) est vérifiée. Par hypothèse, les graphes associés à ces ensembles ne contiennent pas de couplages parfaits. Nous avons donc $|\mathcal{E}|$ solutions associées aux ensembles \mathcal{F}_0 et \mathcal{F}_F , $F \in \mathcal{E} \setminus \{F'\}$, telles que leurs vecteurs d'incidence sont affinement indépendants. \square

6.4.2.2 Inégalités de cliques

Les contraintes d'incompatibilités (6.19) indiquent que les sommets associés aux éléments de \mathcal{F} forment un stable dans le graphe G_I . Les contraintes valides pour le stable [66] sont donc valides pour le G-PSGSCP pour le graphe G_I . On peut donc, par exemple, considérer les contraintes associées aux cliques du graphe G_I

$$\sum_{F \in \mathcal{E}_K} x_F \leq 1, \quad \text{pour tout clique } K \in G_I. \quad (6.23)$$

Les contraintes (6.23) sont appelées *contraintes de cliques*. Nous donnons dans le théorème suivant les conditions nécessaires et suffisantes pour que les contraintes de cliques définissent des facettes.

Théorème 6.5 *Soit K une clique de G_I . L'inégalité (6.23) associée à K définit une facette de $P(G, U_c, \pi, G_I)$ si et seulement si pour tout $F \in \mathcal{E} \setminus \mathcal{E}_K$, il existe un $F' \in \mathcal{E}_K$ compatible avec F tel que le graphe associé à $\{F, F'\}$ ne contient pas de couplage parfait.*

Preuve. (\Rightarrow) Si la condition n'est pas respectée alors il existe un élément de $\mathcal{E} \setminus \mathcal{E}_K$, disons F_1 , tel que pour tout $F^* \in \mathcal{E}_K$ compatible avec F_1 , il existe un couplage parfait dans le graphe associé à $\{F_1, F^*\}$. Ceci implique que les contraintes suivantes sont valides :

$$x_F + x_{F_1} \leq 1, \quad \text{pour tout } F \in \mathcal{E}_K. \quad (6.24)$$

En effet, étant donné $F \in \mathcal{E}_K$ si F est incompatible avec F_1 , l'inégalité (6.24) associée à F n'est rien d'autre que la contrainte d'incompatibilités (6.19) associée à F_1 et F . Si F est compatible avec F_1 , alors, le graphe associé à $\{F, F_1\}$ contient un couplage parfait, disons M . D'après l'hypothèse 6.2, on déduit que $F_M = \{F, F_1\}$. L'inégalité (6.24) associée à F n'est dans ce cas rien d'autre que la contrainte de couplages (6.18) associée à M .

En additionnant les inégalités (6.24) pour tout $F \in \mathcal{E}_K$ et $|K| - 1$ fois la contrainte de cliques (6.23) associée à K puis en divisant par $|K|$ et en arrondissant le second membre à l'entier inférieur, nous obtenons la contrainte

$$\sum_{F \in \mathcal{E}_K} x_F + x_{F'} \leq 1. \quad (6.25)$$

Ceci implique que la contrainte de cliques (6.23) associée à K est dominée par la contrainte (6.25). Comme cette dernière définit une face propre, la contrainte de cliques (6.23) associée à K ne définit donc pas une facette.

(\Leftarrow) Supposons maintenant que la condition est vérifiée. Notons par $ax \leq \alpha$ la contrainte (6.23) associée à K . Soit $bx \leq \beta$ une inégalité valide définissant une facette de $P(G, U_c, \pi, G_I)$. Supposons que $\{x \in P(G, U_c, \pi, G_I) : ax = \alpha\} \subseteq \{x \in P(G, U_c, \pi, G_I) : bx = \beta\}$. Puisque $P(G, U_c, \pi, G_I)$ est de pleine dimension, il suffit de montrer qu'il existe $\rho > 0$ tel que $b = \rho a$.

Soient F_1 et F_2 deux éléments distincts de \mathcal{E}_K . D'après l'hypothèse 6.2, les ensembles $\mathcal{F}_1 = \{F_1\}$, et $\mathcal{F}_2 = \{F_2\}$ vérifient les conditions (i) et (ii) et sont donc solutions. De plus, les vecteurs d'incidence vérifient à l'égalité la contrainte $ax \leq \alpha$, ce qui implique que l'on a $bx^{\mathcal{F}_1} = bx^{\mathcal{F}_2}$. On en déduit alors que $b(F_1) = b(F_2)$. Comme F_1 et F_2 sont des éléments quelconques de \mathcal{E}_K , il s'ensuit que

$$b(F') = b(F), \quad \text{pour tout } F \neq F' \in \mathcal{E}_K. \quad (6.26)$$

Soit F_3 un élément quelconque de $\mathcal{E} \setminus \mathcal{E}_K$. D'après la condition du théorème, il existe un élément de \mathcal{E}_K , compatible avec F_3 , disons F_4 , tel que le graphe $H_{\mathcal{F}_3}$ associé à $\mathcal{F}_3 = \{F_3, F_4\}$ ne contient pas de couplage parfait. Il s'ensuit alors que \mathcal{F}_3 est une solution du G-PSGSCP. De plus, par hypothèse, l'ensemble $\mathcal{F}_4 = \{F_4\}$ est tel que le graphe $H_{\mathcal{F}_4}$ est sans couplage parfait, ce qui implique que \mathcal{F}_4 est aussi une solution du G-PSGSCP. De plus, les vecteurs d'incidence de \mathcal{F}_3 et \mathcal{F}_4 vérifient $ax = \alpha$, ce qui implique qu'ils vérifient aussi $bx^{\mathcal{F}_3} = bx^{\mathcal{F}_4}$. On déduit alors que $b(F_3) = 0$. Comme F_3 est un élément quelconque de $\mathcal{E} \setminus \mathcal{E}_K$, on obtient alors

$$b(F) = 0, \quad \text{pour tout } F \in \mathcal{E} \setminus \mathcal{E}_K. \quad (6.27)$$

Posons $\rho = b(F_1)$. On en déduit que $b(F) = \rho$ pour tout $F \in \mathcal{E}_K$. Par l'équation (6.27), il s'ensuit que $b = \rho a$. De plus, comme pour tout ensemble d'arêtes $F \in \mathcal{E}_K$, il existe un sous-graphe H contenant F tel que la contrainte soit vérifiée à l'égalité, la face définie par $ax \leq \alpha$ est différente d'une face triviale $\{x \in P(G, U_c, \pi, G_I) : x_F = 0\}$. Cela implique que la face induite par $bx \leq \beta$ n'est pas contenue dans une face triviale. Il s'ensuit que $b(F) \geq 0$ pour tout $F \in \mathcal{E}_K$. Comme $bx \leq \beta$ définit une facette de $P(G, U_c, \pi, G_I)$, il doit donc exister au moins un ensemble d'arêtes $F \in \mathcal{E}_K$ tel que $b(F) > 0$. Par conséquent, $\rho > 0$ et la preuve est terminée. \square

Remarquons que la condition du théorème implique que la clique K est maximale.

Nous allons voir, dans la section suivante, l'extension des contraintes de couplages proches (4.19).

6.4.3 Inégalités de couplages-cliques

Nous discutons ici de l'extension des contraintes de couplages proches (4.19).

Un couplage parfait M de G est dit à *dominance maximale* s'il n'existe pas un couplage parfait M' de G tel que $F_{M'} \subset F_M$ et si le sous-ensemble $\{v_F : F \in F_M\} \subset V_{\mathcal{E}}$ est un stable de G_I .

Soient (G, U_c, π, G_I) une instance du G-PSGSCP, M un couplage parfait à dominance maximale de G et K une clique dans G_I . On dit que le couplage M est *couplages-cliques* par rapport à K si pour tout élément $F \in \mathcal{E}_K$, au moins une des trois conditions est vérifiée :

- i) F est incompatible avec deux éléments distincts de F_M ,
- ii) F est incompatible avec un élément de F_M , disons F' , et le graphe $H_{F_M \cup \{F\} \setminus \{F'\}}$ contient un couplage parfait.
- iii) pour tout $F' \in F_M$ les graphes $H_{F_M \cup \{F\} \setminus \{F'\}}$ contiennent un couplage parfait.

D'après la définition des couplages à dominance maximale, on déduit que $F_M \cap \mathcal{E}_K = \emptyset$. En effet, si un élément F appartient à $F_M \cap \mathcal{E}_K$ alors F ne vérifie ni i) ni ii) puisque les sommets de G_I associées aux éléments de F_M forment un stable. De plus, comme il n'existe pas de couplage parfait M' tel que $F_{M'} \subset F_M$, F ne vérifie pas iii), ce qui implique que M n'est pas un couplage clique par rapport à K .

Proposition 6.6 *Soit M un couplage parfait à dominance maximale de G et K une clique dans G_I . Si le couplage M est couplages-cliques par rapport à K , alors la contrainte associée à M et K , donnée par*

$$\sum_{F \in \mathcal{E}_K} x_F + \sum_{F \in F_M} x_F \leq |F_M| - 1, \quad (6.28)$$

est valide pour le G-PSGSCP.

Preuve. Dans un premier temps, on montre que pour tout $F' \in \mathcal{E}_K$, la contrainte associée à F_M et F' , donnée par

$$\sum_{F \in F_M} x_F + x_{F'} \leq |F_M| - 1. \quad (6.29)$$

est valide si F' vérifie i), ii) ou iii).

En effet, supposons que F' vérifie i). Si F' est incompatible avec deux éléments de F_M , disons F_1 et F_2 , alors les contraintes d'incompatibilités associées aux arêtes $v_{F_1}v_{F'}$ et $v_{F_2}v_{F'}$ de G_I sont valides. En additionnant ces contraintes, la contrainte de couplages (6.18) associée à M et les contraintes triviales (6.21) associées aux éléments de $F_M \setminus \{F_1, F_2\}$ on obtient alors la contrainte :

$$2 \sum_{F \in F_M} x_F + 2x_{F'} \leq 2|F_M| - 1. \quad (6.30)$$

En divisant cette contrainte par deux et en prenant la partie entière inférieure du membre de droite, on obtient la contrainte (6.29).

Supposons que F' vérifie *ii*). Dans ce cas, F' est incompatible avec un élément de F_M , disons F'' , et le graphe associé à $\mathcal{F}_1 = F_M \cup \{F'\} \setminus \{F''\}$ contient un couplage parfait, disons N . En additionnant les contraintes de couplages (6.18) associées à M et N , les contraintes triviales associées aux éléments $\mathcal{F}_1 \setminus F_N$ et la contrainte d'incompatibilités associée à l'arête $v'_F v''_F$ de G_I puis en divisant par deux et en prenant la partie entière inférieure du membre de droite, on obtient la contrainte (6.29).

Supposons que F' vérifie *iii*). Dans ce cas, il existe F'' tel que pour tout $F'' \in F_M$ les graphes $H_{F_M \cup \{F'\} \setminus \{F''\}}$ contiennent un couplage parfait. Les contraintes de couplages

$$\sum_{F \in F_M \setminus F''} x_F + x_{F'} \leq |F_M| - 1, \quad \text{pour tout } F'' \in F_M \quad (6.31)$$

$$\sum_{F \in F_M} x_F \leq |F_M| - 1, \quad (6.32)$$

sont donc valides. En additionnant ces inégalités et en les divisant par $|F_M|$ et en arrondissant le second membre à l'entier inférieur, nous obtenons la contrainte (6.29). Nous venons de montrer que les contraintes (6.29) sont valides pour tout $F \in \mathcal{E}_K$. En additionnant les contraintes (6.29) associées à F_M et F' , avec $F' \in \mathcal{E}_K$, et $|\mathcal{E}_K| - 1$ fois la contrainte de cliques (6.23) associée à K puis en divisant par $|\mathcal{E}_K|$ et en prenant la partie entière inférieure du membre de droite, on obtient la contrainte (6.28) associée à M et K . \square

Nous appelons les contraintes (6.28) les *contraintes de couplages-cliques*.

Remarquons que les contraintes de couplages proches (4.19) sont un cas particulier des contraintes de couplages-cliques. En effet, dans le cas non-imbriqué, le graphe G_I est composé uniquement d'un couplage parfait ce qui implique que pour un couplage parfait M de G si on considère un élément $F \in F_M$, il n'existe qu'un seul élément incompatible dans \mathcal{E} noté dans le chapitre 4 par \bar{F} . On en déduit que les contraintes de couplages proches sont définies par une clique de cardinalité 2 et vérifiant la deuxième condition des contraintes de couplages-cliques.

Nous allons voir dans la section suivante l'extension des contraintes de couplages à dominance maximale (4.13).

6.4.3.1 Inégalités de couplages à dominance maximale

Nous discutons ici des conditions nécessaires et suffisantes pour que les contraintes de couplages (6.18) définissent des facettes de $P(G, U_c, \pi, G_I)$.

Rappelons qu'un couplage parfait M de G est dit à dominance maximale s'il n'existe pas un couplage parfait M' de G tel que $F_{M'} \subset F_M$ et si le sous-ensemble $\{v_F : F \in F_M\} \subset V_{\mathcal{E}}$ est un stable de G_I .

Notons \mathcal{M}_D l'ensemble des couplages à dominance maximale. On appelle *contrainte de couplages à dominance maximale* une inégalité de couplages (6.18) associée à un couplage à dominance maximale de \mathcal{M}_D . Les contraintes de couplages à dominance maximale s'écrivent alors

$$\sum_{F \in F_M} x_F \leq |F_M| - 1, \quad \text{pour tout } M \in \mathcal{M}_D. \quad (6.33)$$

Proposition 6.7 *Soit M un couplage parfait de G n'appartenant pas à \mathcal{M}_D . La contrainte (6.18) associée à M est redondante.*

Preuve. S'il existe un couplage parfait M' tel que $F_{M'} \subset F_M$ alors la contrainte (6.18) associée à M est combinaison linéaire de la contrainte (6.18) associée à M' et des contraintes triviales $x_F \leq 1$ pour tout $F \in F_M \setminus F_{M'}$.

Par ailleurs, s'il existe $F', F'' \in F_M$ tel que $v_{F'}v_{F''} \in E_I$, alors la contrainte (6.18) associée à M est combinaison linéaire de la contrainte $x_{F'} + x_{F''} \leq 1$ et des contraintes triviales $x_F \leq 1$ pour tout $F \in F_M \setminus \{F', F''\}$. \square

Nous allons donner, dans la suite, les conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes.

Théorème 6.8 *Soit M un couplage parfait à dominance maximale. L'inégalité (6.18) associée à M définit une facette de $P(G, U_c, \pi, G_I)$ si et seulement si pour tout F tel que $F \in \mathcal{E} \setminus F_M$, le couplage parfait M n'est pas un couplage-clique par rapport à $K = \{v_F\}$.*

Preuve. (\Rightarrow) Si la condition du théorème n'est pas respectée alors comme K n'est pas vide il est clair que la contrainte de couplages-cliques (6.28) associée à M et K domine la contrainte de couplages (6.18).

(\Leftarrow) Supposons maintenant que la condition est vérifiée. Notons par $ax \leq \alpha$ la contrainte (6.18) associée à M . Soit $bx \leq \beta$ une contrainte qui définit une facette de

$P(G, U_c, \pi, G_I)$. Supposons que $\{x \in P(G, U_c, \pi, G_I) : ax = \alpha\} \subseteq \{x \in P(G, U_c, \pi, G_I) : bx = \beta\}$. Puisque $P(G, U_c, \pi, G_I)$ est de pleine dimension, il suffit de montrer qu'il existe $\rho > 0$ tel que $b = \rho a$.

Soient F_1 et F_2 deux éléments distincts de F_M . Considérons les ensembles $\mathcal{F}_1 = F_M \setminus \{F_1\}$ et $\mathcal{F}_2 = F_M \setminus \{F_2\}$. Comme le couplage parfait M est à dominance maximale les graphes associés à \mathcal{F}_1 et \mathcal{F}_2 ne contiennent pas de couplage parfait. Il est clair que \mathcal{F}_1 et \mathcal{F}_2 sont des solutions du G-PSGSCP. Comme les vecteurs d'incidence de \mathcal{F}_1 et \mathcal{F}_2 vérifient $ax = \alpha$, il s'ensuit que $bx^{\mathcal{F}_1} = bx^{\mathcal{F}_2}$. On en déduit que $b(F_1) = b(F_2)$. Comme F_1 et F_2 sont deux éléments quelconques de F_M , on déduit que

$$b(F') = b(F^*) = \rho, \quad \text{pour tout } F', F^* \in F_M, \quad (6.34)$$

où ρ est un scalaire dans \mathbb{R}

Considérons un élément de $\mathcal{E} \setminus F_M$, disons F_3 . D'après la condition du théorème M n'est pas couplage-clique par rapport à $\{v_{F_3}\}$. D'après la définition des couplages-cliques, il existe F_4 appartenant à F_M , tel que le graphe associé à $\mathcal{F}_3 = (F_M \setminus \{F_4\}) \cup \{F_3\}$ ne contient pas de couplage parfait et F_3 est compatible avec tous les éléments de $F_M \setminus \{F_4\}$, ce qui implique que \mathcal{F}_3 vérifie la condition (i) de la définition 6.1. \mathcal{F} est donc une solution. Par ailleurs, comme M est à dominance maximale, le graphe associé à \mathcal{F}_4 ne contient pas de couplage parfait et \mathcal{F}_4 est composé d'éléments compatibles. Comme les vecteurs d'incidence de \mathcal{F}_3 et \mathcal{F}_4 vérifient $ax = \alpha$, il s'ensuit que $bx^{\mathcal{F}_3} = bx^{\mathcal{F}_4}$, ce qui implique que $b(F_3) + b(F_4) = b(F_4)$. Il s'ensuit que $b(F_3) = 0$. Comme F_3 est un élément quelconque de $\mathcal{E} \setminus F_M$, on déduit que

$$b(F) = 0, \quad \text{pour tout } F \in \mathcal{E} \setminus F_M. \quad (6.35)$$

Par (6.34)-(6.35), il s'ensuit que $b = \rho a$. De plus, comme pour tout ensemble d'arêtes $F \in F_M$, il existe un sous-graphe H contenant F tel que la contrainte soit vérifiée à l'égalité, la face définie par $ax \leq \alpha$ est différente d'une face triviale $\{x \in P(G, U_c, \pi, G_I) : x_F = 0\}$. Cela implique que la face induite par $bx \leq \beta$ n'est pas contenue dans une face triviale. Il s'ensuit que $b(F) \geq 0$ pour tout $F \in F_M$. Comme $bx \leq \beta$ définit une facette de $P(G, U_c, \pi, G_I)$, il doit donc exister au moins un ensemble d'arêtes $F \in F_M$ tel que $b(F) > 0$. Par conséquent, $\rho > 0$ et la preuve est terminée. \square

6.4.4 Inégalités de k -couplages

Nous allons étendre les inégalités de k -couplages (4.32) pour le problème du G-PSGSCP. Dans ce qui suit, on note par Θ l'ensemble des sous-ensembles θ de \mathcal{E} différents

de l'ensemble vide tel que tous les éléments de θ sont compatibles. Autrement dit, un sous-ensemble non-vide $\theta \subseteq \mathcal{E}$ appartient à Θ s'il n'existe pas deux éléments de θ , disons F' et F^* , tel que $v_{F'}v_{F^*} \in G_I$. Soient $\theta \in \Theta$, $\theta' \subseteq \theta$ et k un entier appartenant à $\{1, \dots, |\theta'| + 1\}$. Le graphe H_θ est dit *k-couplages* par rapport à θ' si après la suppression de n'importe quels $k - 1$ ensembles d'arêtes de θ' dans H_θ , le graphe résultant contient toujours un couplage parfait.

Proposition 6.9 *Soient $\theta \in \Theta$, $\theta' \subseteq \theta$ et $k \in \{1, \dots, |\theta'| + 1\}$. Si le graphe H_θ est k-couplages par rapport à θ' alors la contrainte associée à θ' , θ et k , donnée par*

$$\sum_{F \in \theta'} x_F + k \sum_{F \in \theta \setminus \theta'} x_F \leq |\theta'| + k(|\theta \setminus \theta'| - 1), \quad (6.36)$$

est valide pour $P(G, U_c, \pi, G_I)$.

Preuve. La preuve est similaire à celle de la proposition 4.9. □

La contrainte (6.36) est appelée *contrainte de k-couplages*. Remarquons que la contrainte possède exactement la même forme que dans le cas du PSGSCP. Les propositions 4.10 et 4.11 restent valides dans le cas du G-PSGSCP.

6.5 Conclusion

Dans ce chapitre, nous avons d'abord présenté les systèmes algébro-différentiels conditionnels imbriqués, correspondant à une généralisation des systèmes algébro-différentiels conditionnels lorsque plusieurs conditions peuvent être associées à une équation conditionnelle et une condition peut être associée à plusieurs équations. De plus, des relations logiques entre les conditions sont prises en compte. Nous avons ensuite présenté une formulation du problème de l'analyse structurelle de ces systèmes en termes de graphes. Nous avons également formulé ce problème à l'aide d'un programme linéaire en nombres entiers. Nous avons ensuite exhibé les relations entre cette nouvelle formulation et celle donnée pour le cas des systèmes algébro-différentiels conditionnels non-imbriqués. Finalement, nous avons défini de nouvelles inégalités valides pour ce problème, certaines d'entre elles étant des généralisations de contraintes valides introduites dans le chapitre 4 pour le cas non-imbriqué. Nous avons également donné des conditions nécessaires et des conditions suffisantes pour que certaines de ces contraintes définissent des facettes. L'une des perspectives serait de développer un algorithme de coupes et branchements permettant de résoudre efficacement le problème de l'analyse structurelle pour les systèmes algébro-différentiels conditionnels imbriqués.

Chapitre 7

Décomposition des systèmes algébro-différentiels et partitionnement de graphes

La résolution numérique des systèmes algébro-différentiels se fait à l'aide d'algorithmes de simulation. Il est alors très intéressant de pouvoir paralléliser la résolution de ces systèmes, afin de les résoudre plus rapidement. La méthode du complément de Schur permet la parallélisation de la résolution de ces systèmes sur plusieurs processeurs. Cette méthode permet de décomposer la résolution si la matrice Jacobienne est sous la forme appelée matrice bloc diagonale bordée. Cette matrice représente la structure du système et fournit les informations suffisantes pour sa résolution. Nous nous intéressons dans ce chapitre au problème de réécriture de la matrice Jacobienne sous la forme bloc diagonale bordée. Nous allons voir que ce problème se ramène à un problème de partitionnement dans le graphe associé à la matrice Jacobienne.

Les problèmes de partitionnement de graphes sont rencontrés dans plusieurs domaines : la conception de circuits intégrés électroniques, la répartition de charge pour les machines parallèles, la dynamique des fluides, le calcul matriciel, etc.

Le problème de partitionnement de graphe consiste à partitionner les sommets ou les arêtes du graphe pour optimiser une certaine fonction. En général, le partitionnement de graphe s'effectue sur les sommets et se présente de la manière suivante. Étant donné un graphe $G = (V, E)$ et un entier $k \geq 2$, le problème consiste à déterminer une partition V_1, \dots, V_k de V qui satisfait certaines contraintes et optimise une certaine fonction. On rappelle que V_1, \dots, V_k est une partition de V si $V_i \cap V_j = \emptyset, \forall i \neq j \in \{1, \dots, k\}$, et $V = V_1 \cup \dots \cup V_k$. Cette définition du problème est très générale, il existe

plusieurs variantes de ce problème, comme par exemple, le problème de la k -coupe [56, 78], le problème k -way [43, 46, 47] ou le problème du séparateur [14, 19, 33].

Dans ce chapitre, nous étudions le problème du séparateur. Dans la première section, nous donnons les définitions des matrices Jacobiennes et des matrices blocs diagonales bordées. Dans la deuxième section, nous montrons le lien entre le problème du séparateur et les matrices Jacobiennes. La section 7.3 donne une vue d'ensemble du problème du séparateur. Dans la section 7.4 nous donnons plusieurs formulations sous la forme de programmes linéaires en nombres entiers. Nous discutons dans la section 7.5 de certains résultats liés au polyèdre des solutions du problème. Et nous analysons enfin quelques résultats expérimentaux.

7.1 Matrice Jacobienne et résolution parallèle

La résolution numérique d'un SAD peut être effectuée à l'aide de la méthode du complément de Schur. Celle-ci permet une résolution en parallèle sur plusieurs processeurs. Pour ce faire, il faut dans un premier temps réécrire la matrice Jacobienne du système sous la forme bloc diagonale bordée. La matrice Jacobienne est la matrice des dérivées partielles du premier ordre d'une fonction vectorielle. Soit F une fonction de \mathbb{R}^n dans \mathbb{R}^m :

$$F : \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \rightarrow \begin{pmatrix} f_1(x_1, \dots, x_n) \\ \vdots \\ f_m(x_1, \dots, x_n) \end{pmatrix}$$

La *matrice Jacobienne* de F , notée $J(F)$, de m lignes et n colonnes, est donnée par :

$$J(F) = \frac{\partial F}{\partial x} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Nous allons voir à quoi correspond la matrice Jacobienne pour les systèmes différentiels ordinaires (SDO) et les systèmes algébro-différentiels. La matrice Jacobienne possède les informations suffisantes pour la résolution de ces systèmes. Rappelons que la forme générale des SDO est donnée par

$$\dot{y} = f(y, t),$$

et la forme générale des SAD est donnée par

$$f(\dot{y}, y, t) = 0.$$

Les matrices Jacobiennes associées aux SDO et SAD sont données respectivement par

$$J_{SDO} = I - \gamma \frac{\partial f}{\partial y},$$

$$J_{SAD} = \frac{\partial f}{\partial y} + \alpha \frac{\partial f}{\partial \dot{y}},$$

où γ et α sont des paramètres qui dépendent du pas de temps et de la méthode de résolution choisie et I correspond à la matrice identité. Pour la matrice Jacobienne J_{SDO} , la matrice identité correspond aux variables \dot{y} . Dans les méthodes de résolution numériques, les variables y et \dot{y} sont liées mais ne jouent pas le même rôle. On considère donc des matrices Jacobiennes qui dépendent de y et \dot{y} . Dans la suite, en fonction du système considéré, la matrice Jacobienne J_{SDO} ou J_{SAD} sera notée J .

Remarquons que la matrice Jacobienne J contient des informations similaires à la matrice $(E - \lambda A)$ décrite dans le chapitre 2.

On suppose dans la suite du chapitre que les systèmes sont carrés, i.e. contiennent autant de variables que d'équations. Si le système n'est pas carré alors, comme nous l'avons vu dans le chapitre 2, il est impossible de le résoudre avec des méthodes numériques.

Considérons par exemple le SDO (S_6) suivant :

$$eq_1 : \quad \dot{x}_1 = 2x_1 + 2x_2 + 4x_3 + 2, \quad (7.1)$$

$$eq_2 : \quad \dot{x}_2 = 6x_2 + 2, \quad (7.2)$$

$$eq_3 : \quad \dot{x}_3 = 6x_1 + x_2 + 2, \quad (7.3)$$

$$eq_4 : \quad \dot{x}_4 = x_1 + x_4 + 6. \quad (7.4)$$

La matrice Jacobienne associée à (S_6) est donnée par :

$$J = \begin{pmatrix} 1 - 2\gamma & -2\gamma & -4\gamma & 0 \\ 0 & 1 - 6\gamma & 0 & 0 \\ -6\gamma & -\gamma & 1 & 0 \\ -\gamma & 0 & 0 & 1 - \gamma \end{pmatrix}.$$

Considérons aussi par exemple le SAD (S_7) suivant :

$$eq_1 : \quad x_1 + 2x_2 + 4x_3 + 2 = 0, \quad (7.5)$$

$$eq_2 : \quad \dot{x}_2 + 6x_2 + 2 = 0, \quad (7.6)$$

$$eq_3 : \quad x_3 + 6x_1 + x_2 + 2 = 0, \quad (7.7)$$

$$eq_4 : \quad \dot{x}_4 + x_1 + \dot{x}_3 + 6 = 0. \quad (7.8)$$

La matrice Jacobienne associée à (S_7) est donnée par :

$$J = \begin{pmatrix} 1 & 2 & 4 & 0 \\ 0 & 6 + \alpha & 0 & 0 \\ 6 & 1 & 1 & 0 \\ 1 & 0 & \alpha & \alpha \end{pmatrix}.$$

La méthode du complément de Schur [40] consiste à répartir la résolution du système sur k processeurs, pour cela la matrice Jacobienne doit être réécrite sous la forme appelée *bloc diagonale bordée*, c'est-à-dire, sous la forme suivante.

$$J = \begin{pmatrix} A_1 & 0 & \dots & 0 & F_1 \\ 0 & A_2 & \dots & 0 & F_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & A_k & F_k \\ E_1 & E_2 & \dots & E_k & F_C \end{pmatrix}.$$

Les colonnes F_i correspondent aux variables d'interfaces.

La résolution d'un système différentiel se fait par approximations successives des points de la fonction correspondant à la solution optimale. La méthode du complément de Schur [40] permet de résoudre chaque approximation en deux étapes. La première étape permet de calculer les variables d'interfaces de la matrice Jacobienne. Puis les sous-systèmes correspondant aux matrices A_i , $i = 1, \dots, k$ peuvent être résolus en parallèle. La méthode du complément de Schur est expliquée en détail dans [37].

Nous allons voir dans la section suivante comment modéliser la matrice Jacobienne en termes de graphe.

7.2 Matrices Jacobiennes et graphes

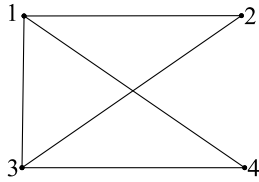
Comme nous l'avons vu dans la précédente section, à chaque étape de la résolution d'un SAD (ou SDO), les valeurs des variables d'interface sont recalculées et transmises de nouveau sur chaque processeur. De plus, les variables d'interface peuvent être transmises si et seulement si tous les processeurs ont terminé leurs calculs. Les sous-systèmes doivent donc être assez équilibrés pour éviter que les processeurs attendent celui qui possède le plus grand sous-système, qui sera le plus long à calculer. Les variables d'interfaces étant calculées au préalable, minimiser leur nombre revient à minimiser le temps total de la résolution. Le problème se ramène donc à minimiser le nombre de variables d'interfaces, de telle manière que les sous-systèmes associés aux matrices A_1, \dots, A_k sur chaque processeur soient assez équilibrés. Afin de réécrire la matrice Jacobienne J de taille $n \times n$ associée à un système sous la forme bloc diagonale bordée, seule la position des valeurs non nuls nous intéresse. Nous allons donc travailler sur une matrice d'incidence M_I définie de la manière suivante. Soit M_I la matrice d'incidence de taille $n \times n$, telle que $M_I(i, j) = 1$ si $J(i, j) \neq 0$ ou $J(j, i) \neq 0$ et $M_I(i, j) = 0$ sinon. Le problème revient à réorganiser la matrice d'incidence M_I pour obtenir k sous-matrices indépendantes sur la diagonale n'ayant ni ligne ni colonne en commun et de taille globale maximum. Ce problème peut être modélisé en termes de graphes de la manière suivante. Considérons le graphe $G = (V, E)$, où l'ensemble des sommets V correspond aux lignes/colonnes de la matrice et l'ensemble d'arêtes E est tel qu'il existe une arête $ij \in E$, avec $i > j$, si et seulement si $M_I(i, j) = 1$. Le problème revient donc à déterminer un ensemble de sommet V_0 de cardinalité minimum tel que l'ensemble de sommets $V \setminus V_0$ puisse être partitionné en k sous-ensembles V_1, \dots, V_k deux à deux déconnectés et suffisamment équilibrés.

Considérons la matrice Jacobienne J associée au système (S_7) . La matrice M_I associée à J s'écrit :

$$M_I = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Le graphe G associé à la matrice M_I est donné par la figure 7.1.

Nous allons voir dans la section suivante que cela se ramène à résoudre un problème de séparateur sous certaines conditions.

FIGURE 7.1 – Graphe G .

7.3 Problème du séparateur

Soit $G = (V, E)$ un graphe non-orienté, connexe, avec $|V| = n$. Un k -séparateur, pour $k \geq 2$, est un ensemble de sommet $V_0 \subseteq V$ tel que $V \setminus V_0$ puisse être partitionné en k sous-ensembles V_1, \dots, V_k , déconnectés deux à deux, c'est-à-dire qu'il n'existe pas d'arête $u_i u_j$ avec $u_i \in V_r, u_j \in V_t$, où $r \neq t \in \{1, \dots, k\}$. Le *problème du k -séparateur* consiste à trouver un k -séparateur de cardinalité minimum. Dans certaines applications on impose que les éléments V_1, \dots, V_k de la partition soient suffisamment équilibrés, c'est-à-dire que la différence des cardinalités de deux ensembles V_i données est bornée par un entier q fixé. On appelle ce problème, le *problème du k -séparateur équilibré*. Une autre variante qui est souvent considérée pour le problème du k -séparateur consiste à borner la taille maximale des éléments V_1, \dots, V_k en assurant que tous les sous-ensembles V_1, \dots, V_k sont non-vides. Cette variante est appelée le *problème du k -séparateur borné*. On note que dans la version équilibrée du problème du k -séparateur, les ensembles V_1, \dots, V_k peuvent être vides. Alors que dans les autres, ces ensembles ne doivent pas être vides. Les problèmes du k -séparateur borné et du k -séparateur équilibré sont connus pour être NP-difficiles et ce pour $k \geq 2$ [58]. Alors que le problème du 2-séparateur peut-être résolu en temps polynomial. La complexité du problème du k -séparateur pour $k \geq 3$ n'est pas connue.

Dans ce chapitre nous considérons le problème du k -séparateur équilibré. Le problème de la réécriture de la matrice Jacobienne se ramène à ce problème dans le graphe G associé à cette matrice.

La recherche d'un k -séparateur équilibré intervient dans de nombreux domaines. Dans celui des télécommunications, l'ensemble V_0 correspond à un goulot d'étranglement. Il représente dans ce cas les sommets importants, qui seront le plus utilisés dans le réseau et qui caractérise la capacité du réseau [24]. Le problème du k -séparateur équilibré est aussi utilisé dans les stratégies "diviser pour régner" [6]. Le problème du k -séparateur équilibré a aussi des applications intéressantes dans la résolution des systèmes linéaires

algébriques [72]. En effet, vu la grande taille de ces systèmes, leurs résolutions imposent une puissance de calcul de plus en plus élevée. Alors paralléliser la résolution de ces systèmes devient de plus en plus indispensables. Et paralléliser la résolution sur k processeurs se ramène à la résolution du problème de k -séparateur équilibré. Dans la littérature, le problème du k -séparateur a surtout été étudié pour résoudre des instances de très grande taille (des graphes contenant plus de 1.000.000 de sommets). Comme ce problème est NP-difficile, ces études ont abouti à des heuristiques permettant d'avoir des solutions acceptables (mais loin de l'optimum) pour des instances de très grande taille [45, 46, 47]. Ces heuristiques ont données lieu à des logiciels comme Métis [5] ou CHACO [2] qui sont largement utilisés par des industriels. La résolution des problèmes de petites tailles (des graphes contenant quelques centaines de sommets) par ces heuristiques fournissent de très mauvais résultats. Pour résoudre ces problèmes il serait plus intéressant d'utiliser des méthodes exactes.

Balas et Souza [8] ont étudié le problème du 2-séparateur borné en utilisant une approche polyédrale. Cette étude a abouti au développement d'un algorithme de coupes et branchements [17] pour résoudre le problème à l'optimum. Ils ont résolu ce problème sur des graphes allant jusqu'à 200 sommets. Leur étude exploite la relation entre les séparateurs et les dominants dans les graphes

Didi Biha et Meurs [18] ont également étudié le problème du 2-séparateur borné. Ils ont exhibé une relation entre le nombre de chemins sommet-disjoints entre une paire de sommets et un 2-séparateur borné dans les graphes. En utilisant cette relation, ils ont proposé une nouvelle contrainte valide pour le problème, permettant de borner la cardinalité de l'ensemble V_0 . Cette contrainte leur a permis d'avoir de meilleurs résultats sur les instances utilisées par Balas et Souza.

Dans ce qui suit nous donnons des formulations approchées et exactes pour le problème du k -séparateur équilibré.

7.4 Formulations

Nous allons présenter dans un premier temps deux formulations approchées du problème du k -séparateur équilibré. La motivation de ces formulations est d'éviter la contrainte d'équilibrage, qui généralement complexifie grandement le problème. Nous donnons ensuite une formulation exacte de ce problème.

7.4.1 Formulations approchées

Soit $G = (V, E)$ un graphe et k un entier fixé. Associons à chaque sommet $i \in V$ et chaque $t = 1, \dots, k$, une variable binaire x_i^t valant 1 si le sommet i appartient à V_t et 0 sinon.

La première formulation que l'on propose permet d'ordonner les éléments de la partition en fonction de leur cardinalité : $|V_1| \leq |V_2| \leq \dots \leq |V_k|$. L'idée ici est de maximiser $|V_1|$. Comme $|V_1| \leq |V_2| \leq \dots \leq |V_k|$ alors les cardinalités des ensembles V_2, \dots, V_k seront implicitement maximisées et peuvent être par conséquent assez équilibrés. Aussi V_0 peut-être suffisamment réduite. Nous avons donc la formulation.

$$\begin{aligned} \max \sum_{i=1}^n x_i^1 \\ \sum_{i \in V} x_i^t \leq \sum_{i \in V} x_i^r, \quad \text{pour tout } t = r - 1, r \in \{2, \dots, k\}, \end{aligned} \quad (7.9)$$

$$\sum_{t=1}^k x_i^t \leq 1, \quad \text{pour tout } i \in V, \quad (7.10)$$

$$x_i^t + x_j^r \leq 1, \quad \text{pour tout } t \neq r \in \{1, \dots, k\} \text{ et pour tout } ij \in E, \quad (7.11)$$

$$x_i^t \geq 0, \quad \text{pour tout } i \in V, t = 1, \dots, k, \quad (7.12)$$

$$x_i^t \leq 1, \quad \text{pour tout } i \in V, t = 1, \dots, k, \quad (7.13)$$

$$x_i^t \text{ entier}, \quad \text{pour tout } i \in V, t = 1, \dots, k, \quad (7.14)$$

L'inégalité (7.9) exprime la relation d'ordre sur la cardinalité des ensembles de la partition ($|V_1| \leq |V_2| \leq \dots \leq |V_k|$). Les inégalités (7.10) expriment le fait qu'un sommet ne peut pas appartenir à deux ensembles de la partition. Les inégalités (7.11) assurent qu'il n'existe pas d'arête entre deux ensembles de la partition.

Dans la deuxième formulation on minimise $|V_0| + |V_k|$ en conservant toujours l'ordre $|V_1| \leq |V_2| \leq \dots \leq |V_k|$. Si $|V_0| + |V_k|$ est minimisée, alors V_1, \dots, V_{k-1} peuvent être proches de V_k et réaliser ainsi un certain équilibrage et V_0 peut-être suffisamment petit.

Alors la formulation s'écrit sous la forme :

$$\begin{aligned} & \max \sum_{i \in V} x_i^1 + \sum_{i \in V} x_i^2 + \dots + \sum_{i \in V} x_i^{k-1} \\ & \text{s.c. (7.9)-(7.14)} \end{aligned}$$

Les deux formulations précédentes seront analysées expérimentalement dans la section 7.6 et comparées à la formulation exacte qui sera l'objet des sections suivantes.

7.4.2 Problème du k -séparateur équilibré

Soient $G = (U, E)$ un graphe, k un entier supérieur ou égal à 2 et q un entier supérieur ou égal à 0. On note dans le reste de ce chapitre le problème du k -séparateur équilibré par PkSE.

Pour le problème du k -séparateur équilibré, minimiser $|V_0|$ revient à maximiser $|V_1| + |V_2| + \dots + |V_k|$. Le problème du k -séparateur équilibré est donc équivalent au programme en nombres entiers P_k suivant.

$$\begin{aligned} & \max \sum_{i \in V} \sum_{t=1}^k x_i^t \\ & \sum_{i \in V} x_i^t - \sum_{i \in V} x_i^r \leq q \quad \text{pour tout } t \neq r \in \{1, \dots, k\}, \end{aligned} \quad (7.15)$$

$$\sum_{t=1}^k x_i^t \leq 1, \quad \text{pour tout } i \in V, \quad (7.16)$$

$$x_i^t + x_j^r \leq 1 \quad \text{pour tout } t \neq r \in \{1, \dots, k\} \text{ et pour tout } ij \in E, \quad (7.17)$$

$$x_i^t \geq 0 \quad \text{pour tout } i \in V, t \in \{1, \dots, k\}, \quad (7.18)$$

$$x_i^t \leq 1 \quad \text{pour tout } i \in V, t \in \{1, \dots, k\}, \quad (7.19)$$

$$x_i^t \text{ entier} \quad \text{pour tout } i \in V, t \in \{1, \dots, k\}, \quad (7.20)$$

Ici les variables x_i^t sont définies de la même manière que dans les formulations précédentes. Les inégalités (7.15) expriment l'équilibre entre les ensembles V_i . Elles imposent que deux ensembles de la partition diffèrent au maximum de q éléments. Ces contraintes seront appelées *inégalités d'équilibrage*. Les contraintes (7.16) seront appelées *inégalités de partition*. Les contraintes (7.17) seront appelées *inégalités d'arêtes*. Les inégalités (7.19) et (7.18) sont les inégalités triviales et les conditions (7.20) sont les contraintes d'intégrité.

7.5 Étude polyédrale

Dans cette section, on étudie l'enveloppe convexe des solutions du problème du k -séparateur équilibré donné par la formulation P_k . On caractérise la dimension du polytope des solutions et on donne des conditions nécessaires et suffisantes pour que les contraintes de base de la formulation définissent des facettes. On introduit aussi de nouvelles contraintes valides et étudie leur aspect facial. Le but de cette étude et de développer un algorithme de coupes et branchements pour le problème.

Soit $P(G, k, q)$ l'enveloppe convexe des solutions du programme de la formulation P_k , c'est-à-dire,

$$P(G, k, q) = \text{conv}(\{x \in \{0, 1\}^{kn} \mid x \text{ satisfait (7.15) - (7.17)}\}).$$

Si $\pi = (V_0, V_1, \dots, V_k)$ est une partition de G , alors le vecteur d'incidence de π , x^π est donné par

$$x_{i,r}^\pi = \begin{cases} 1 & \text{si } i \in V_r. \\ 0 & \text{sinon,} \end{cases} \quad \text{pour tout } i \in V \text{ et pour tout } r \in \{1, \dots, k\}.$$

7.5.1 Dimension de $P(G, k, q)$

Dans cette section, on caractérise la dimension de $P(G, k, q)$.

Proposition 7.1 $P(G, k, q)$ est de pleine dimension si $q > 0$.

Preuve. Supposons que $q > 0$. Il suffit d'exhiber $nk + 1$ solutions du PkSE dont les vecteurs d'incidence sont affinement indépendants. Considérons les solutions $\pi^0 = (V_0^0, V_1^0, \dots, V_k^0)$ avec $V_0^0 = V$, $V_i^0 = \emptyset$ pour $i = 1, \dots, k$, et $\pi^{i,t} = (V_0^{i,t}, V_1^{i,t}, \dots, V_k^{i,t})$, $t \in \{1, \dots, k\}$, $i \in V$ avec $V_0^{i,t} = V \setminus \{i\}$, $V_t^{i,t} = \{i\}$, $V_r^{i,t} = \emptyset$ pour $r \in \{1, \dots, k\} \setminus \{t\}$. Les vecteurs d'incidence de ces solutions sont affinement indépendants. \square

On remarque que si $q = 0$, alors certaines égalités apparaissent dans le programme (P_k) et par conséquent $P(G, k, q)$ n'est pas de pleine dimension. Nous considérons, dans la suite de notre étude que $q > 0$.

7.5.2 Inégalités valides

Dans cette section, nous allons voir, dans un premier temps, des conditions nécessaires et suffisantes pour que les contraintes de base de la formulation définissent des facettes. Nous introduisons ensuite de nouvelles contraintes valides et discutons de leurs aspect facial.

7.5.2.1 Inégalités triviales et facettes

On remarque que les inégalités (7.19) ne définissent pas de facettes de $P(G, k, q)$, car elles sont dominées par les contraintes (7.16). Par contre les contraintes (7.18) le sont comme le montre le résultat suivant.

Proposition 7.2 *Les inégalités (7.18) définissent des facettes de $P(G, k, q)$.*

Preuve. Il suffit d'exhiber nk solutions du k -séparateur équilibré dont les vecteurs d'incidence sont affinement indépendants et vérifiant $x_j^r = 0$. Considérons les solutions

- $\pi^0 = (V_0^0, V_1^0, \dots, V_k^0)$ avec $V_0^0 = V$, $V_i^0 = \emptyset$ pour $i = 1, \dots, k$.
- $\pi^{i,t} = (V_0^{i,t}, V_1^{i,t}, \dots, V_k^{i,t})$, $t \in \{1, \dots, k\}$, $i \in V \setminus \{j\}$ avec $V_0^{i,t} = V \setminus \{i\}$, $V_t^{i,t} = \{i\}$, $V_s^{i,t} = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$,
- $\pi^{j,t} = (V_0^{j,t}, V_1^{j,t}, \dots, V_k^{j,t})$, $t \in \{1, \dots, k\} \setminus \{r\}$, avec $V_0^{j,t} = V \setminus \{j\}$, $V_t^{j,t} = \{j\}$, $V_s^{j,t} = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$

Nous avons nk solutions dont les vecteurs d'incidence sont affinement indépendants et vérifient $x_j^r = 0$. \square

7.5.2.2 Inégalités de partition

Dans la proposition suivante, nous donnons des conditions nécessaires et suffisantes pour que les contraintes de partition définissent des facettes. Pour cela, on introduit deux notations. Soient $G = (V, E)$ un graphe et W un sous-ensemble de sommets de V . Notons par $N(W)$ l'ensemble des sommets voisins de W , c'est-à-dire les sommets adjacents aux sommets de W . De plus, on note par $\tilde{N}(W)$ l'ensemble $W \cup N(W)$.

Proposition 7.3 *L'inégalité (7.16), associée à $i \in V$, définit une facette de $P(G, k, q)$ si et seulement si soit $q > 1$ soit pour toute arête $ij \in E$ il existe dans G un stable contenu dans $V \setminus \tilde{N}(\{i, j\})$ de cardinalité supérieure ou égale à $k - 1$.*

Preuve. (\Leftarrow) Supposons tout d'abord que $q = 1$ et qu'il existe une arête $ij \in E$ telle que G ne contient pas de stable inclus dans $V \setminus \tilde{N}(\{i, j\})$ de cardinalité supérieure ou égale à $k - 1$. Dans ce cas, la contrainte (7.16) est impliquée par la contrainte

$$\sum_{t=1}^k x_i^t + \sum_{t=1}^k x_j^t \leq 1. \quad (7.21)$$

Enfin, la contrainte (7.21) est valide. En effet, supposons le contraire, c'est-à-dire qu'il existe une solution π tel que x^π ne satisfait pas la contrainte. Dans ce cas, i et j n'appartiennent pas à V_0 . Comme $ij \in E$, alors i et j appartiennent à un même sous-ensemble, disons V_1 . Comme π est un k -séparateur, $N(\{i, j\}) \subseteq V_0$ ou bien $N(\{i, j\}) \subseteq V_1$. Afin que π soit équilibré, V_t , pour tout $t \in \{2, \dots, k\}$, doit contenir au moins un sommet de $V \setminus \tilde{N}(\{i, j\})$. Comme $V \setminus \tilde{N}(\{i, j\})$ ne contient pas de stable de cardinalité supérieure ou égale à $k - 1$, il n'est pas possible d'avoir un sommet de $\tilde{N}(\{i, j\})$ dans V_i , pour tout $i \in \{2, \dots, k\}$. Alors nous avons une contradiction.

(\Rightarrow) Notons par $ax \leq \alpha$ la contrainte (7.16). Soit $bx \leq \beta$ une inégalité qui définit une facette de $P(G, k, q)$. Supposons que $\{x \in P(G, k, q) : ax = \alpha\} \subseteq \{x \in P(G, k, q) : bx = \beta\}$. Puisque $P(G, k, q)$ est de pleine dimension, il suffit de montrer qu'il existe $\rho > 0$ tel que $b = \rho a$.

Étant donné $r \neq t \in \{1, \dots, k\}$, considérons les partitions

- $\pi_r = (V_0^r, V_1^r, \dots, V_k^r)$ avec $V_0^r = V \setminus \{i\}$, $V_r^r = \{i\}$, $V_s^r = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{r\}$.
- $\pi_t = (V_0^t, V_1^t, \dots, V_k^t)$ avec $V_0^t = V \setminus \{i\}$, $V_t^t = \{i\}$, $V_s^t = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$.

Les partitions π_r et π_t induisent des k -séparateurs équilibrés. Comme les vecteurs d'incidence de π_r et π_t vérifient $ax = \alpha$, il s'ensuit que $bx^{\pi_t} = bx^{\pi_r}$. Par conséquent, $b(w_i^r) = b(w_i^t)$. On en déduit alors

$$b(w_i^t) = b(w_i^r) = \rho, \quad \text{pour tout } t \in \{1, \dots, k\} \text{ et } r \in \{1, \dots, k\} \setminus \{t\}, \quad (7.22)$$

pour un certain scalaire ρ dans \mathbb{R} .

Soient $j \in V \setminus \{i\}$ un sommet non incident à i et $r \in \{1, \dots, k\}$. Considérons les partitions

- $\pi_{jr}^1 = (V_0^1, V_1^1, \dots, V_k^1)$ avec $V_0^1 = V \setminus \{i\}$, $V_t^1 = \{i\}$, $V_s^1 = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$,
- $\pi_{jr}^2 = (V_0^2, V_1^2, \dots, V_k^2)$ avec $V_0^2 = V \setminus \{ij\}$, $V_t^2 = \{i\}$, $V_r^2 = \{j\}$, $V_s^2 = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t, r\}$.

Il est clair que les partitions π_{jr}^1 et π_{jr}^2 induisent des k -séparateurs équilibrés. Comme

les vecteurs d'incidence de π_{jr}^1 et π_{jr}^2 vérifient $ax = \alpha$, il s'ensuit que $bx^{\pi_{jr}^1} = bx^{\pi_{jr}^2}$. Nous avons alors

$$b(w_j^r) = 0, \quad \text{pour tout } j \in V \setminus \{i\} \text{ tel que } ij \notin E \text{ et pour } r = 1, \dots, k. \quad (7.23)$$

Nous montrons maintenant que $b(w_j^t) = 0$ pour tout sommet $j \in V \setminus \{i\}$ incident à i et pour $t = 1, \dots, k$.

Supposons d'abord que $q > 1$. Dans ce cas, étant donné $j \in V \setminus \{i\}$ tel que $ij \in E$ et $t = 1, \dots, k$, considérons les partitions

- $\pi_{jt}^1 = (V_0^1, V_1^1, \dots, V_k^1)$ avec $V_0^1 = V \setminus \{i\}$, $V_t^1 = \{i\}$, $V_s^1 = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$,
- $\pi_{jt}^2 = (V_0^2, V_1^2, \dots, V_k^2)$ avec $V_0^2 = V \setminus \{ij\}$, $V_t^2 = \{i, j\}$, $V_s^2 = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$.

Comme π_{jt}^1 et π_{jt}^2 induisent des k -séparateurs équilibrés et leurs vecteurs d'incidence vérifient $ax = \alpha$, il s'ensuit que $bx^{\pi_{jt}^1} = bx^{\pi_{jt}^2}$. Par conséquent, $b(w_j^t) = 0$.

Supposons maintenant que $q = 1$. Étant donné $j \in V \setminus \{i\}$ tel que $ij \in E$ et $t = 1, \dots, k$, par hypothèse il existe un stable $S \subseteq V \setminus N(\{i, j\})$ tel que $|S| \geq k - 1$. Sans perte de généralité, on suppose que S contient l'ensemble $\{v_s : s \in \{1, \dots, k\}, s \neq t\}$. Considérons les partitions

- $\pi_{jt}^3 = (V_0^3, V_1^3, \dots, V_k^3)$ avec $V_0^3 = V \setminus \{i\} \setminus S$, $V_t^3 = \{i\}$, $V_s^3 = \{v_s\}$ pour $s \in \{1, \dots, k\} \setminus \{t\}$,
- $\pi_{jt}^4 = (V_0^4, V_1^4, \dots, V_k^4)$ avec $V_0^4 = V \setminus \{ij\} \setminus S$, $V_t^4 = \{i, j\}$, $V_s^4 = \{v_s\}$ pour $s \in \{1, \dots, k\} \setminus \{t\}$.

Comme les partitions π_{jt}^3 et π_{jt}^4 induisent des k -séparateurs équilibrés et leurs vecteurs d'incidence vérifient $ax = \alpha$, il s'ensuit que $bx^{\pi_{jt}^3} = bx^{\pi_{jt}^4}$. Par conséquent, $b(w_j^t) = 0$.

Nous avons alors

$$b(w_j^t) = 0, \quad \text{pour tout } ij \in E \text{ et pour } t = 1, \dots, k. \quad (7.24)$$

Par (7.22)-(7.24), il s'ensuit que $b = \rho a$. De plus, comme pour tout sommet $i \in V$, il existe un séparateur contenant i tel que la contrainte soit vérifiée à l'égalité, la face définie par $ax \leq \alpha$ est différente d'une face triviale $\{x \in P(G, k, q) : x_{ij} = 0\}$. Cela implique que la face induite par $bx \leq \beta$ n'est pas contenue dans une face triviale. Il s'ensuit alors que $b(w_i^t) \geq 0$ pour tout $i \in V$ et $t \in \{1, \dots, k\}$. Comme $bx \leq \beta$ définit une facette de $P(G, k, q)$, il doit donc exister $i \in V$ et $t \in \{1, \dots, k\}$ tel que $b(w_i^t) > 0$. Par conséquent $\rho > 0$ et la preuve est terminée. \square

7.5.2.3 Inégalités d'arêtes

Les inégalités d'arêtes (7.17) ne définissent pas des facettes. Nous donnons dans la proposition suivante une version renforcée des inégalités d'arêtes.

Proposition 7.4 *Soient $ij \in E$ une arête et un entier $z \leq k$. La contrainte*

$$x_i^z + \sum_{t \in \{1, \dots, k\} \setminus \{z\}} x_j^t \leq 1, \quad (7.25)$$

est valide.

Preuve. Soient $ij \in E$ une arête et un entier $z \leq k$. Supposons qu'il existe une solution π de $P(G, k, q)$ telle que le vecteur d'incidence x^π ne satisfait pas (7.25), c'est-à-dire x^π vérifie

$$x_i^{\pi z} + \sum_{t \in \{1, \dots, k\} \setminus \{z\}} x_j^{\pi t} \geq 2, \quad (7.26)$$

Il est clair qu'il n'existe pas de partition telle que $j \in V_t$ et $j \in V_r$ avec $t \neq r \in \{1, \dots, k\} \setminus \{z\}$. Comme il existe une arête ij il ne peut pas exister de k -séparateur équilibré tel que $i \in V_z$ et $j \in V_r$, pour tout $r \in \{1, \dots, k\} \setminus \{z\}$. La contrainte (7.25) est donc valide. \square

Proposition 7.5 *L'inégalité (7.25) associée à $ij \in E$ et $z \in \{1, \dots, k\}$ définit une facette de $P(G, k, q)$ si et seulement si soit $q > 1$ soit G contient un stable de cardinalité supérieure ou égale à $k - 1$ inclus dans $V \setminus \tilde{N}(\{i, j\})$.*

Preuve. (\Leftarrow) Supposons que $q = 1$ et qu'il n'existe pas dans G un stable de cardinalité supérieure ou égale à $k - 1$ inclus dans $V \setminus \tilde{N}(\{i, j\})$. Dans ce cas, la contrainte (7.21) $\sum_{t=1}^k x_i^t + \sum_{t=1}^k x_j^t \leq 1$ est valide et domine la contrainte (7.25). La preuve de la validité de cette contrainte est similaire à celle donnée dans la preuve précédente.

(\Rightarrow) Notons par $ax \leq \alpha$ la contrainte (7.25). Soit $bx \leq \beta$ une inégalité qui définit une facette de $P(G, k, q)$. Supposons que $\{x \in P(G, k, q) : ax = \alpha\} \subseteq \{x \in P(G, k, q) : bx = \beta\}$. Puisque $P(G, k, q)$ est de pleine dimension, il suffit de montrer qu'il existe $\rho > 0$ tel que $b = \rho a$.

Soient $t_1 \neq t_2 \in \{1, \dots, k\} \setminus \{z\}$. Considérons les partitions
- $\pi^1 = (V_0^1, V_1^1, \dots, V_k^1)$ avec $V_0^1 = V \setminus \{j\}$, $V_{t_1}^1 = \{j\}$, $V_t^1 = \emptyset$ pour $t \in \{1, \dots, k\} \setminus \{t_1\}$,
- $\pi^2 = (V_0^2, V_1^2, \dots, V_k^2)$ avec $V_0^2 = V \setminus \{j\}$, $V_{t_2}^2 = \{j\}$, $V_t^2 = \emptyset$ pour $t \in \{1, \dots, k\} \setminus \{t_2\}$.
Les partitions π^1 et π^2 induisent des k -séparateurs équilibrés, ce qui implique que x^{π^1} et x^{π^2} appartiennent à $P(G, k, q)$. De plus, x^{π^1} et x^{π^2} vérifient $ax = \alpha$. On obtient alors $bx^{\pi^1} = bx^{\pi^2}$, ce qui implique que $b(w_j^{t_1}) = b(w_j^{t_2})$. Comme t_1 et t_2 sont quelconque, on en déduit alors

$$b(w_j^t) = b(w_j^r) = \rho, \quad \text{pour tout } t \in \{1, \dots, k\} \setminus \{z\} \text{ et } r \in \{1, \dots, k\} \setminus \{t, z\}, \quad (7.27)$$

et un certain scalaire ρ dans \mathbb{R} .

Considérons la solution $\pi^3 = (V_0^3, V_1^3, \dots, V_k^3)$ avec $V_0^3 = V \setminus \{i\}$, $V_z^3 = \{i\}$, $V_t^3 = \emptyset$ pour $t \in \{1, \dots, k\} \setminus \{z\}$. x^{π^3} est une solution de $P(G, k, q)$ vérifiant $ax = \alpha$, ce qui implique que $bx^{\pi^1} = bx^{\pi^3}$. On en déduit alors que

$$b(w_i^z) = b(w_j^t), \quad \text{pour tout } t \in \{1, \dots, k\} \setminus \{z\}. \quad (7.28)$$

On montre maintenant que $b(w_l^r) = 0$ pour tout $l \in V \setminus \{i, j\}$ et pour tout $r \in \{1, \dots, k\} \setminus \{z\}$. Soient $v_1 \in V \setminus \{i, j\}$ et $r_1 \in \{1, \dots, k\} \setminus \{z\}$. Supposons que $iv_1 \notin E$. Dans ce cas, la partition

- $\pi^4 = (V_0^4, V_1^4, \dots, V_k^4)$ avec $V_0^4 = V \setminus \{i, v_1\}$, $V_z^4 = \{i\}$, $V_{r_1}^4 = \{v_1\}$, $V_t^4 = \emptyset$ pour $t \in \{1, \dots, k\} \setminus \{z, r_1\}$,

est une solution telle que $ax = \alpha$. On déduit alors que $bx^{\pi^3} = bx^{\pi^4}$, ce qui implique que $b(w_{v_1}^{r_1}) = 0$. Supposons maintenant que $iv_1 \in E$.

Si $q > 1$, alors les partitions

$\pi^5 = (V_0^5, V_1^5, \dots, V_k^5)$ avec $V_0^5 = V \setminus \{j, v_1\}$, $V_{r_1}^5 = \{v_1, j\}$, $V_s^5 = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{r_1\}$,

$\pi^6 = (V_0^6, V_1^6, \dots, V_k^6)$ avec $V_0^6 = V \setminus \{j\}$, $V_{r_1}^6 = \{j\}$, $V_s^6 = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{r_1\}$,
sont des solutions et vérifient $ax^{\pi^5} = ax^{\pi^6} = \alpha$. Donc $bx^{\pi^5} = bx^{\pi^6}$. On a alors $b(w_{v_1}^{r_1}) = 0$.

Si $q = 1$, comme $v_1 \in \tilde{N}(\{i, j\})$, on sait, par hypothèse, qu'il existe $k - 1$ sommets, disons u_t , $t \in \{1, \dots, k\} \setminus \{r_1\}$, formant un stable S dans G et qui sont incidents ni à j ni à v_1 . Dans ce cas, les partitions

$\pi^7 = (V_0^7, V_1^7, \dots, V_k^7)$ avec $V_0^7 = V \setminus (S \cup \{j, v_1\})$, $V_{r_1}^7 = \{v_1, j\}$, $V_t^7 = \{u_t\}$ pour $t \in \{1, \dots, k\} \setminus \{r_1\}$,

$\pi^8 = (V_0^8, V_1^8, \dots, V_k^8)$ avec $V_0^8 = V \setminus (S \cup \{j\})$, $V_{r_1}^8 = \{j\}$, $V_t^8 = \{u_t\}$ pour $t \in \{1, \dots, k\} \setminus \{r_1\}$.

sont des solutions telle que $ax = \alpha$. On déduit que $bx^{\pi^7} = bx^{\pi^8}$, on a alors $b(w_{v_1}^{r_1}) = 0$. Comme v_1 est un sommet quelconque de $V \setminus \{i, j\}$ et $r_1 \in \{1, \dots, k\} \setminus \{z\}$, on en déduit, en répétant le raisonnement, que

$$b(w_l^r) = 0, \quad \text{pour tout } l \in V \setminus \{i, j\} \text{ et pour tout } r \in \{1, \dots, k\} \setminus \{z\}. \quad (7.29)$$

On montre maintenant que $b(w_l^z) = 0$ pour tout $l \in V \setminus \{i, j\}$. Soient $v_1 \in V \setminus \{i, j\}$ et $r_1 \in \{1, \dots, k\} \setminus \{z\}$. Supposons que $qv_1 \notin E$. Dans ce cas, la partition $\pi^9 = (V_0^9, V_1^9, \dots, V_k^9)$ avec $V_0^9 = V \setminus \{j, v_1\}$, $V_z^9 = \{v_1\}$, $V_{r_1}^9 = \{j\}$, $V_t^9 = \emptyset$ pour $t \in \{1, \dots, k\} \setminus \{z, r_1\}$, est une solution telle que $ax = \alpha$. Si on pose $r_1 = t_1$, on en déduit que $bx^{\pi^9} = bx^{\pi^1}$, ce qui implique que $b(w_{v_1}^z) = 0$. Supposons maintenant que $qv_1 \in E$.

Si $q > 1$, alors les partitions

$\pi^{10} = (V_0^{10}, V_1^{10}, \dots, V_k^{10})$ avec $V_0^{10} = V \setminus \{i, v_1\}$, $V_z^{10} = \{v_1, i\}$, $V_s^{10} = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{z\}$,

$\pi^{11} = (V_0^{11}, V_1^{11}, \dots, V_k^{11})$ avec $V_0^{11} = V \setminus \{i\}$, $V_z^{11} = \{i\}$, $V_s^{11} = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{z\}$,

sont des solutions telle que $ax = \alpha$. Par conséquent, $bx^{\pi^{10}} = bx^{\pi^{11}}$. On a donc $b(w_{v_1}^z) = 0$.

Si $q = 1$, comme $V_1 \in \tilde{N}(\{i, j\})$, on sait, par hypothèse, qu'il existe $k - 1$ sommets, disons u_t , $t \in \{1, \dots, k\} \setminus \{r_1\}$, formant un stable S dans G et qui sont incidents ni à i ni à v_1 . Dans ce cas, les partitions

$\pi^{12} = (V_0^{12}, V_1^{12}, \dots, V_k^{12})$ avec $V_0^{12} = V \setminus (S \cup \{i, v_1\})$, $V_z^{12} = \{v_1, i\}$, $V_t^{12} = \{u_t\}$ pour $t \in \{1, \dots, k\} \setminus \{z\}$,

$\pi^{13} = (V_0^{13}, V_1^{13}, \dots, V_k^{13})$ avec $V_0^{13} = V \setminus (S \cup \{i\})$, $V_z^{13} = \{i\}$, $V_t^{13} = \{u_t\}$ pour $t \in \{1, \dots, k\} \setminus \{z\}$

sont des solutions telles que $ax^{\pi^{12}} = ax^{\pi^{13}} = \alpha$. Alors $bx^{\pi^{12}} = bx^{\pi^{13}}$, on a donc $b(w_{v_1}^z) = 0$. Comme v_1 est un sommet quelconque de $V \setminus \{i, j\}$, on en déduit, en répétant le raisonnement, que

$$b(w_l^z) = 0, \quad \text{pour tout } l \in V \setminus \{i, j\}. \quad (7.30)$$

Par (7.27)-(7.30), il s'ensuit que $b = \rho a$. De plus, comme pour tout sommet $i \in V$, il existe un séparateur contenant i tel que la contrainte soit vérifiée à l'égalité, la face définie par $ax \leq \alpha$ est différente d'une face triviale $\{x \in P(G, k, q) : x_{ij} = 0\}$. Cela implique que la face induite par $bx \leq \beta$ n'est pas contenue dans une face triviale. Il s'ensuit que $b(w_i^t) \geq 0$ pour tout $i \in V$ et $t \in \{1, \dots, k\}$. Comme $bx \leq \beta$ définit une facette de $P(G, k, q)$, il doit donc exister au moins un séparateur $i \in V$ et $t \in \{1, \dots, k\}$ tel que $b(w_i^t) > 0$. Par conséquent $\rho > 0$ et la preuve est terminée. \square

7.5.2.4 Inégalités d'équilibre

Nous proposons, dans cette section, une famille de contraintes qui domine les contraintes d'équilibre (7.15) de la formulation de base.

Pour tout $i \in V$, on note par $\bar{N}'(i)$ l'ensemble des sommets de $V \setminus \{i\}$ qui ne sont pas adjacents à i . $\bar{N}'(i) = V \setminus (N(i) \cup \{i\})$.

Proposition 7.6 *Soit $W \subseteq V$ l'ensemble des sommets tel que pour tout $v \in W$, soit $|\bar{N}'(v)| < q + k - 1$ soit $\bar{N}'(v)$ ne contient pas de stable de cardinalité $k - 2$. Alors la contrainte*

$$\sum_{i \in V} x_i^t - \sum_{j \in V \setminus W} x_j^r \leq q \quad \text{pour tout } t \neq r \in \{1, \dots, k\}, \quad (7.31)$$

est valide.

Preuve. Supposons qu'il existe $t \neq r \in \{1, \dots, k\}$ tel que la contrainte (7.31) associée à r et t n'est pas valide. Sans perte de généralité, on considère $t = 1$ et $r = 2$. Si (7.31) n'est pas valide, alors il existe $\bar{x} \in \{0, 1\}^{nk}$ solution de $P(G, k, q)$ ne vérifiant pas (7.31), c'est-à-dire

$$\sum_{i \in V} x_i^1 - \sum_{j \in V \setminus W} x_j^2 \geq q + 1. \quad (7.32)$$

Comme \bar{x} satisfait

$$\sum_{i \in V} x_i^1 - \sum_{j \in V} x_j^2 \leq q, \quad (7.33)$$

$$- \sum_{i \in V} x_i^1 + \sum_{j \in V \setminus W} x_j^2 \leq -q - 1. \quad (7.34)$$

On en déduit

$$\sum_{j \in W} x_j^2 \geq 1. \quad (7.35)$$

De plus, on a $q + 1$ sommets dans V_1 . Comme \bar{x} est solution, on doit avoir un sommet dans V_t , $t \in \{1, \dots, k\} \setminus \{1, 2\}$. Soit $j \in W$ dans V_2 . Comme on a au minimum $q + 1$ sommets dans V_1 et 1 sommet dans V_t , $t \in \{1, \dots, k\} \setminus \{1, 2\}$, $|\bar{N}'(j)| \geq q + k - 1$. De plus, les $k - 2$ sommets dans V_t , $t \in \{1, \dots, k\} \setminus \{1, 2\}$, forment un stable de cardinalité $k - 2$ dans $\bar{N}'(j)$. Il s'en suit que $j \notin W$ et donc une contradiction. \square

On appelle ces contraintes les *inégalités de w -équilibre*.

Nous allons nous intéresser maintenant aux conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes.

Proposition 7.7 *L'inégalité (7.31) définit une facette de $P(G, k, q)$ si et seulement si pour tout sommet $j \in W$, $|\delta(j)| \leq n - q - 1$ ($\Leftrightarrow \bar{N}'(j) \geq q$).*

Preuve. S'il existe un sommet $z \in W$ tel que $\bar{N}'(z) \leq q - 1$, alors la contrainte

$$\sum_{i \in V} x_i^t - \sum_{j \in V \setminus W} x_j^r + x_z^r \leq q, \tag{7.36}$$

est valide.

En effet, supposons qu'il existe $t \neq r \in \{1, \dots, k\}$ tel que la contrainte (7.36) associée à r et t n'est pas valide. Sans perte de généralité, on considère $t = 1$ et $r = 2$. Si (7.36) n'est pas valide, alors il existe $\bar{x} \in \{0, 1\}^{nk}$ solution de $P(G, k, q)$ ne vérifiant pas (7.36), c'est-à-dire

$$\sum_{i \in V} x_i^1 - \sum_{j \in V \setminus W} x_j^2 + x_z^2 \geq q + 1. \tag{7.37}$$

Comme \bar{x} satisfait

$$\sum_{i \in V} x_i^1 - \sum_{j \in V} x_j^2 \leq q, \tag{7.38}$$

$$- \sum_{i \in V} x_i^1 + \sum_{j \in V \setminus W} x_j^2 - x_z^2 \leq -q - 1. \tag{7.39}$$

On en déduit

$$\sum_{j \in W} x_j^2 + x_z^2 \geq 1. \tag{7.40}$$

Donc le sommet z ne peut pas être dans l'ensemble V_2 , car $\bar{N}'(z) \leq q - 1$. Dans ce cas $|V_1| < q$ et la contrainte (7.37) est violée.

Supposons que nous avons $q + 1$ sommets dans V_1 et le sommet z n'appartient pas à V_2 . Comme \bar{x} est solution, on doit avoir un sommet dans V_t , $t \in \{1, \dots, k\} \setminus \{1, 2\}$. Soit

$j \in W \setminus \{z\}$ dans V_2 . Comme on a au minimum $q + 1$ sommets dans V_1 et 1 sommet dans V_t , $t \in \{1, \dots, k\} \setminus \{1, 2\}$, $|\bar{N}'(j)| \geq q + k - 1$. De plus, les $k - 2$ sommets dans V_t , $t \in \{1, \dots, k\} \setminus \{1, 2\}$, forment un stable de cardinalité $k - 2$ dans $\bar{N}'(j)$. Il s'en suit que $j \notin W$ et donc une contradiction.

On en déduit que la contrainte (7.36) est valide. Et la contrainte (7.36) domine la contrainte (7.31). Ce qui implique que la contrainte (7.31) ne définit pas une facette.

Supposons maintenant que pour tout sommet $j \in V$, $|\delta(j)| \leq n - q - 1$. Notons par $ax \leq \alpha$ la contrainte (7.31). Soit $bx \leq \beta$ une inégalité qui définit une facette de $P(G, k, q)$. Supposons que $\{x \in P(G, k, q) : ax = \alpha\} \subseteq \{x \in P(G, k, q) : bx = \beta\}$. Puisque $P(G, k, q)$ est de pleine dimension, il suffit de montrer qu'il existe $\rho > 0$ tel que $b = \rho a$.

Étant donnés $i, j \in V$, on note par U_{ij} un sous-ensemble de sommets de V tel que $U_{ij} \subseteq V \setminus \{i, j\}$ et $|U_{ij}| = q - 1$. Considérons les partitions

- $\pi^i = (V_0^i, V_1^i, \dots, V_k^i)$ avec $V_0^i = V \setminus (\{i\} \cup U_{ij})$, $V_t^i = \{i\} \cup U_{ij}$, $V_s^i = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$,
- $\pi^j = (V_0^j, V_1^j, \dots, V_k^j)$ avec $V_0^j = V \setminus (\{j\} \cup U_{ij})$, $V_t^j = \{j\} \cup U_{ij}$, $V_s^j = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$.

Les partitions π^i et π^j sont des solutions. Comme les vecteurs d'incidence de π^i et π^j vérifient $ax = \alpha$, il s'ensuit que $bx^{\pi^i} = bx^{\pi^j}$. Par conséquent, $b(w_i^t) = b(w_j^t)$. On déduit alors

$$b(w_i^t) = b(w_j^t) = \rho, \quad \text{pour } i, j \in V, \quad (7.41)$$

et un scalaire ρ dans \mathbb{R} .

Soient $j \in V$ et $s \in \{1, \dots, k\} \setminus \{r, t\}$. Par hypothèse, il existe un sous-ensemble de sommets U_j de $\bar{N}'(j)$, $|U_j| = q$ et donc $N(j) \cap |U_j| = \emptyset$. Considérons les partitions

- $\pi_j^1 = (V_0^1, V_1^1, \dots, V_k^1)$ avec $V_0^1 = V \setminus U_j$, $V_t^1 = U_j$, $V_s^1 = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$,
- $\pi_j^2 = (V_0^2, V_1^2, \dots, V_k^2)$ avec $V_0^2 = V \setminus (\{j\} \cup U_j)$, $V_t^2 = U_j$, $V_s^2 = \{j\}$, $V_z^2 = \emptyset$ pour $z \in \{1, \dots, k\} \setminus \{t, s\}$,

qui sont des solutions. Comme les vecteurs d'incidence de π_j^1 et π_j^2 vérifient $ax = \alpha$, il s'ensuit que $bx^{\pi_j^1} = bx^{\pi_j^2}$. Par conséquent, $b(w_j^s) = 0$. Comme j est un élément quelconque de V , on déduit que

$$b(w_i^s) = 0, \quad \text{pour tout } i \in V \text{ et pour } s \in \{1, \dots, k\} \setminus \{r, t\}. \quad (7.42)$$

Étant donné $j \in W$, par hypothèse il existe un sous-ensemble de sommets U_j de $\bar{N}'(j)$, $|U_j| = q$ et donc $N(j) \cap |U_j| = \emptyset$, considérons les partitions

- $\pi_j^3 = (V_0^3, V_1^3, \dots, V_k^3)$ avec $V_0^3 = V \setminus U_j$, $V_t^3 = U_j$, $V_s^3 = \emptyset$ pour $s \in \{1, \dots, k\} \setminus \{t\}$,
- $\pi_j^4 = (V_0^4, V_1^4, \dots, V_k^4)$ avec $V_0^4 = V \setminus (\{j\} \cup U_j)$, $V_t^4 = U_j$, $V_r^4 = \{j\}$, $V_z^4 = \emptyset$ pour $z \in \{1, \dots, k\} \setminus \{t, r\}$.

Les partitions π_j^3 et π_j^4 sont des solutions. Comme les vecteurs d'incidence de π_j^3 et π_j^4 vérifient $ax = \alpha$, il s'ensuit que $bx^{\pi_j^3} = bx^{\pi_j^4}$. Par conséquent, $b(w_j^r) = 0$. Comme j est un élément quelconque de W , on déduit que

$$b(w_j^r) = 0, \quad \text{pour tout } j \in W. \quad (7.43)$$

Soient $j \in V \setminus W$ et $r \neq t \in \{1, \dots, k\}$, comme $j \notin W$ alors il existe un sous-ensemble de sommets \tilde{V}_j de $\bar{N}'(j)$, $|\tilde{V}_j| = q + k - 1$. De plus l'ensemble \tilde{V}_j contient un stable S de G tel que $|S| = k - 2$. On note les sommets de S par v_z tel que $z \in \{1, \dots, k\} \setminus \{t, r\}$. On note par \tilde{V}'_j un sous-ensemble de sommets de $\tilde{V}_j \setminus S$ tel que $|\tilde{V}'_j| = q$. Considérons les partitions

- $\pi_j^5 = (V_0^5, V_1^5, \dots, V_k^5)$ avec $V_0^5 = V \setminus (\tilde{V}'_j \cup S)$, $V_t^5 = \tilde{V}'_j$, $V_r^5 = \emptyset$, $V_s^5 = \{v_s\}$ pour $s \in \{1, \dots, k\} \setminus \{t, r\}$,
- $\pi_j^6 = (V_0^6, V_1^6, \dots, V_k^6)$ avec $V_0^6 = V \setminus (\{j\} \cup \tilde{V}_j)$, $V_t^6 = \tilde{V}_j \setminus S$, $V_r^6 = \{j\}$, $V_s^6 = \{v_s\}$ pour $s \in \{1, \dots, k\} \setminus \{t, r\}$,

sont des solutions avec $ax^{\pi_j^5} = \alpha$ et $ax^{\pi_j^6} = \alpha$. Alors $bx^{\pi_j^5} = bx^{\pi_j^6}$, par conséquent, $b(w_j^r) = -b(w_j^t)$. Comme j est un élément quelconque de $V \setminus W$, on déduit que

$$b(w_j^r) = -\rho, \quad \text{pour tout } j \in V \setminus W. \quad (7.44)$$

Par (7.41)-(7.44), il s'ensuit que $b = \rho a$. De plus, on peut montrer comme avant que $\rho > 0$ et la preuve est terminée. \square

7.5.2.5 Inégalités de dominants

Nous étudions ici une extension de contraintes proposées par Balas et Souza [17]. Dans un premier temps nous verrons les contraintes dits de dominants connexes, puis les contraintes dits de dominants non-connexes.

Un dominant dans un graphe $G = (V, E)$ est un sous-ensemble de sommets D de l'ensemble des sommets V tel que tout sommet qui n'appartient pas à D possède au moins une arête commune avec un sommet de D . En effet, un dominant, si sa cardinalité est supérieure à q , ne peut pas être inclus dans un des ensembles V_t , $t \in \{1, \dots, k\}$. Si un dominant est inclus dans l'ensemble V_t alors les ensembles V_r , $r \in \{1, \dots, k\} \setminus \{t\}$ sont vides.

Proposition 7.8 Soit D_W un dominant connexe du sous-graphe $G(W)$ où $W \subseteq V$. Si $|D_W| > q$ et $|W| > n - (k-1)|D_W| + (k-1)q$ alors la contrainte

$$\sum_{i \in D_W} \sum_{t=1}^k x_i^t \leq |D_W| - 1, \quad (7.45)$$

est valide.

Preuve. Soient $W \subseteq V$, D_W un dominant connexe de $G(W)$. Si $D_W \subseteq V_t$, où $t \in \{1, \dots, k\}$, alors il existe au plus $|V \setminus W|$ sommets pouvant appartenir aux ensembles V_r pour $r \in \{1, \dots, k\} \setminus \{t\}$. Comme la solution doit être équilibrée, il est donc possible d'avoir $D_W \subseteq V_t$ si

$$\frac{n - |W|}{k-1} \geq |D_W| - q,$$

ou d'une manière équivalente

$$|W| \leq n - (k-1)|D_W| + (k-1)q.$$

Donc si $|W| > n - (k-1)|D_W| + (k-1)q$ tous les sommets de $|D_W|$ ne peuvent pas appartenir au même ensemble V_t pour $t \in \{1, \dots, k\}$. De plus comme D_W est connexe il n'est donc pas possible d'avoir $|D_W \cap V_t| > 0$ et $|D_W \cap V_r| > 0$ pour $t \in \{1, \dots, k\}$ et $r \in \{1, \dots, k\} \setminus \{t\}$. On en déduit que la contrainte (7.45) est valide. \square

Ces contraintes sont appelées *inégalités de dominants connexes*.

Proposition 7.9 Soient D_W et D'_W deux dominants connexes de $G(W)$ où $W \subseteq V$ tels que $|D_W| > |D'_W|$, $|D'_W| > q$ et $|W| > n - (k-1)|D_W| + (k-1)q$. Alors la contrainte (7.45) associée à D'_W domine celle associée à D_W .

Preuve. Les contraintes de dominants connexes associées à D_W et D'_W sont respectivement données par

$$\sum_{i \in D'_W} \sum_{t=1}^k x_i^t + \sum_{i \in D_W \setminus D'_W} \sum_{t=1}^k x_i^t \leq |D'_W| - 1 + |D_W \setminus D'_W|, \quad (7.46)$$

$$\sum_{i \in D'_W} \sum_{t=1}^k x_i^t \leq |D'_W| - 1. \quad (7.47)$$

Il est clair que la contrainte (7.46) est combinaison linéaire des contraintes (7.47) et de l'inégalité de partition (7.16) associée à i pour tout $i \in D_W \setminus D'_W$. \square

Proposition 7.10 Soit D_W un dominant non-connexe du sous-graphe $G(W)$ où $W \subseteq V$. Si $|D_W| > q$ et $|W| > n - (k - 1)|D_W| + (k - 1)q$, alors les contraintes

$$\sum_{i \in D_W} x_i^t \leq |D_W| - 1, \quad \text{pour tout } t = 1, \dots, k, \quad (7.48)$$

sont valides.

Preuve. La preuve est similaire à la preuve de la proposition 7.8, sauf qu'ici comme le dominant n'est pas connexe, on considère une contrainte par ensemble de la partition. \square

Ces contraintes sont appelées *inégalités de dominants non-connexes*.

Proposition 7.11 Soient D_W et D'_W deux dominants non connexes de $G(W)$ où $W \subseteq V$ tel que $|D_W| > |D'_W|$, $|D'_W| > q$ et $|W| > n - (k - 1)|D_W| + (k - 1)q$. Alors la contrainte (7.45) associée à D'_W domine celle associée à D_W .

Preuve. La preuve est similaire à celle de la proposition 7.9. \square

Nous montrons dans la suite qu'il existe une contrainte qui domine la contrainte (7.48).

Proposition 7.12 Soit D_W un dominant non connexe de $G(W)$ où $W \subseteq V$ tel que $|D_W| > q$ et $|W| > n - (k - 1)|D_W| + (k - 1)q$. Pour tout sommet j , où $j \in D_W$ tel qu'il existe une arête ij dans $E(D_W)$, les contraintes

$$\sum_{i \in D_W} x_i^t + \sum_{r \in \{1, \dots, k\} \setminus \{t\}} x_j^r \leq |D_W| - 1, \quad \text{pour tout } t \in \{1, \dots, k\}, \quad (7.49)$$

sont valides.

Preuve. Supposons qu'il existe une solution π telle que x^π vérifie la contrainte

$$\sum_{i \in D_W} x_i^{\pi t} + \sum_{r \in \{1, \dots, k\} \setminus \{t\}} x_j^{\pi r} \geq |D_W|. \quad (7.50)$$

Il est clair que $\sum_{r \in \{1, \dots, k\} \setminus \{t\}} x_j^{\pi r} \leq 1$. En effet il n'est pas possible pour un sommet j d'appartenir à deux sous-ensembles. Comme $|D_W| > q$, il n'est pas possible de prendre

l'ensemble des sommets de D_W dans l'ensemble V_t . De plus il n'est pas possible de prendre le sommet j dans un ensemble V_r , tel que $r \in \{1, \dots, k\} \setminus \{t\}$, et les sommets $D_W \setminus \{j\}$ dans l'ensemble V_t , puisqu'il existe une arête ij . On en déduit qu'il ne peut pas exister de solution π telle que x^π vérifie la contrainte (7.50). Donc la contrainte (7.49) est valide. \square

On observe que la contrainte (7.49) domine (7.48).

Proposition 7.13 *Soit D_W un dominant non connexe de $G(W)$ où $W \subseteq V$ tel que $|D_W| > q$ et $|W| > n - (k-1)|D_W| + (k-1)q$. Pour tout sommet j , où $j \notin D_W$ tel qu'il existe les arêtes ij et zj dans E avec $i, z \in D_W$, les contraintes*

$$\sum_{i \in D_W} x_i^t + \sum_{r \in \{1, \dots, k\} \setminus \{t\}} x_j^r \leq |D_W| - 1, \quad \text{pour tout } t \in \{1, \dots, k\}, \quad (7.51)$$

sont valides.

Preuve. Supposons qu'il existe une solution π de $P(G, k, q)$ telle que x^π vérifie la contrainte

$$\sum_{i \in D_W} x_i^{\pi t} + \sum_{r \in \{1, \dots, k\} \setminus \{t\}} x_j^{\pi r} \geq |D_W|. \quad (7.52)$$

Il est clair que $\sum_{r \in \{1, \dots, k\} \setminus \{t\}} x_j^{\pi r} \leq 1$. En effet il n'est pas possible pour un sommet j d'appartenir à deux sous-ensembles. Comme $|D_W| > q$, il n'est pas possible de prendre l'ensemble des sommets de D_W dans l'ensemble V_t . De plus il n'est pas possible de prendre le sommet j dans un ensemble V_r , tel que $r \in \{1, \dots, k\} \setminus \{t\}$, et un ensemble $D'_W \subset D_W$, tel que $|D'_W| = |D_W| - 1$, dans l'ensemble V_t , puisque D'_W contient soit le sommet i soit le sommet z . On en déduit qu'il ne peut pas exister de solution π tel que x^π vérifie la contrainte (7.52). Donc la contrainte (7.51) est valide. \square

on observe que la contrainte (7.51) domine la contrainte (7.48).

7.5.2.6 Inégalités de borne

Dans [18] Didi Biha et Meurs ont proposé une contrainte valide pour $k = 2$, basée sur les chemins sommets disjoints. Nous allons dans un premier temps présenter leur

contrainte, puis nous indiquerons comment cette contrainte peut être étendue pour notre problème.

Soient i et j deux sommets non adjacents. On note par α_{ij} le nombre maximum de chemins sommets disjoints entre i et j .

$$\alpha_{min} = \min\{\alpha_{ij} : i, j \in V, ij \notin E\}.$$

α_{min} représente le nombre minimum de sommets à supprimer pour déconnecter le graphe en deux sous-ensembles non vides. On en déduit que α_{min} est une borne inférieure sur la taille de l'ensemble V_0 .

Didi Biha et Meurs ont proposé la contrainte suivante

$$\sum_{i \in V} x_i^1 + \sum_{i \in V} x_i^2 \leq |V| - \alpha_{min}, \tag{7.53}$$

pour le problème du 2-séparateur bornés.

On en déduit que la contrainte

$$\sum_{t \in \{1, \dots, k\}} \sum_{i \in V} x_i^t \leq |V| - \alpha_{min}, \tag{7.54}$$

est valide pour le problème du k -séparateur équilibré si et seulement si $\alpha_{min} \leq n - q$.

En effet, si $\alpha_{min} = n - q + 1$ alors la contrainte (7.54) s'écrit de la manière suivante

$$\sum_{t \in \{1, \dots, k\}} \sum_{i \in V} x_i^t \leq q - 1, \tag{7.55}$$

et la solution $\pi^t = (V_0^t, V_1^t, \dots, V_k^t)$ avec $|V_t^t| = q$, $|V_0^t| = n - q$, $|V_s^t| = 0$ pour $s \in \{1, \dots, k\} \setminus \{t\}$ et $t \in \{1, \dots, k\}$ ne vérifie pas (7.55). Par contre, si $\alpha_{min} \leq n - q$ alors il est nécessaire de déconnecter le graphe en deux sous-ensembles non vides. En effet, dans ce cas nous pouvons avoir une solution $\pi^t = (V_0^t, V_1^t, \dots, V_k^t)$ avec $|V_t^t| = q$, $|V_0^t| = n - q$, $|V_s^t| = 0$ pour $s \in \{1, \dots, k\} \setminus \{t\}$.

Nous présentons dans la section suivante une étude expérimentale basée sur les résultats présentés ci-dessus.

7.6 Étude expérimentale

Dans un premier temps nous comparons des formulations approchées et la formulation exacte. Nous avons testé ces formulations, sur une instance réelle de 150 sommets. Nous avons considéré le problème du k -séparateur pour $k = 2, 3, 4$. Nous avons utilisé CPLEX 11.0, sur un Pentium core 2 duo 2,66 GHz avec 2Go RAM, pour résoudre les programmes linéaires en nombres entiers. Pour l'ensemble des tests effectués avec la formulation exacte nous avons fixé $q = 1$.

Les entrées des tableaux qui suivent sont :

Formulation	:	indique quelle formulation est utilisée,
k	:	nombre d'ensembles deux à deux déconnectés,
$ V_0 $:	le nombre de variables dans l'interface,
$ V_i $:	le nombre de variables dans l'ensemble V_i de la partition,
TT	:	le temps de résolution.

Nous notons les deux formulations approchées respectivement par F_1 et F_2 et la formulation exacte, avec $q = 1$, par F_3 . Nous donnons aussi les résultats fournis par Métis le logiciel utilisé pour le moment par LMS-Imagine pour résoudre le problème du k -séparateur équilibré.

La Table 7.1 rassemble les résultats avec $k \in \{2, 3, 4\}$. Les solutions données par nos trois formulations sont identiques pour $k = 2$ et $k = 3$. Les ensembles V_1 et V_2 pour $k = 2$ et les ensembles V_1, V_2 et V_3 pour $k = 3$ sont équilibrés. Alors que pour $k = 4$, la formulation F_1 donne un équilibrage parfait. La formulation F_2 donne une solution loin de l'équilibrage, la formulation F_3 donne un bon équilibrage et une meilleure solution, sous les critères considérés, que la formulation F_1 .

Nous remarquons aussi que pour l'ensemble des valeurs de k , la solution donnée par Métis est toujours très loin de l'optimum. En effet, pour $k = 4$ nous avons $|V_0| = 101$ avec Métis et seulement $|V_0| = 28$ avec la formulation exacte. La solution est donc grandement améliorée par la résolution exacte. Ces résultats expérimentaux montrent donc que l'approche exacte est plus efficace que celle élaborée dans Métis [5]. En effet, cette approche donne des solutions qui sont optimales en fonctions des critères considérés, alors que Métis propose des solutions heuristiques n'ayant aucune garantie de qualité.

Formulation	k	$ V_0 $	$ V_1 $	$ V_2 $	$ V_3 $	$ V_4 $	TT
F_1	2	18	66	66	-	-	2,87
F_2	2	18	66	66	-	-	2,84
F_3	2	18	66	66	-	-	2,3
Métis	2	78	37	35	-	-	< 1seconde
F_1	3	24	42	42	42	-	299,65
F_2	3	24	42	42	42	-	421,2
F_3	3	24	42	42	42	-	421,43
Métis	3	88	21	16	25	-	< 1seconde
F_1	4	30	30	30	30	30	> 4heures
F_2	4	25	28	31	33	33	> 4heures
F_3	4	28	31	30	31	30	> 4heures
Métis	4	101	9	8	13	19	< 1seconde

TABLE 7.1 – Comparaisons des formulations approchées, formulation exacte et des résultats de Métis.

Dans la suite, nous présentons des résultats expérimentaux obtenus par le problème du k -séparateur pour plusieurs instances réelles et aléatoires. D’abord nous décrivons les instances testées.

7.6.1 Description des instances

Les instances testées sont deux instances réelles qui possèdent respectivement 35 et 150 sommets. Ces instances notées respectivement, par R-35 et R-150, représentent des cas tests du projet PARADE. Nous avons aussi utilisé dix instances aléatoires. Les instances aléatoires que nous avons considérées proviennent du challenge DIMACS et peuvent être téléchargées depuis [3]. Ces instances données dans la table 7.2, possèdent entre 11 et 191 sommets. On note par n le nombre de sommets et par m le nombre d’arêtes.

Nous avons développé une heuristique primale pour obtenir puis améliorer au cours du temps, une borne supérieure réalisable pour le problème. Cette heuristique est présentée dans la section suivante.

Instance	n	m
david	87	406
games120	120	638
miles750	128	2113
miles1000	128	3216
myciel3	11	20
myciel4	23	71
myciel5	47	236
myciel6	95	755
myciel7	191	2360
queen7-7	49	476

TABLE 7.2 – Récapitulatif des instances aléatoires.

7.6.2 Heuristique primale

Nous présentons, dans cette section, une extension de l'heuristique primale proposée par Balas et Souza dans [17]. Lors de l'exécution de cette heuristique, on essaye de transformer chaque solution de la relaxation linéaire en une solution réalisable (donc entière) pour le problème. Pour la formulation P_k , cette heuristique consiste à trouver un k -séparateur équilibré en utilisant un algorithme glouton. Notons par $x^* = (x_1^*, \dots, x_k^*)$ la solution optimale de la relaxation linéaire. Nous allons construire un k -séparateur équilibré $\pi^* = (V_0^*, V_1^*, \dots, V_k^*)$ à partir de la solution x^* . On part du k -séparateur équilibré $\bar{\pi} = (\bar{V}_0, \bar{V}_1, \dots, \bar{V}_k)$ où $\bar{V}_0 = V$ et $\bar{V}_i = \emptyset$, pour tout $i = 1, \dots, k$. L'heuristique améliore cette solution en tentant pour chaque $r \in \{1, \dots, k\}$ de supprimer un sommet v de \bar{V}_0 pour le mettre dans \bar{V}_r . Ce déplacement de v de \bar{V}_0 à \bar{V}_r n'est effectué que si la nouvelle partition est un k -séparateur équilibré. A chaque itération et pour tout $r \in \{1, \dots, k\}$, le sommet v choisi dans \bar{V}_0 est le sommet $v \in \bar{V}_0$ tel que \bar{x}_v^r est maximum et la nouvelle partition est un k -séparateur équilibré. L'algorithme 4 illustre notre heuristique primale. L'algorithme ChercheSommetMax détermine le sommet v de \bar{V}_0 tel que la nouvelle partition, obtenue en déplaçant v de \bar{V}_0 à \bar{V}_r est un k -séparateur équilibré et tel que \bar{x}_v^r est maximum. Sa complexité est en $O(n\Delta_G)$ où Δ_G représente le degré maximum du graphe G , c'est-à-dire $\Delta_G = \max_{v \in V} (|\delta(v)|)$. Tester si la nouvelle partition, disons π' , est un k -séparateur équilibré se fait en $O(\Delta_G)$. En effet, pour tester si π' est un k -séparateur, il suffit de tester que tous les voisins de v appartiennent à $\bar{V}_0 \cup \bar{V}_r$. La complexité au pire cas de l'algorithme s'explique par le fait que \bar{V}_0 contient au plus n sommets. La complexité de l'heuristique est en $O(n^2\Delta_G)$. En effet, la complexité de la première boucle est en $O(kn\Delta_G)$. Il faut ensuite remarquer que la seconde

boucle s'exécute au plus $\frac{n-q}{k} + q$ fois. En effet, à partir de la première itération où on n'ajoute pas un sommet pour tout \bar{V}_r , $r \in \{1, \dots, k\}$, au plus q itérations supplémentaires peuvent être faites. Ceci s'explique par le fait que si pour un $r \in \{1, \dots, k\}$ quelconque, on ne peut pas ajouter un sommet dans \bar{V}_r , alors aucun nouveau sommet ne pourra être ajouté à \bar{V}_r par la suite puisque l'on ne supprime pas de sommet dans un ensemble \bar{V}_i , $i \in \{1, \dots, k\}$ durant l'heuristique. Puisque le k -séparateur doit être équilibré, on ne peut pas ajouter plus de q sommets dans les autres éléments de la partition. Cette heuristique peut être implémenté en un temps $O(n^2\Delta_G)$.

Algorithme 4: Heuristique primale

Entrées : - La solution optimale de la relaxation linéaire $x^* = (x_1^*, \dots, x_k^*)$.

Sorties : - Un k -séparateur équilibré.

début

$\bar{\pi} = (\bar{V}_0, \bar{V}_1, \dots, \bar{V}_k)$, avec $\bar{V}_0 = V$ et $\bar{V}_r = \emptyset$ pour $r \in \{1, \dots, k\}$;

 modifie \leftarrow vrai;

tant que *modifie* = vrai **faire**

 modifie \leftarrow faux;

pour chaque $r \in \{1, \dots, k\}$ **faire**

$v = \text{ChercheSommetMax}(\bar{\pi}, r, \bar{x})$;

si $v \neq$ **alors**

$\bar{V}_r = \bar{V}_r \cup \{v\}$;

$\bar{V}_0 = \bar{V}_0 \setminus \{v\}$;

 modifie \leftarrow vrai;

fin

fin

fin

retourner $\bar{\pi}$;

fin

Nous avons utilisé cette heuristique dans notre algorithme de coupes et branchements.

Nous discutons dans la prochaine section une procédure de séparation pour les inégalités de dominants connexes.

7.6.3 Séparation des inégalités de dominants connexes

Nous présentons une extension de l'heuristique de séparation des contraintes de dominants connexes proposée par Balas et Souza dans [17]. Les contraintes de dominants

Algorithme 5: ChercheSommetMax

Entrées : - Un k -séparateur équilibré $\bar{\pi}$, un entier r et un vecteur \bar{x} .

Sorties : - Un sommet v .

début

$S = \emptyset;$

$coutS = 0;$

pour chaque $v \in \bar{V}_0$ **faire**

$\pi' = (V'_0, V'_1, \dots, V'_k)$, avec $V'_0 = \bar{V}_0 \setminus \{v\}$, $V'_r = \bar{V}_r \cup \{v\}$ et $V'_i = \bar{V}_i$ pour $i \in \{1, \dots, k\} \setminus \{r\}$;

si π' *est un k -séparateur équilibré et $coutS < \bar{x}_v^r$* **alors**

$S = v;$

$coutS = \bar{x}_v^r;$

fin

fin

retourner S ;

fin

connexes sont en nombres exponentiels. Afin de pouvoir les utiliser dans un algorithme de coupes et branchements, il est nécessaire d'étudier le problème de séparation de ces contraintes. Ce problème peut être défini comme suit. Rappelons que x^* est la solution courante à séparer. Existe-t-il un dominant D_W tel que la contrainte de dominants connexes associée à D_W est violée par x^* ?

Posons $x_i^0 = 1 - \sum_{t=1}^k x_i^{t*}$. Les contraintes de dominants connexes (7.45) pour x^* sont équivalentes à

$$\sum_{i \in D_W} x_i^0 \geq 1. \quad (7.56)$$

La contrainte (7.57) est violée par la solution x^* si et seulement si

$$\sum_{i \in D_W} x_i^0 < 1. \quad (7.57)$$

L'algorithme que nous proposons cherche un dominant connexe D de $G(W)$ où $W \subseteq V$, $|W| > n - (k-1)|D| + (k-1)q$ et $|D| > q$. Pour cela, on ajoute à chaque étape, dans D , le sommet $i \in N(D)$ tel que le ratio $\frac{x_i^0}{N(D \cup i)}$ soit minimum. On ajoute des sommets à D tant que $|D| + |N(D)| \leq |V| - (k-1) * |D| + (k-1) * q$. L'algorithme 6 illustre notre algorithme de séparation où \mathcal{C} représente un ensemble de dominants connexes tels que leurs contraintes associées sont violées par x^* . La complexité de cette heuristique est

$O(n(n+m)\log n)$. En effet, nous avons la même complexité que l'heuristique proposée par Balas et Souza [17].

Algorithme 6: Heuristique de séparation des contraintes de dominants connexes

Entrées : - La solution optimale de la relaxation linéaire $x^* = (x_1^*, \dots, x_k^*)$.

Sorties : - Ensemble \mathcal{C} de contraintes de dominants connexes violées par x^*

début

$S = \emptyset$;

 Pour tout $i \in V$ on pose $x_i^0 = 1 - \sum_{t=1}^k x_i^{t*}$;

pour chaque $i \in V$ **faire**

$D = \{i\}$;

tant que $|D| + |N(D)| \leq |V| - (k-1) * |D| + (k-1) * q$ **faire**

 Trouver le sommet $j \in N(D)$ tel que $\frac{x_j^0}{N(D \cup j)}$ est minimum;

 Ajouter j dans D ;

fin

si la somme des x_j^0 pour $j \in D$ est inférieure à 1 et que $|D| > q$ **alors**

 ajouter, dans S , la contrainte associée au dominant D ;

fin

fin

 retourner S ;

fin

7.6.4 Algorithme de coupes et branchements

Nous avons utilisé les résultats développés précédemment dans le cadre d'un algorithme de coupes et branchements pour le problème du k -séparateur équilibré. L'algorithme commence par résoudre la relaxation linéaire du problème qui est compacte (le nombre de contraintes et de variables est polynomial) c'est-à-dire, *le programme linéaire initial* donné par

$$\begin{aligned} \max \sum_{i \in V} \sum_{t=1}^k x_i^t \\ \sum_{i \in V} x_i^t - \sum_{i \in V} x_i^r \leq q \end{aligned} \quad \text{pour tout } t \neq r \in \{1, \dots, k\}, \quad (7.58)$$

$$\sum_{t=1}^k x_i^t \leq 1, \quad \text{pour tout } i \in V, \quad (7.59)$$

$$x_i^z + \sum_{t \in \{1, \dots, k\} \setminus \{z\}} x_j^t \leq 1 \quad \text{pour tout } z \leq k \text{ et pour tout } ij \in E, \quad (7.60)$$

$$\sum_{t \in \{1, \dots, k\}} \sum_{i \in V} x_i^t \leq |V| - \alpha_{min}, \quad (7.61)$$

$$x_i^t \geq 0 \quad \text{pour tout } i \in V, t \in \{1, \dots, k\}, \quad (7.62)$$

$$x_i^t \leq 1 \quad \text{pour tout } i \in V, t \in \{1, \dots, k\}, \quad (7.63)$$

$$(7.64)$$

La solution optimale $x^* \in \mathbb{R}^{|\mathcal{E}|}$ de cette relaxation est réalisable pour le problème si x^* est un vecteur entier. En général, la solution x^* n'est pas réalisable. Dans ce cas, à chaque itération, l'algorithme génère des inégalités valides pour le problème. Les inégalités ainsi trouvées sont ajoutées au programme linéaire courant et le nouveau programme est résolu. Ce processus est repris jusqu'à ce qu'aucune inégalité violée ne soit détectée. Si la solution n'est toujours pas réalisable, l'algorithme procède alors à un branchement. L'algorithme 7 ci-dessous illustre les principales phases de notre algorithme de coupes et branchements.

Les contraintes de dominants connexes sont séparées en utilisant l'heuristique définie dans la section 7.6.3.

On peut remarquer que toutes les inégalités sont globales (*i.e.* valides dans tout l'arbre de branchements). Par ailleurs, plusieurs contraintes peuvent être ajoutées à chaque itération.

7.6.5 Résultats expérimentaux

Nous avons considéré le problème du k -séparateur équilibré pour $k \in \{2, 3, 4\}$ et $q \in \{5, 10\}$. Nous avons utilisé JAVA avec la bibliothèque Concert Technology qui est couplée avec le solveur linéaire CPLEX 12.1 [41], sur un Xeon biprocesseur quad core

Algorithme 7: Algorithme de coupes et branchements

Entrées : un graphe $G = (V, E)$, un entier k et un entier q **Sorties :** Solution optimale de (P_k) **début**1 : PL \leftarrow Programme linéaire initial

2 : Résoudre le programme linéaire PL.

Soit x la solution optimale de PL.3 : **si** x est réalisable pour $P(G, k, q)$ **alors**| x est une solution optimale. STOP**fin**4 : **si** des contraintes de couplages de dominants connexes violées par x sont trouvées **alors**

| Les ajouter à PL.

| Aller en 2.

fin5 : **sinon**

| Brancher sur une variable fractionnaire.

fin

6 : Prendre la meilleure solution de tous les sous-problèmes.

fin

2,27 GHz avec 16Go RAM, pour effectuer nos calculs et implémenter notre algorithme de coupes et branchements. Dans cette étude nous avons limité le temps maximum de calcul à cinq heures.

Les entrées des tableaux qui suivent sont :

<i>Instance</i>	: nom de l'instance,
<i>k</i>	: nombre d'ensembles deux à deux déconnectés,
<i>q</i>	: différence maximum entre deux éléments de $\{ V_1 , \dots, V_k \}$,
<i>TT</i>	: temps de résolution en secondes,
<i>No</i>	: nombre de nœuds dans l'arbre de branchements,
<i>Gap</i>	: l'erreur relative entre la meilleure borne supérieure et la meilleure borne inférieure (si le problème est résolu à l'optimum alors le gap vaut 0),
<i>Opt</i>	: Optimal si le système P_k est résolu à l'optimum et Feasible autrement,
<i>TT$_{\alpha}$</i>	: temps de résolution en secondes en rajoutant la contrainte de borne à la formulation P_k ,
<i>No$_{\alpha}$</i>	: nombre de nœuds dans l'arbre de branchements en rajoutant la contrainte de borne à la formulation P_k ,
<i>Gap$_{\alpha}$</i>	: l'erreur relative entre la meilleur borne supérieure et la meilleure borne inférieure en rajoutant la contrainte de borne à la formulation P_k ,
<i>Opt$_{\alpha}$</i>	: Optimal si le système P_k avec la contrainte de borne est résolu à l'optimum et Feasible autrement.
<i>TT$_{dom}$</i>	: temps de résolution en secondes en rajoutant la contrainte de borne et les contraintes de dominants connexes à la formulation P_k ,
<i>No$_{dom}$</i>	: nombre de nœuds dans l'arbre de branchements en rajoutant la contrainte de borne et les contraintes de dominants connexes à la formulation P_k ,
<i>Gap$_{dom}$</i>	: l'erreur relative entre la meilleur borne supérieure et la meilleure borne inférieure en rajoutant la contrainte de borne et les contraintes de dominants connexes à la formulation P_k ,
<i>Opt$_{dom}$</i>	: Optimal si le système P_k avec la contrainte de borne et les contraintes de dominants connexes est résolu à l'optimum et Feasible autrement.
<i>N$_{DC}$</i>	: nombre de contraintes de dominants connexes générées,

Dans un premier temps nous avons considéré la formulation renforcée P_r du problème du k -séparateur équilibré donnée par le programme suivant :

$$\max \left\{ \sum_{i \in V} \sum_{t=1}^k x_i^t \mid x_i^t \in \{0, 1\}^{nk} : x_i^t \text{ satisfait (7.15), (7.16), (7.25)} \right\}$$

Remarquons que nous n'avons pas utilisé les contraintes (7.31) car les ensembles W associés à ces contraintes sont toujours vides sauf pour la plus petite instance myciel3 qui est toujours résolue à l'optimum en moins d'une seconde.

Les tables 7.3 et 7.4 donnent les résultats expérimentaux obtenus par CPLEX. On remarque que pour $k = 2$ toutes les instances sont résolues à l'optimum. Alors que pour $k = 3$ vingt-deux instances sur vingt-quatre sont résolues à l'optimum et pour $k = 4$ seulement dix-huit sont résolues à l'optimum dans le temps imparti. Il est toujours plus difficile, sur les instances testées, de résoudre le problème avec $k = 4$.

Nous avons testé la même formulation en ajoutant la contrainte de borne (7.54).

Le tableau 7.5 résume les temps, en secondes, nécessaires pour trouver les α_{min} , ainsi que la valeur de α_{min} pour chaque instance.

Les tables 7.6 et 7.7 donnent les résultats expérimentaux obtenus par CPLEX lorsqu'on ajoute la contrainte de borne. Nous n'avons pas considéré les instances réelles. En effet, comme $\alpha_{min} = 0$ dans ce cas, la contrainte de borne est combinaison linéaire des contraintes (7.16). On a $\alpha_{min} = 0$ car le graphe associé à chaque instance réelle n'est pas connexe. En effet, dans les deux instances il existe au moins un sommet déconnecté du reste du graphe.

Nous remarquons que nous n'arrivons pas à résoudre plus d'instances en utilisant la contrainte de borne. Par contre le temps de résolution, pour plusieurs instances, est beaucoup plus faible. Pour six instances nous gagnons plus de 1500 secondes. Il arrive pour certaines instances que le temps de résolution soit plus long mais cela ne dépasse jamais 400 secondes. La contrainte de borne est, en moyenne, efficace pour améliorer les temps de résolution. De plus, la relaxation linéaire de P_r donne toujours une solution égale au nombre de sommets. En effet, la solution x^* , où $x_i^{*r} = \frac{1}{k}$ pour $i = 1, \dots, n$ et $r = 1, \dots, k$, est toujours valide et elle sature les contraintes (7.16). On en déduit que la solution x^* est optimale pour la relaxation linéaire de P_r . La contrainte de borne permet de renforcer la relaxation linéaire de P_r , puisque dans ce cas la solution de la relaxation linéaire est égale au nombre de sommets moins α_{min} . Nous avons vu que la contrainte de borne permet, en moyenne, d'améliorer les performances. En effet, nous avons gagné, sur la résolution de l'ensemble des instances, plus de 27.000 secondes, soit à peu près 7heures et 30minutes. Ce qui représente plus de 5% de gain de temps sur l'ensemble des instances.

Les tables 7.8 et 7.9 donnent les résultats expérimentaux obtenus par notre algorithme de coupes et branchements.

Instance	k	q	TT	No	Gap	Opt
R-35	2	5	0,2	17	0.0	Optimal
R-35	2	10	0,1	10	0.0	Optimal
R-35	3	5	0,9	13	0.0	Optimal
R-35	3	10	0,8	9	0.0	Optimal
R-35	4	5	2,3	16	0.0	Optimal
R-35	4	10	1,8	9	0.0	Optimal
R-150	2	5	0,9	41	0,0	Optimal
R-150	2	10	0,9	52	0,0	Optimal
R-150	3	5	79,9	523	0,0	Optimal
R-150	3	10	111,5	561	0,0	Optimal
R-150	4	5	1246,1	2026	0,0	Optimal
R-150	4	10	148,2	423	0,0	Optimal
david	2	5	0,2	82	0,0	Optimal
david	2	10	0,2	77	0,0	Optimal
david	3	5	2,6	185	0,0	Optimal
david	3	10	1,5	70	0,0	Optimal
david	4	5	49,3	3749	0,0	Optimal
david	4	10	4,2	61	0,0	Optimal
games120	2	5	142,9	243491	0,0	Optimal
games120	2	10	164,5	307833	0,0	Optimal
games120	3	5	15479,0	1888714	0,0	Optimal
games120	3	10	17819,0	2276893	0,0	Optimal
games120	4	5	18000	866003	0,2	Feasible
games120	4	10	18000	752502	0,2	Feasible
miles750	2	5	1,5	116	0,0	Optimal
miles750	2	10	1,6	218	0,0	Optimal
miles750	3	5	126,5	546	0,0	Optimal
miles750	3	10	80,9	368	0,0	Optimal
miles750	4	5	12974,3	42494	0,0	Optimal
miles750	4	10	7804,5	21150	0,0	Optimal
miles1000	2	5	2,4	111	0,0	Optimal
miles1000	2	10	4,7	181	0,0	Optimal
miles1000	3	5	1840,6	6350	0,0	Optimal
miles1000	3	10	1906,7	5268	0,0	Optimal
miles1000	4	5	18000	47575	0,4	Feasible
miles1000	4	10	18000	45336	0,5	Feasible

TABLE 7.3 – Résultat du P_r (partie 1).

Instance	k	q	TT	No	Gap	Opt
myciel3	2	5	0,0	7	0,0	Optimal
myciel3	2	10	0,0	0	0,0	Optimal
myciel3	3	5	0,0	15	0,0	Optimal
myciel3	3	10	0,0	3	0,0	Optimal
myciel3	4	5	0,0	11	0,0	Optimal
myciel3	4	10	0,0	3	0,0	Optimal
myciel4	2	5	0,1	21	0,0	Optimal
myciel4	2	10	0,1	27	0,0	Optimal
myciel4	3	5	0,1	211	0,0	Optimal
myciel4	3	10	0,1	48	0,0	Optimal
myciel4	4	5	0,2	96	0,0	Optimal
myciel4	4	10	0,2	22	0,0	Optimal
myciel5	2	5	0,1	136	0,0	Optimal
myciel5	2	10	0,1	123	0,0	Optimal
myciel5	3	5	1,1	218	0,0	Optimal
myciel5	3	10	1,0	597	0,0	Optimal
myciel5	4	5	2,3	516	0,0	Optimal
myciel5	4	10	34,1	13225	0,0	Optimal
myciel6	2	5	8,3	8205	0,0	Optimal
myciel6	2	10	6,9	6532	0,0	Optimal
myciel6	3	5	125,1	16062	0,0	Optimal
myciel6	3	10	199,5	27024	0,0	Optimal
myciel6	4	5	6279,3	492033	0,0	Optimal
myciel6	4	10	624,5	29135	0,0	Optimal
myciel7	2	5	1605,7	568520	0,0	Optimal
myciel7	2	10	4809,9	2160200	0,0	Optimal
myciel7	3	5	18000	247478	0,2	Feasible
myciel7	3	10	18000	247196	0,1	Feasible
myciel7	4	5	18000	74163	0,1	Feasible
myciel7	4	10	18000	79618	0,1	Feasible
queen7-7	2	5	2,0	6239	0,0	Optimal
queen7-7	2	10	1,8	4631	0,0	Optimal
queen7-7	3	5	18,9	11995	0,0	Optimal
queen7-7	3	10	22,1	11632	0,0	Optimal
queen7-7	4	5	55,4	19661	0,0	Optimal
queen7-7	4	10	59,3	12466	0,0	Optimal

TABLE 7.4 – Résultat du P_r (partie 2).

Instance	α_{min}	T_2
R-35	0	0
R-150	0	1
david	1	51
games120	2	116
miles750	6	273
miles1000	11	316
myciel3	3	0
myciel4	4	0
myciel5	5	4
myciel6	6	64
myciel7	7	1041
queen7-7	18	2

TABLE 7.5 – Récapitulatif des valeurs α_{min} .

Nous remarquons que nous avons beaucoup moins de nœuds dans l'arbre de branchement pour l'instance réelle R-150 avec $k > 2$. De plus, nous mettons moins de temps pour l'instance R-150 avec $k = 4$ et $q = 5$ et l'instance myciel6 avec $k = 4$ et $q = 5$. Nous générons des contraintes de dominants connexes uniquement dans 39 instances sur 72. Nous remarquons aussi que nous résolvons moins d'instances à l'optimum en moins de 5 heures. En effet, le temps passé dans la séparation est très important, en moyenne plus de 85% du temps est passé dans la génération des contraintes de dominants connexes. Ces résultats préliminaires montrent que les contraintes de dominants connexes permettent de réduire le nombre de nœud dans l'arbre de branchement quand $k = 4$ sur les instances R-150, david, myciel6 et queen7-7. Il serait donc très intéressant de générer ces contraintes plus rapidement et, si possible, en générer plus en utilisant des heuristiques mieux adaptées.

Il faut noter que les résultats peuvent changer, notamment sur le nombre de nœuds dans l'arbre de branchement, entre la résolution du P_r avec contrainte de borne et la résolution du P_r avec contrainte de borne et contraintes de dominants connexes. En effet, afin d'implémenter la séparation des contraintes dans la bibliothèque Concert Technology il est nécessaire de supprimer quelques routines de CPLEX.

Les prochaines étapes de cette étude préliminaire peuvent se résumer en trois grandes étapes. Dans un premier temps faire l'étude polyédrale des contraintes de dominants connexes afin d'adapter, au mieux, l'heuristique de séparation de ces contraintes. En-

suite il faudrait étudier les contraintes de dominants non-connexes pour les intégrer dans notre algorithme de coupes et branchements. Enfin il faudrait améliorer la valeur de α_{min} des contraintes de borne.

7.7 Conclusion

Nous avons étudié dans ce chapitre le problème du k -séparateur équilibré. Nous avons proposé une formulation exacte de ce problème en terme de programme linéaire en nombres entiers. Nous avons effectué une étude polyédrale et nous avons discuté de quelques résultats expérimentaux.

L'une des perspectives serait d'approfondir l'étude polyédrale des contraintes de dominants connexes et non-connexes afin d'améliorer l'algorithme de coupes et branchements et de renforcer la relaxation linéaire. Une seconde perspective serait de trouver un algorithme de séparation des inégalités de dominants connexes efficace afin de les ajouter plus efficacement dans notre algorithme de coupes et branchements. Enfin il serait intéressant de voir s'il est possible d'améliorer la valeur de α_{min} pour notre problème.

Instance	k	q	TT_α	No_α	Gap_α	Opt_α
david	2	5	0,2	21	0,0	Optimal
david	2	10	0,2	21	0,0	Optimal
david	3	5	1,4	99	0,0	Optimal
david	3	10	1,3	27	0,0	Optimal
david	4	5	5,0	649	0,0	Optimal
david	4	10	4,0	103	0,0	Optimal
games120	2	5	73,0	166605	0,0	Optimal
games120	2	10	64,0	141328	0,0	Optimal
games120	3	5	10428,8	1607539	0,0	Optimal
games120	3	10	11009,0	2101050	0,0	Optimal
games120	4	5	18000	1234936	0,2	Feasible
games120	4	10	18000	1062501	0,1	Feasible
miles750	2	5	1,5	127	0,0	Optimal
miles750	2	10	1,3	192	0,0	Optimal
miles750	3	5	420,9	3857	0,0	Optimal
miles750	3	10	62,0	433	0,0	Optimal
miles750	4	5	8650,2	40312	0,0	Optimal
miles750	4	10	5322,2	19360	0,0	Optimal
miles1000	2	5	3,0	146	0,0	Optimal
miles1000	2	10	2,6	100	0,0	Optimal
miles1000	3	5	684,4	3463	0,0	Optimal
miles1000	3	10	822,9	3994	0,0	Optimal
miles1000	4	5	18000	67726	0,3	Feasible
miles1000	4	10	18000	62120	0,4	Feasible
myciel3	2	5	0,0	9	0,0	Optimal
myciel3	2	10	0,0	0	0,0	Optimal
myciel3	3	5	0,0	9	0,0	Optimal
myciel3	3	10	0,0	0	0,0	Optimal
myciel3	4	5	0,0	9	0,0	Optimal
myciel3	4	10	0,0	0	0,0	Optimal

TABLE 7.6 – Résultat du P_r en ajoutant la contrainte de borne (partie 1).

Instance	k	q	TT_α	No_α	Gap_α	Opt_α
myciel4	2	5	0,1	23	0,0	Optimal
myciel4	2	10	0,1	20	0,0	Optimal
myciel4	3	5	0,1	78	0,0	Optimal
myciel4	3	10	0,1	20	0,0	Optimal
myciel4	4	5	0,4	453	0,0	Optimal
myciel4	4	10	0,3	81	0,0	Optimal
myciel5	2	5	0,2	107	0,0	Optimal
myciel5	2	10	0,1	69	0,0	Optimal
myciel5	3	5	1,1	195	0,0	Optimal
myciel5	3	10	0,9	338	0,0	Optimal
myciel5	4	5	2,3	515	0,0	Optimal
myciel5	4	10	25,8	13216	0,0	Optimal
myciel6	2	5	3,4	2943	0,0	Optimal
myciel6	2	10	3,1	2069	0,0	Optimal
myciel6	3	5	5,8	373	0,0	Optimal
myciel6	3	10	552,0	127939	0,0	Optimal
myciel6	4	5	4572,8	492041	0,0	Optimal
myciel6	4	10	494,6	29135	0,0	Optimal
myciel7	2	5	1130,0	628412	0,0	Optimal
myciel7	2	10	294,7	166212	0,0	Optimal
myciel7	3	5	18000	408464	0,2	Feasible
myciel7	3	10	18000	333416	0,1	Feasible
myciel7	4	5	18000	106624	0,1	Feasible
myciel7	4	10	18000	106447	0,1	Feasible
queen7-7	2	5	2,2	7095	0	Optimal
queen7-7	2	10	3,5	3950	0	Optimal
queen7-7	3	5	15,7	9201	0	Optimal
queen7-7	3	10	16,7	9983	0	Optimal
queen7-7	4	5	56,6	26292	0	Optimal
queen7-7	4	10	67,5	11904	0	Optimal

TABLE 7.7 – Résultat du P_r en ajoutant la contrainte de borne (partie 2).

Instance	k	q	TT_{dom}	No	Gap_{dom}	opt	N_{ctdom}	T_{sepdom}
R-35	2	5	1,35	22	0,0	Optimal	0	0,95
R-35	2	10	1,48	11	0,0	Optimal	25	1,16
R-35	3	5	3,77	9	0,0	Optimal	30	1,47
R-35	3	10	5,47	54	0,0	Optimal	25	3,68
R-35	4	5	3,34	17	0,0	Optimal	30	0,80
R-35	4	10	3,80	9	0,0	Optimal	25	1,17
R-150	2	5	180,79	107	0,0	Optimal	110	177,42
R-150	2	10	189,28	90	0,0	Optimal	107	186,19
R-150	3	5	79,16	46	0,0	Optimal	120	51,61
R-150	3	10	462,58	278	0,0	Optimal	114	374,35
R-150	4	5	949,01	567	0,0	Optimal	125	405,95
R-150	4	10	285,51	137	0,0	Optimal	121	151,14
david	2	5	13,04	93	0,0	Optimal	81	12,35
david	2	10	23,08	95	0,0	Optimal	76	22,41
david	3	5	37,90	282	0,0	Optimal	82	31,30
david	3	10	44,83	182	0,0	Optimal	77	40,05
david	4	5	51,44	419	0,0	Optimal	82	36,68
david	4	10	76,99	340	0,0	Optimal	77	61,75
games120	2	5	18000	16133	0,2	Feasible	0	17965,91
games120	2	10	18000	14239	0,2	Feasible	0	17970,21
games120	3	5	18000	23090	0,2	Feasible	69	17425,26
games120	3	10	18000	20213	0,3	Feasible	0	17484,96
games120	4	5	18000	33214	0,5	Feasible	71	15874,04
games120	4	10	18000	21345	0,3	Feasible	0	16331,82
miles750	2	5	418,26	385	0,0	Optimal	6	412,44
miles750	2	10	797,99	622	0,0	Optimal	114	789,62
miles750	3	5	1641,40	1994	0,0	Optimal	74	1219,68
miles750	3	10	1898,37	1848	0,0	Optimal	72	1447,90
miles750	4	5	18000	15068	0,5	Feasible	79	12314,51
miles750	4	10	18000	12922	0,2	Feasible	51	9783,31
miles1000	2	5	260,37	347	0,0	Optimal	0	250,00
miles1000	2	10	1077,63	836	0,0	Optimal	0	1065,18
miles1000	3	5	1048,73	988	0,0	Optimal	0	560,28
miles1000	3	10	1150,81	924	0,0	Optimal	0	672,34
miles1000	4	5	18000	16784	0,9	Feasible	2	9312,45
miles1000	4	10	18000	16123	0,9	Feasible	5	9502,56

TABLE 7.8 – Résultat du P_r en ajoutant la contrainte de borne et les contraintes de dominants connexes (partie 1).

Instance	k	q	TT_{dom}	No	Gap_{dom}	opt	N_{ctdom}	T_{sepdom}
myciel3	2	5	0,21	7	0,0	Optimal	0	0,19
myciel3	2	10	0,07	0	0,0	Optimal	0	0,06
myciel3	3	5	0,28	9	0,0	Optimal	0	0,24
myciel3	3	10	0,08	0	0,0	Optimal	0	0,08
myciel3	4	5	0,24	7	0,0	Optimal	0	0,19
myciel3	4	10	0,08	0	0,0	Optimal	0	0,08
myciel4	2	5	1,20	40	0,0	Optimal	0	1,12
myciel4	2	10	0,99	18	0,0	Optimal	0	0,92
myciel4	3	5	1,01	24	0,0	Optimal	0	0,64
myciel4	3	10	1,56	26	0,0	Optimal	0	1,13
myciel4	4	5	3,45	291	0,0	Optimal	0	2,52
myciel4	4	10	1,54	26	0,0	Optimal	0	1,05
myciel5	2	5	8,12	223	0,0	Optimal	0	7,63
myciel5	2	10	17,32	204	0,0	Optimal	0	16,90
myciel5	3	5	17,85	444	0,0	Optimal	7	14,79
myciel5	3	10	55,13	994	0,0	Optimal	0	51,01
myciel5	4	5	297,25	9295	0,0	Optimal	0	251,62
myciel5	4	10	161,39	2852	0,0	Optimal	0	148,70
myciel6	2	5	1903,82	10986	0,0	Optimal	69	1886,75
myciel6	2	10	6214,51	22110	0,0	Optimal	16	6178,01
myciel6	3	5	14857,50	90643	0	Optimal	84	13483,23
myciel6	3	10	9928,05	44818	0	Optimal	37	9255,01
myciel6	4	5	838,42	4322	0	Optimal	1	622,75
myciel6	4	10	2859,09	9614	0	Optimal	0	2379,68
myciel7	2	5	18000	14324	0,2	Feasible	21	17762,45
myciel7	2	10	18000	8912	0,2	Feasible	145	17819,42
myciel7	3	5	18000	16237	0,3	Feasible	97	16321,35
myciel7	3	10	18000	10032	0,3	Feasible	43	15312,89
myciel7	4	5	18000	12435	0,3	Feasible	24	11342,32
myciel7	4	10	18000	7923	0,2	Feasible	37	12874,13
queen7-7	2	5	236,77	4898	0,0	Optimal	0	232,99
queen7-7	2	10	690,65	8772	0,0	Optimal	0	684,54
queen7-7	3	5	623,33	14368	0,0	Optimal	0	579,18
queen7-7	3	10	707,19	7959	0,0	Optimal	0	672,39
queen7-7	4	5	845,94	20253	0,0	Optimal	0	725,12
queen7-7	4	10	1309,19	13771	0,0	Optimal	0	1169,15

TABLE 7.9 – Résultat du P_r en ajoutant la contrainte de borne et les contraintes de dominants connexes (partie 2).

Conclusion

Dans cette thèse, nous avons étudié les problèmes de l'analyse structurelle pour les systèmes algébro-différentiels conditionnels et les systèmes algébro-différentiels conditionnels imbriqués ainsi que le problème de parallélisation des systèmes algébro-différentiels.

Dans un premier temps, nous avons étudié le problème de l'analyse structurelle pour les systèmes algébro-différentiels conditionnels. Nous avons montré que ce problème se ramène au problème du sous-graphe sans couplage parfait (PSGSCP). Nous avons ensuite étudié la complexité du problème PSGSCP. Nous avons montré que ce dernier est NP-complet en utilisant une réduction d'une variante du problème du stable dans les graphes tripartis. Pour cela nous étions amenés à montrer que cette variante du problème du stable est elle même NP-complet. Comme conséquence nous avons obtenu que le problème de l'analyse structurelle conditionnelle est NP-complet. Nous avons étudié deux formulations en nombres entiers pour le problème PSGSCP. Nous avons aussi étudié une approche polyédrale pour ce problème. Nous avons identifié de nouvelles contraintes valides, et nous avons donné des conditions nécessaires et des conditions suffisantes pour que ces inégalités définissent des facettes. Nous avons montré que le problème de séparation pour certaines de ces classes peut être résolu en temps polynomial. Cette étude faciale nous a permis de développer et implémenter un algorithme de coupes et branchements pour le problème de l'analyse structurelle. Nous avons montré que la difficulté de résolution du PSGSCP ne dépend pas uniquement de la taille de l'instance.

Nous nous sommes ensuite intéressés à une extension de ce problème pour les systèmes algébro-différentiels conditionnels imbriqués. Nous avons montré que ce problème se ramène au problème généralisé du PSGSCP (G-PSGSCP). Nous avons étudié une formulation en nombres entiers pour ce problème. Nous avons aussi étudié une approche polyédrale pour ce problème. Nous avons identifié de nouvelles contraintes valides et nous avons étendu plusieurs contraintes de la formulation associée au PSGSCP. Nous avons donné des conditions nécessaires et des conditions suffisantes pour que ces inégalités définissent des facettes.

Enfin, dans le dernier chapitre, nous avons considéré le problème de parallélisation des systèmes algébro-différentiels. Ce problème se ramène au problème de k -séparateur équilibré. Nous avons proposé deux formulations approchées et une formulation exacte en nombres entiers. Nous avons aussi étudié une approche polyédrale sur la formulation exacte du problème. Nous avons identifié de nouvelles contraintes valides et nous avons étendu plusieurs contraintes proposées par Balas et Souza [8] ainsi que Didi Biha et Meurs [18]. Nous avons donné des conditions nécessaires et des conditions suffisantes pour que ces inégalités définissent des facettes. Nous avons testé cette formulation, renforcée par quelques nouvelles contraintes, sur des instances réelles et des instances aléatoires. Nous avons montré que la contrainte de bornes, proposée par Didi Biha et Meurs, adaptées à notre problème permet d'améliorer les performances.

Ce travail de recherche ouvre plusieurs perspectives liées à la fois à l'exploration théorique et à l'expérimentation numérique. Tout d'abord pour pouvoir résoudre le problème de l'analyse structurelle sur des systèmes algébro-différentiels conditionnels de grandes tailles, il serait très intéressant d'améliorer les performances de l'algorithme de coupes et branchements. L'une des pistes serait d'étudier la complexité de la séparation des contraintes de k -couplages et de sous-graphe bien contraint. Nous conjecturons que ces problèmes sont NP-difficiles. Il serait intéressant de trouver des conditions nécessaires et suffisantes pour que les contraintes de k -couplages et de sous-graphe régulier définissent des facettes. Ces conditions permettraient d'utiliser ces contraintes d'une manière plus efficace. Une autre piste serait d'identifier d'autres familles d'inégalités valides et d'étudier leurs structures faciales.

Une autre perspective intéressante serait de développer un algorithme de coupes et branchements pour résoudre le problème généralisé du PSGSCP. Cet algorithme pourrait s'appuyer sur les nouvelles contraintes et les extensions des contraintes valides que nous avons proposées.

D'autre part pour améliorer le temps de résolution du problème de parallélisation des systèmes algébro-différentiels, il serait intéressant de trouver des conditions nécessaires et suffisantes pour que les contraintes de dominants connexes et non-connexes définissent des facettes. Ces conditions permettraient d'utiliser ces contraintes efficacement. Une autre piste intéressante serait d'affiner la borne α_{min} pour $k > 2$. Enfin, il serait intéressant d'identifier d'autres familles d'inégalités valides et d'étudier leurs structures faciales.

Bibliographie

- [1] Abacus - a branch-and-cut system. <http://www.informatik.uni-koeln.de/abacus/>.
- [2] Chaco software. <http://www.sandia.gov/bahendr/chaco.html>.
- [3] Instance dimacs. <http://mat.gsia.cmu.edu/COLOR/instances.html>.
- [4] Lemon graph. <http://lemon.cs.elte.hu/trac/lemon>.
- [5] Metis software. <http://glaros.dtc.umn.edu/gkhome/views/metis>.
- [6] A.V. Aho, J.E. Hopcroft, and J.D. Ullman. The Design and Analysis of Computer Algorithms. *Addison-wesley, Reading, MA*, 1974.
- [7] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows : Theory, Algorithms, and Applications*. Prentice Hall, February 1993.
- [8] E. Balas and C. de Souza. The vertex separator problem : a polyhedral investigation. *Mathematical Programming*, 103(3) :583–608, 2005.
- [9] M.O. Ball and U. Derigs. An analysis of alternate strategies for implementing matching algorithms. *Networks*, 13 :517–549, 1983.
- [10] K. Brenan. *Stability and Convergence of Difference Approximations for Higher Index Differential-Algebraic Systems with Applications in Trajectory Control*. Ph. d. thesis, University of California at Los Angeles, 1983.
- [11] R.L. Brown and C.W. Gear. A program for the solution of simultaneous implicit differential and nonlinear equations. Technical Report UIUCDCS-R-73-575, University of Illinois at Urbana-Champaign, 1973.
- [12] P. Bujakiewicz. *Maximum Weighted Matching for High Index Differential Algebraic Equations*. Doctor's dissertation, Delft University of Technology, 1994.
- [13] S.L. Campbell. The numerical solution of higher index linear time varying singular systems of differential equations. *SIAM J. Sci. Statist. Comput.*, 6 :334–348, 1985.
- [14] C. Chevalier. *Conception et mise en oeuvre d'outils efficaces pour le partitionnement et la distribution parallèles de problèmes numériques de très grande taille*. P.h.d., L'Université de Bordeaux 1, 2007.

- [15] S.A. Cook. The complexity of theorem-proving procedures. In *Proceedings 3rd Annual ACM Symposium on Theory of Computing*, pages 151–158, New York, 1971. Association for Computing Machinery.
- [16] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, 1963.
- [17] C. de Souza and E. Balas. The vertex separator problem : algorithms and computations. *Mathematical Programming*, 103(3) :609–631, 2005.
- [18] M. Didi Biha and M.J. Meurs. An Exact Algorithm for Solving the Vertex Separator Problem. *Journal of Global Optimization*, 49 :425–434, 2010.
- [19] H.N. Djidjev. Partitioning Planar Graphs with Vertex Cost : Algorithms and Applications. *Algorithmica*, 28 :51–75, 2000.
- [20] I. S. Duff and C. W. Gear. Computing the structural index. *SIAM J. Alg. Disc. Meth*, 7 :594–603, 1986.
- [21] I.S. Duff and C.W. Gear. Computing the structural index. Technical report, Technical Memorandum 50, Mathematics and Computer Science Division, Argonne National Laboratory, 1985.
- [22] J. Edmonds. Covers and packings in a family of sets. *Bull. American Mathematical Society*, 68(5) :494–499, 1962.
- [23] J. Edmonds. Maximum matching and a polyhedron with 0,1-vertices. *Journal of Research of the National Bureau of Standards*, 69 :125 – 130, 1965.
- [24] C.M. Fiduccia and R.M. Mattheyses. A linear-time heuristic for improving network partition. In *Proceedings of the 19th Design Automation Conference*, pages 175–181, 1982.
- [25] L.R Ford and D.R. Fulkerson. Maximal flow through a network. *Can. J. Math.*, 8 :399–404, 1956.
- [26] H.N. Gabow and Tarjan R.E. Algorithms for two bottleneck optimization problems. *Journal of Algorithms*, 9 :411–417, 1988.
- [27] H.N. Gabow and Tarjan R.E. Faster scaling algorithms for network problems. *SIAM Journal on Computing*, 18 :1013–1036, 1989.
- [28] Z. Galil, S. Micali, and H.N. Gabow. Priority queues with variable priority and an $o(ev \log v)$ algorithm for finding a maximal weighted matching in general graphs. *23rd Annual Symposium on Foundations of Computer Science (Chicago, 1982)*, pages 255–261, 1982.
- [29] Z. Galil and A. Naamad. An $0(ev \log^2 v)$ algorithm for the maximal flow problem. *J. Comput. Syst. Sci.*, pages 203–217, 1980.

- [30] T. Gallai. Uber extreme punkt- und kantenmengen. *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae, Sectio Mathematica 2*, pages 133–138, 1959.
- [31] F.R. Gantmacher. *The Theory of Matrices vol 2*. New York : Chelsea, 1974.
- [32] M.R. Garey and D.S. Johnson. Computers and Intractability : A Guide to the Theory of NP-Completeness. *W. H. Freeman and Company, New-York*, 1979.
- [33] N. Garg, H. Saran, and V.V. Vazirani. Finding separator cuts in planar graphs within twice the optimal. *SIAM J. Computing*, pages 159–179, 1999.
- [34] C.W. Gear. Simultaneous numerical solution of differential-algebraic equations. *IEEE Trans. Circuit Theory*, CT-18 :89–95, 1971.
- [35] A.V. Golberg and Tarjan R.E. Finding minimum-cost circulations by successive approximation. *Mathematics of Operations Research*, 15 :430–466, 1990.
- [36] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimisation. *Combinatorica 1*, pages 70–89, 1981.
- [37] D. Guibert. *Analyse de méthodes de résolution parallèles d'EDO/EDA raides*. Ph.d. thesis, l'Université Claude Bernard - Lyon 1, 2009.
- [38] P. Hall. On representatives of subset. *The Journal of the London Mathematical Society*, 10 :26–30, 1935.
- [39] J.E. Hopcroft and R.M. Karp. A $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *Conference Record 1971 Twelfth Annual Symposium on Switching and Automata Theory*, pages 122–125, 1971.
- [40] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [41] Ilog, Inc. Solver cplex, 2011. <http://www.ilog.fr/products/cplex/>.
- [42] D. Jian-Wan, C. Li-Ping, Z. Fan-Li, W. Yi-Zhong, and W. Guo-Biao. An Analyzer for Declarative Equation Based Models. *Modelica 2006*, 01 :349–357, 2006.
- [43] Y. Kamidoi, S. Wakabayashi, and N. Yoshida. A divide-and-conquer approach to the minimum k-way cut problem. *Algorithmica*, 32 :262–276, 2002.
- [44] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4) :373–395, December 1984.
- [45] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20 :359–392, 1998.
- [46] G. Karypis and V. Kumar. Multilevel k-way Partitioning Scheme for Irregular Graphs. *Journal of Parallel and Distributed Computing*, 48 :96–129, 1998.

- [47] G. Karypis and V. Kumar. Parallel Multilevel k-way Partitioning Scheme for Irregular Graphs. *SIAM Review*, 41 :278–300, 1999.
- [48] D.J. Kelly and G.M. O’Neill. *The minimum cost flow problem and the network simplex method*. Master’s thesis, University College, Dublin, Ireland, 1991.
- [49] L. Khachiyan. A polynomial algorithm in linear programming. *Soviet Math. Dokl.*, 20 :191–194, 1979.
- [50] D. König. Graphok és matrixok. *Matematikai és Fizikai Lapok*, 38 :116–119, 1931.
- [51] M. Lacroix, R. Mahjoub, and S. Martin. Structural analysis in Differential-Algebraic Systems and Combinatorial Optimization. *Proceedings of 39th International Conference on Computers & Industrial Engineering (CIE39)*, pages 331–337, 2009.
- [52] M. Lacroix, R. Mahjoub, and S. Martin. Structural analysis for Differential-Algebraic Systems : Complexity, Formulation and Facets. *Proceedings ISCO 2010, Electronic Notes in Discrete Mathematics 36*, pages 1073–1080, 2010.
- [53] M. Lacroix, R. Mahjoub, and S. Martin. Combinatorial Optimization Model and MIP formulation for the Structural Analysis of Conditional Differential-Algebraic Systems. *Computers & Industrial Engineering*, pages 422–429, 2011.
- [54] M. Lacroix, R. Mahjoub, S. Martin, and C. Picouleau. On the NP-Completeness of the Perfect Matching Free Subgraph Problem. *Rapport technique N° 312, LAMSADE, Université Paris-Dauphine*, 2011.
- [55] M. Lacroix, R. Mahjoub, S. Martin, and C. Picouleau. On the NP-Completeness of the Perfect Matching Free Subgraph Problem . *En correction, à paraître dans Theoretical Computer Science*, 2011.
- [56] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6) :787–832, 1999.
- [57] A. Leitold and K. M. Hangos. Structural solvability analysis of dynamic process models. *Computers and Chemical Engineering*, 25 :1633–1646, 2001.
- [58] R.J. Lipton and R.E. Tarjan. A Separator Theorem for Planar Graphs. *SIAM J. Appl.Math.*, 36 :177–189, 1979.
- [59] L. Lovász. On Determinants, Matchings, And Random Algorithms. Technical report, Bolyai Institute, Jozsef Attila University, 1979.
- [60] L. Lovász and M. D. Plummer. *Matching Theory*. North-Holland, 1986.
- [61] P. Lötstedt and L. Petzold. Numerical Solution of Nonlinear Differential Equations with Algebraic Constraints I : Convergence Results for Backward Differentiation Formulas. *Mathematics Of Computation*, 46(174) :491–516, 1986.

- [62] D.G. Luenberger. Time-Invariant Descriptor Systems. *Automatica*, 14 :473–480, 1978.
- [63] K. Murota. *Systems Analysis by Graphs and Matroids*. Springer-Verlag, 1987.
- [64] K. Murota. *Matrices and Matroids for Systems Analysis*. Springer-Verlag, 2000.
- [65] H. Nilsson. Type-Based Structural Analysis for Modular Systems of Equations. *Proceedings of the 2nd International Workshop on Equation-Based Object-Oriented Languages and Tools*, pages 71–81, 2008.
- [66] G. Oriolo. Clique family inequalities for the stable set polytope for quasi-line graphs. *Discrete Applied Mathematics*, 132 :185–201, 2004.
- [67] Chartier P. *Méthodes numériques pour les équations différentielles ordinaires et algébriques avec application aux systèmes hamiltoniens*. H.d.r., L’Université de Rennes 1 Institut Mathématique de Rennes, 2000.
- [68] C. C. Pantelides. The Consistent Initialization of Differential-algebraic Systems. *SIAM Journal of Scientific and Statistical Computing*, pages 213–231, 1988.
- [69] J. Petersen. Die Theorie der Regulären Graphen. *Acta Math.*, 15 :193–200, 1891.
- [70] L. Petzold. Differential/Algebraic Equations are not ODE’S. *SIAM J. SCI. STAT. COMPUT.*, 3(3) :367–384, 1982.
- [71] C. Phillips and T. J. Warnow. The asymmetric median tree A new model for building consensus trees. *Combinatorial Pattern Matching*, pages 234–252, 1996.
- [72] A. Pothen. Graph partitioning algorithms with applications to scientific computing. In *Parallel Numerical Algorithms*. *Kluwer Academic Press*, 1996.
- [73] M. Z. Poulsen. *Structural Analysis of DAEs*. Ph.d. thesis, Informatics and Mathematical Modelling Technical University of Denmark, 2001.
- [74] W.R. Pulleyblank. *Handbooks in Operations Research and Management Science*, chapter Polyhedral combinatorics, pages 371–446. 1989.
- [75] T. Rübner-Peterson. An efficient algorithm using backward time-scaled differences for solving stiff differential algebraic systems. Technical report, Insitute of Circuit Theory and Telecommunication, Technical University of Denmark, 1973.
- [76] G. Reibig and U. Feldmann. A simple and general method for detecting structural inconsistencies in large electrical networks. *Circuits and Systems I : Fundamental Theory and Applications*, 50(11) :237–240, 2002.
- [77] A. Schrijver. Theory of linear and integer programming. *Wiley Interscience, Chichester*, 1986.
- [78] A. Schrijver. *Combinatorial Optimization : Polyhedra and Efficiency (Algorithms and Combinatorics)*. Springer book, 2004.

-
- [79] R.F. Sincovec, B. Dembart, M.A. Epton, A.M. Erisman, S.W. Manke, and E.L. Yip. Solvability of Large Scale Descriptor Systems. Technical report, Boeing Computer Services Company, 1979.
- [80] R.F. Sincovec, A.M. Erisman, E.L. Yip, and M.A. Epton. Analysis of Descriptor Systems Using Numerical Algorithms. *IEE Transactions on Automatic Control*, AC-26(1) :139–147, 1981.
- [81] J.W. Starnes. A numerical algorithm for the solution of implicit algebraic-differential systems of equations. Technical report, Dept. of Mathematics and Statistics, Univ. of New Mexico, 1976.
- [82] E.L. Yip and R.F. Sincovec. Solvability, Controllability, and Observability of Continuous Descriptor Systems. *IEE Transactions on Automatic Control*, AC-26(3) :702–707, 1981.

Analyse structurelle des systèmes algébro-différentiels conditionnels : complexité, modèles et polyèdres.

Résumé

Les systèmes algébro-différentiels permettent de modéliser des systèmes physiques complexes comme les circuits électriques et les mouvements dynamiques. L'analyse structurelle des systèmes algébro-différentiels permet de vérifier si un tel système ne pourra pas être résolu par des méthodes numériques. Dans cette thèse, nous étudions ce problème dans le cas des systèmes algébro-différentiels conditionnels. Nous montrons que ce dernier est NP-difficile. Nous proposons une formulation en termes de graphes et deux formulations en nombres entiers pour ce problème. Nous étudions le polyèdre associé et décrivons plusieurs classes de contraintes valides. Nous donnons des conditions nécessaires et suffisantes pour que ces contraintes définissent des facettes. Nous développons un algorithme de coupes et branchements. Nous étudions aussi une extension de ce problème pour les systèmes algébro-différentiels conditionnels imbriqués. Nous nous intéressons au problème de séparateur.

Mots clés : Analyse structurelle, système algébro-différentiel, graphe, couplage, complexité, polytope, facette, séparation, algorithme de coupes et branchements, problème du séparateur.

Structural analysis for conditional differential-algebraic systems : complexity, formulation and facets.

Abstract

Differential algebraic systems are used for modeling complex physical systems as electrical networks and dynamic movements. The structural analysis for differential algebraic systems permits to verify if these systems can not be solved with numerical methods. In this thesis, we consider the structural analysis problem for differential algebraic systems with conditional equations. We show the NP-completeness of this latter problem. We propose a formulation in terms of graphs and two integer programming formulations. We study the polytope associated to this problem and describe several classes of valid inequalities. We give necessary and sufficient conditions for these inequalities to be facet defining. We also discuss separation algorithms for these constraints. We develop a branch-and-cut algorithm based on these results. We also study an extension of this problem to differential algebraic systems with conditional embedded equations. We study the separator problem.

Keywords : Structural analysis, differential algebraic system, graph, matching, complexity, polytope, facet, separation problem, branch-and-cut algorithm, separator problem.