

THÈSE DE DOCTORAT DE L'UNIVERSITÉ PARIS 6

SPÉCIALITÉ INFORMATIQUE

PRÉSENTÉE PAR

M. Pierre Bonami

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ PARIS 6

Sujet de la thèse :

Étude et mise en œuvre d'approches polyédriques pour  
la résolution de programmes en nombres entiers ou  
mixtes généraux

soutenue le 21 octobre 2003

devant le jury composé de :

Gérard Cornuéjols	Rapporteur
Virginie Gabrel	Examinatrice
Jean-Yves Jaffray	Examineur
Nelson Maculan	Rapporteur
Michel Minoux	Directeur de thèse



## Résumé

Ces vingt dernières années, les méthodes de plans coupants ont pris une importance croissante dans les algorithmes de résolution des programmes linéaires en nombre entiers. Le but du travail présenté ici est d'étudier et de proposer des méthodes de plans coupants pour des programmes en nombres entiers généraux prenant mieux en compte les structures des problèmes que les méthodes actuelles. De nombreuses expérimentations ont été effectuées dans le but d'essayer de mesurer l'efficacité et l'impact des différentes méthodes proposées sur des problèmes variés.

Nous étudions en particulier le problème de séparation de la meilleure inégalité valide suivant le critère du ratio, et différentes méthodes permettant d'approcher celle-ci en faisant diminuer sensiblement les temps de calcul au travers de l'utilisation d'heuristiques, de relaxations mixtes et d'une nouvelle méthode de lifting relaxé. Nous abordons ensuite les problèmes de séparation au travers de la programmation disjonctive, en proposant notamment de séparer sur la fermeture élémentaire des coupes de "Lift and Project" et montrons que l'utilisation des coupes disjonctives uniquement de rang 1 permet d'obtenir des renforcements très significatifs sur certains problèmes. Nous étudions ensuite les relaxations par reformulation-linéarisation de type Sherali Adams et proposons une nouvelle méthode d'optimisation combinant de façon originale décomposition et relaxation. Dans cette partie, nous abordons aussi l'étude théorique et pratique de l'intérêt de la relaxation de Sherali-Adams par rapport à la fermeture élémentaire des coupes de lift and project. Enfin nous proposons et expérimentons plusieurs nouveaux schémas algorithmiques pour l'utilisation de coupes conditionnellement valides.

**Mots Clés :** Programmation mathématique, Optimisation combinatoire, Plans coupants, Lift and Project, Reformulation-Linearisation, Coupes conditionnellement valides

## Abstract

Over the last twenty years, the cutting plane approach has taken an increasing role in the algorithmic resolution of mixed and integer linear programs. The work presented here aims at studying several cutting plane methods and at proposing new techniques for general mixed integer programs. Our goal is to study methods which require more computations than commonly used methods but produce stronger inequalities and tighter relaxations. We present extensive computational experiments in order to try to measure the practical efficiency of each method proposed.

In particular, we study the problem of the separation of the best valid inequality through the ratio criterion, and then several methods to try to approach those best inequalities through the use of heuristics, mixed relaxations, and of a new method of relaxed lifting. We then turn our interest to the separation through disjunctive programming. We propose a new method to optimize and to separate cutting planes on the elementary closure of lift and project cuts, and produce computational experiments which show that one can get significant reinforcement of the linear relaxations using only cuts of rank 1. We then study the related Sherali-Adams reformulation-linearization technique. We propose a new method to optimize over this relaxation through iterative decomposition and relaxation. We also study the relative interest of using Sherali-Adams relaxation or lift and project closure both practically through computational experiments and theoretically through the study of specific structured combinatorial problems. In the last part, we propose several new algorithmic methods for using conditionnaly valid cuts.

**Keywords :** Mathematical Programming, Combinatorial Optimization, Cutting Planes , Lift and Project, Reformulation-Linearisation Techniques, Conditionnaly Valid Inequalities



# Remerciements

A l'issue de ce travail, je tiens à remercier Michel Minoux pour avoir dirigé mes recherches.

Je lui suis très reconnaissant pour l'attention constante qu'il a portée à mes travaux depuis mon stage de DEA jusqu'à maintenant, depuis les discussions du début sur le sujet, les idées nouvelles et intéressantes à développer jusqu'à la phase finale de la rédaction et aux nombreuses relectures qui l'accompagnent. Son enthousiasme pour la matière et la confiance qu'il m'a accordée m'ont bien aidé à continuer de progresser tout au long de ce travail.

Je remercie Nelson Maculan et Gérard Cornuéjols d'avoir accepté d'être les rapporteurs de ma thèse et Virginie Gabrel et Jean-Yves Jaffray d'avoir accepté de faire partie du jury.

Je remercie aussi chaleureusement Viet-Hung Nguyen pour l'aide et les conseils qu'il m'a donnés pendant la préparation de cette thèse.

Enfin, je remercie tous les autres qui m'ont aidé à des titres divers à mener ce travail à bien. Ils sont heureusement trop nombreux pour être cités ici sans en oublier aucun.



# Table des matières

<b>Introduction générale</b>	<b>11</b>
<b>1 Introduction</b>	<b>15</b>
1.1 Principales inégalités valides utilisés dans les solveurs . . . . .	15
1.1.1 Structures de stables et contraintes d'exclusion binaires . . . . .	16
1.1.2 Structure de sac à dos . . . . .	17
1.1.3 Contraintes de semi-affectation et de sac à dos . . . . .	19
1.1.4 Coupes de Chvátal-Gomory . . . . .	19
1.1.5 Coupes mixtes de Gomory . . . . .	20
1.2 Exemples de problèmes d'application . . . . .	21
1.2.1 Problème de planification des arrêts de sites de production à EDF . . . . .	21
1.2.2 Problème de dimensionnement de réseau avec fonctions de coûts en escaliers	22
1.2.3 Problème de l'affectation des personnels navigants dans une compagnie aérienne	23
1.2.4 Le problème du sac à dos multidimensionnel . . . . .	24
1.2.5 Les problèmes de satisfiabilité (SAT) . . . . .	25
1.2.6 Les problèmes MAX-SAT . . . . .	25
<b>2 Méthodes de génération d'inégalités valides générales</b>	<b>27</b>
2.1 Rappels d'algèbre linéaire et de théorie des polyèdres . . . . .	28
2.2 Séparation d'inégalités valides générales . . . . .	29
2.2.1 Formulation du problème de séparation . . . . .	30
2.2.2 Transformation du problème de séparation en un problème d'optimisation . .	31
2.2.3 Qualité de l'inégalité générée et conditions de normalisation . . . . .	33
2.2.4 Résolution algorithmique du problème de séparation . . . . .	38
2.3 Renforcement d'une inégalité coupante . . . . .	39
2.3.1 Formulation du problème du renforcement . . . . .	39
2.3.2 Résolution du problème de renforcement d'une inégalité . . . . .	40
2.4 Extension d'une inégalité valide . . . . .	44
2.5 Transformation d'une inégalité en une facette . . . . .	46
2.6 Applications de l'extension d'inégalités valides . . . . .	48
2.6.1 Le cas du lifting . . . . .	48
2.6.2 Séparation d'inégalités génériques pour le voyageur de commerce . . . . .	49
<b>3 Étude comparative de méthodes de séparation par oracles : application au sac à dos multidimensionnel</b>	<b>53</b>
3.1 Séparations des meilleurs inégalités suivant le critère du ratio . . . . .	54

3.2	Séparation approchée par résolution heuristique du sous-problème . . . . .	57
3.3	Séparation exacte sur des relaxations mixtes . . . . .	57
3.4	Résolution du problème de séparation sur une restriction du problème et lifting . . .	58
3.4.1	Présentation de la procédure de lifting relaxé . . . . .	60
3.4.2	Résultats expérimentaux . . . . .	62
<b>4</b>	<b>Séparation et renforcement par la programmation disjonctive - Expérimentation</b>	<b>69</b>
4.1	La programmation disjonctive . . . . .	69
4.2	Stratégies de séparation par la programmation disjonctive . . . . .	72
4.2.1	Séparation sur des disjonctions simples . . . . .	72
4.2.2	Séparation sur des disjonctions multiples . . . . .	73
4.2.3	Séparation relativement à l'intersection des polyèdres disjonctifs . . . . .	74
4.3	Comparaisons pratiques des différentes stratégies de séparation . . . . .	76
4.3.1	Séparation sur des sacs à dos multidimensionnels . . . . .	77
4.3.2	Résultats de séparation sur des problèmes en nombres entiers généraux . . .	79
4.3.3	Résultats sur des problèmes de MAX-2SAT . . . . .	83
4.4	Expression de la relaxation de Sherali-Adams par reformulation-linéarisation . . . .	87
4.5	Formulation par la programmation disjonctive de la relaxation de Sherali-Adams d'ordre 1 . . . . .	89
4.6	Optimisation par décomposition . . . . .	90
4.6.1	Décomposition du problème de séparation . . . . .	91
4.6.2	Optimisation par ajout itératif des contraintes de $\tilde{C}$ . . . . .	92
4.7	Expérimentation du calcul de la relaxation . . . . .	93
4.8	Équivalence des relaxations $P_{L\&P}$ et $P_{SA}$ pour les problèmes de maximisation de fonction pseudo-booléennes quadratique non contraints . . . . .	96
<b>5</b>	<b>Génération d'inégalités coupantes conditionnellement valides</b>	<b>101</b>
5.1	Coupes énumératives . . . . .	101
5.1.1	Coupes d'intersections . . . . .	102
5.1.2	Dérivation des coupes énumératives . . . . .	102
5.1.3	Les coupes énumératives de Burdet . . . . .	103
5.1.4	Les coupes énumératives de Glover . . . . .	104
5.2	Coupes géométriques . . . . .	105
5.2.1	Coupes de proximité à un point extrême de l'hypercube . . . . .	105
5.2.2	Coupes de proximité à une face de l'hypercube . . . . .	106
5.3	Utilisation des coupes de proximité dans la littérature . . . . .	106
5.3.1	Algorithme heuristique([43]) . . . . .	106
5.3.2	Algorithme "explorer puis couper" ([46]) . . . . .	107
5.4	Nouveaux schémas algorithmiques proposés . . . . .	109
5.4.1	Algorithme "couper puis explorer" . . . . .	109
5.4.2	Algorithme d'unification des explorations . . . . .	110
5.5	Expérimentations . . . . .	111
5.5.1	Résultats avec l'algorithme "explorer puis couper" . . . . .	111
5.5.2	Résultats avec l'algorithme "couper puis explorer" simple . . . . .	113
5.5.3	Résultats avec la méthode de coupe puis réunion puis exploration . . . . .	113
5.6	Conclusions . . . . .	116





# Introduction générale

La programmation linéaire en nombres entiers constitue un cadre général particulièrement utile pour modéliser, et bien souvent résoudre, des problèmes d'optimisation dans des domaines d'application extrêmement variés : logistique et transport, productique, réseaux de télécommunications ou de distribution d'énergie, automatique, etc . . .

Grâce aux perfectionnements des techniques de *programmation linéaire continue* (algorithme du simplexe et méthodes de points intérieurs) ainsi qu'aux développements des techniques de la *combinatoire polyédrique*, des progrès très significatifs ont été accomplis depuis une dizaine d'années sur des problèmes combinatoires très fortement structurés tels que les problèmes de circuits hamiltoniens dans les graphes (problème du voyageur de commerce). Des recherches très actives, menées depuis le début des années 80, sur le polyèdre correspondant à l'enveloppe convexe des cycles hamiltoniens d'un graphe ont permis d'identifier les structures de nombreuses classes de facettes de ce polyèdre grâce auxquelles des relaxations très fortes peuvent être construites. De ce fait les évaluations par résolution continue des problèmes relaxés constituent des approximations très serrées des valeurs optimales entières, ce qui permet de diriger des recherches arborescentes (de type Branch and Bound) de façon très efficace. L'évolution, dans le temps, des progrès réalisés peut s'apprécier par la taille des records mondiaux des plus grands problèmes de voyageur de commerce résolus de façon exacte (c.à.d. en apportant la preuve de l'optimalité de la solution obtenue) : en 1980 par des techniques de combinatoire polyédrique mettant en œuvre des inégalités d'élimination de sous-tours et des inégalités dites "de peigne" (comb inequalities) Crowder et Padberg ont pu obtenir pour la première fois la solution exacte d'un problème à  $n = 318$  villes. En 1987, Padberg et Rinaldi, perfectionnant les mêmes techniques atteignent la taille de  $n = 532$ . Avec des techniques plus avancées, les derniers records obtenus se situent au dessus de  $n = 10000$  (cf par exemple Applegate, Bixby, Chvátal et Cook 2001).

Notons que dans la même période, des progrès analogues ont été enregistrés sur un certain nombre d'autres problèmes combinatoires à forte structure, par exemple le problème de l'arbre de Steiner (Chopra Gorres et Rao 1992, Chopra et Rao 1994), le problème d'optimisation de réseaux avec contraintes de connectivité (Goemans 1994, Grötschel, Momma et Stoer 1992,1995) ou des problèmes d'optimisation de réseaux à flots multiples et coûts discrets (Stoer et Dahl 1994 et 1998, Bienstock et Günlük 1996, pour un exposé de synthèse récent cf Minoux 2001).

Le principe commun à tous les travaux mentionnés ci-dessus consiste, par une analyse mathématique préalable approfondie exploitant au maximum toute la connaissance que l'on a de la structure des problèmes à résoudre, à mettre en évidence des classes d'inégalités valides (formant si possible des facettes) pour l'enveloppe convexe des solutions entières. Ces classes d'inégalités valides (de facettes) sont ensuite exploitées pour construire de bonnes relaxations selon un principe de génération

d'inégalités non satisfaites par la solution courante ; ce procédé de génération est encore appelé *séparation*. Notons que les cas où le sous-problème de séparation est soluble polynomialement sont malheureusement peu fréquents, dans la majorité des cas il faut se contenter de mettre en œuvre des heuristiques pour le résoudre si l'on ne veut pas compromettre l'efficacité algorithmique du processus.

Dans les solveurs de programmes linéaires en nombres entiers généraux, l'utilisation faite des plans coupants combine des coupes portant sur des structures spécifiques connues par l'étude de problèmes combinatoires et des coupes générales. Malheureusement, comme nous le verrons dans le chapitre 1, les problèmes en nombres entiers issus d'applications industrielles concrètes se caractérisent : soit par une absence de structure particulière ; soit par le fait qu'ils apparaissent comme combinant, de façon complexe, plusieurs types de structures connues souvent associées à des parties sans structures particulières.

Pour de tels problèmes que nous qualifierons ici de programmes en nombres entiers (purs ou mixtes) généraux, l'approche résultant de l'analyse de problèmes combinatoires structurés apparaît souvent insuffisante. En effet, même si on peut toujours exploiter la connaissance d'inégalités valides pour les parties structurées du problème, celles-ci ne correspondent qu'à des sous-ensembles de variables et de contraintes considérés indépendamment les uns des autres, et de ce fait l'appréhension globale du problème est perdue. Ainsi, pour de tels problèmes, l'utilisation de méthodes générales les plus efficaces possibles paraît très importante.

Du côté des méthodes générales, de nombreuses familles d'inégalités ont aussi été dérivées. Celles-ci sont établies principalement de trois manières : soit en procédant à des arrondis sur les parties entières d'inégalités valides pour la relaxation continue (coupes de Chvátal, coupes *MIR* de Nemhauser et Wolsey), soit en travaillant sur les parties fractionnaires d'inégalités valides pour la relaxation continue (coupes de Gomory entières et mixtes), soit en utilisant la programmation disjonctive (coupes de *lift and project* de Balas, Ceria et Cornuéjols, coupes *split*). En fait, les trois approches aboutissent souvent à des classes d'inégalités valides équivalentes. Ainsi, les coupes de Chvátal et de Gomory entières sont équivalentes. Les coupes de *lift-and-project* renforcées sont équivalentes aux coupes de Gomory déduites des solutions basiques. Les coupes de Gomory mixtes, les coupes *MIR* et *split* sont toutes trois équivalentes.

Parmi toutes ces familles de coupes, une seule est couramment utilisée dans la résolution de programmes en nombre entiers généraux ; elle est constituée des coupes mixtes de Gomory (dérivées de solutions basiques) et des coupes de *lift and project* renforcées. Ces coupes présentent l'avantage de pouvoir être séparées très rapidement, mais elles ne prennent en compte qu'une faible partie de la structure entière du problème (elles sont basées sur la prise en compte de l'intégrité d'une seule variable entière du problème à la fois).

L'objectif du travail présenté ici est précisément de mener une étude aussi complète que possible des techniques de génération d'inégalités valides, de renforcement et/ou de reformulation applicables à des programmes linéaires en nombres entiers généraux susceptibles d'être rencontrés dans des applications industrielles concrètes. Notre intérêt s'est porté en particulier vers l'utilisation d'inégalités coupantes plus coûteuses à établir en temps de calcul mais prenant mieux en compte la structure intrinsèque du problème (parce qu'exprimant par exemple l'intégrité de plusieurs variables du problème simultanément). Nous avons essayé d'aller dans ce sens par plusieurs voies distinctes, en séparant sur les polaires ou les restrictions des polaires des problèmes considérés, en essayant de mieux utiliser les oracles de séparation dont on dispose, ou en séparant des inégalités non valides qui demandent

un effort supplémentaire d'énumération. Les méthodes considérées nécessitant parfois un temps de calcul trop important, on considèrera aussi l'utilisation de méthodes pour résoudre les problèmes de séparation de manière approchée.

Dans un premier chapitre, nous rappellerons plus en détail les principales classes d'inégalités valides couramment utilisées. Nous présenterons ensuite quelques problèmes spécifiques soit purement théoriques, soit provenant d'applications industrielles, qui ont motivé notre intérêt pour des approches générales et que nous traiterons par la suite dans nos expériences de calculs.

Dans le chapitre 2, nous ferons une présentation unifiée des principales méthodes permettant de générer des inégalités valides pour un polyèdre dont on connaît uniquement une description complète dans un espace de dimension supérieure à l'espace original du problème. Nous commencerons par nous intéresser au problème de la séparation de la meilleure inégalité valide. Nous formulerons le problème de séparation sous la forme d'un problème d'optimisation, en discutant les différents critères d'optimisation et les conditions de normalisation. Dans un second temps nous aborderons, sous un angle général, les problèmes de renforcement d'une inégalité valide puis d'extension d'une inégalité valide uniquement sur une restriction du problème.

Dans le chapitre 3, nous présenterons une série de résultats d'expériences portant sur la séparation d'inégalités pour des problèmes de sac à dos multidimensionnels. Nous ferons une comparaison systématique de diverses méthodes basées sur des approches polaires : la méthode exacte séparant la "meilleure" inégalité, des méthodes basées sur des heuristiques et des relaxations mixtes. Nous présenterons une nouvelle méthode dite de "lifting relaxé" permettant d'obtenir des coupes de qualité très proche de la "meilleure" inégalité dans des temps très inférieurs.

Dans le chapitre 4, nous étudierons des méthodes de séparation basées sur des polyèdres décrits par la programmation disjonctive. Après avoir rappelé la formulation d'un programme disjonctif dans un espace de dimension supérieure et la méthode classique de *Lift and Project*, nous proposerons une méthode de décomposition pour réaliser la séparation relativement au polyèdre constitué par l'intersection des polyèdres de lift and project (aussi appelée fermeture élémentaire des coupes de lift and project). Nous proposerons une série de résultats comparatifs entre ces méthodes sur des problèmes de sac à dos multidimensionnels, de programmation mixte et de MAX-2SAT.

Dans le chapitre 5, nous étudierons des méthodes de séparation basées sur des polyèdres décrits par des techniques de reformulation-linéarisation de type Sherali-Adams. Nous proposerons une nouvelle méthode combinant de façon originale décomposition et relaxation pour optimiser sur la relaxation de Sherali-Adams. Cette méthode permet aussi de construire pratiquement tout un ensemble de relaxations de forces intermédiaires entre la fermeture élémentaire des coupes de lift and project et la relaxation de Sherali-Adams. Dans le cas où la relaxation de Sherali-Adams est trop coûteuse à calculer, l'approche proposée donne une méthode systématique pour renforcer, en fonction du temps de calcul disponible, la relaxation donnée par la fermeture élémentaire des coupes de lift and project. Nous donnerons des résultats d'expériences sur des problèmes mixtes généraux. Concernant l'application aux problèmes MAX2SAT (vus comme des cas particuliers de la maximisation d'une fonction pseudo-booléenne quadratique) nous établissons un résultat théorique montrant l'équivalence entre la relaxation de Sherali-Adams et la fermeture élémentaire des coupes de lift-and-project. Ce résultat théorique permet d'expliquer en partie les résultats expérimentaux obtenus précédemment.

Le chapitre 6, est consacré à l'étude de méthodes, assez différentes, basées sur l'utilisation d'inégalités *conditionnellement valides*. Après avoir rappelé quelques méthodes déjà connues d'utilisation de telles inégalités, nous proposerons plusieurs schémas algorithmiques nouveaux utilisant ce type d'inégalités et nous donnerons des résultats d'expériences sur des problèmes de sacs à dos multidimensionnels.

# Chapitre 1

## Introduction

Le renforcement de relaxations de programmes linéaires en nombres entiers par des approches polyédriques est maintenant une méthode de base pour la résolution pratique efficace de programmes linéaire en nombre entiers. Au cours des vingt dernières années de très nombreuses méthodes de génération d'inégalités valides ont été développées. Ces méthodes ont principalement deux provenances : d'un côté des inégalités valides portant sur des structures provenant de l'étude de problèmes combinatoires spécifiques ; de l'autre côté des inégalités générales le plus souvent basées sur la prise en compte de l'intégrité d'une ou de quelques variables.

Parmi ces nombreuses classes d'inégalités valides, quelques unes sont utilisées de manière efficace dans la résolution par énumération implicite ou Branch and Bound de problèmes réels en nombres entiers généraux. Souvent, les problèmes pratiques que l'on rencontre combinent plusieurs structures simples et une partie sans structure particulière. Dans ces cas, des classes d'inégalités valides permettent de générer des plans coupants relativement à certaines sous-structures sans relier ces sous-structures entre elles et sans prendre en compte la partie non structurée du problème. Les méthodes générales, elles, ne font pas d'hypothèse particulière sur des structures sous-jacentes. Enfin il faut mentionner que certaines méthodes générales n'ont jamais fait l'objet d'implémentation permettant de tester leur efficacité pratique sur des problèmes peu structurés.

Nous allons ici commencer par rappeler les principales sous-structures et les classes d'inégalités valides qui sont utilisés efficacement dans la plupart des solveurs actuels sur des problèmes entiers ou mixtes généraux. Ensuite nous donnerons l'exemple de quelques problèmes pratiques que l'on a pu rencontrer et qui combinent plusieurs sous-structures simples.

### 1.1 Principales inégalités valides utilisés dans les solveurs

Il existe de nombreux solveurs pour les programmes mixtes qui pour la plupart procèdent par énumération implicite avec ajout de plans coupants, soit uniquement à la racine de l'arbre d'énumération soit à n'importe quel nœud de l'arbre. Nous avons ici essayé de lister les principales classes d'inégalités valides utilisées dans trois de ces solveurs : CPLEX, XPRESSMP et COIN/Bcp. Cette liste n'est pas exhaustive et ne comprend pas en particulier les nombreuses classes d'inégalités utilisables pour des problèmes spécifiques comme celui du voyageur de commerce.

Comme on l'a dit ces inégalités sont principalement de deux types. Premièrement, on a des inéga-

lités portant sur des structures particulières, pour lesquelles les solveurs doivent essayer de reconnaître les structures avant d'essayer de générer des coupes. Ces inégalités sont celles de clique, de trou impair, de cover et de GUB-cover ; deuxièmement, des inégalités générales qui peuvent être séparées pour tout programme mixte principalement ici celles de Chvátal-Gomory et Gomory mixtes.

### 1.1.1 Structures de stables et contraintes d'exclusion binaires

Étant donné un graphe  $G = (V, E)$ , le problème de stable est celui de trouver un sous-ensemble de sommets non adjacents dans  $G$  de cardinalité maximale. Ce problème se modélise comme un problème 0/1 en définissant une variable par sommet du graphe de la façon suivante :

$$\left\{ \begin{array}{l} \max \sum_i x_i \\ t.q. \\ x_i + x_j \leq 1, \forall (i, j) \in E \\ x \in \{0, 1\}^n \end{array} \right.$$

Le polyèdre du stable a été très étudié et de nombreuses classes d'inégalités valides ont été mises en évidence. Nous considérerons uniquement ici des contraintes de clique et de cycle impair pour lesquelles les algorithmes de séparation sont suffisamment efficaces pour être utilisés dans des solveurs généraux.

Dans un problème 0/1 général, les structures de stables correspondent aux contraintes d'exclusion entre variables binaires du type :

$$x_i + x_j \leq 1.$$

On peut séparer les inégalités décrites ci-dessous en considérant le graphe dont les noeuds sont les variables 0/1 du problème et où une arête lie deux noeuds s'il y a une contrainte d'exclusion entre les deux variables.

#### Coupes de cliques

On appelle clique un ensemble de variables  $C = \{x_{i_1}, x_{i_2}, \dots, x_{i_k}\}$  tel qu'il existe une contrainte d'exclusion entre chaque couple de variables. Pour toute clique l'inégalité suivante est valide [50] :

$$\sum_{i \in C} x_i \leq 1$$

Une clique  $C$  est maximale s'il n'existe pas de clique contenant  $C$ . Si une clique n'est pas maximale, l'inégalité valide induite par cette clique est dominée par toute inégalité valide issue d'une clique la contenant. Il est à noter que si l'on peut générer toutes les cliques maximales d'un problème (ou toutes les cliques de cardinalité  $k$  et toutes les cliques maximales de cardinalité strictement inférieure à  $k$ ), alors on peut éliminer les contraintes d'exclusion binaires du problème original. En général, le problème d'énumérer toutes les cliques maximales est difficile et leur nombre peut être exponentiel en  $n$ . Mais dans certains cas particuliers (par exemple si le graphe induit par les contraintes d'exclusion est un graphe d'intervalles) le nombre de cliques est polynomial et peut être généré en temps polynomial.

Dans les solveurs le problème de trouver des coupes de cliques à partir des contraintes d'exclusions est en général résolu par des algorithmes heuristiques.

### Coupes de cycle impair (odd hole inequality)

Un cycle impair est un ensemble ordonné de variables  $C = \{x_{i_1}, x_{i_2}, \dots, x_{i_{2k+1}}\}$  tel qu'il existe une contrainte d'exclusion entre deux éléments consécutifs  $x_{i_j}$  et  $x_{i_{j+1}}$  et une contrainte d'exclusion entre  $x_{i_1}$  et  $x_{i_{2k+1}}$ . Pour tout cycle impair l'inégalité :

$$\sum_{i \in C} x_i \leq k$$

est valide. On dit qu'un cycle impair est sans corde si pour tout  $x_i \in C$  et  $x_j \in C$  non consécutifs (i.e. tels que  $|j - i| \not\equiv 1 \pmod{2k+1}$ ) il n'existe pas de contrainte d'exclusion entre  $x_i$  et  $x_j$ . Les contraintes de cycle impair sans corde dominent les contraintes de cycle impair avec cordes.

### 1.1.2 Structure de sac à dos

Le problème du sac à dos est un problème 0/1 qui consiste à optimiser une fonction de coût linéaire à coefficients positifs avec une seule contrainte linéaire à coefficients positifs :

$$(KP) \begin{cases} \max c^T x \\ t.q. \\ a^T x \leq b \\ x \in \{0, 1\}^n \end{cases}$$

où  $c \in \mathbb{Z}_+^n$ ,  $a \in \mathbb{Z}_+^n$ ,  $b \in \mathbb{Z}_+$ .

### Inégalités de covers et de covers étendues [11],[57]

Soit  $C \subset I = \{1, \dots, n\}$ . On dit que  $C$  est un ensemble dépendant si  $\sum_{i \in C} a_i > b$ . Si  $C$  est un ensemble dépendant l'inégalité :

$$\sum_{i \in C} x_i \leq |C| - 1, \quad (1.1)$$

appelée inégalité de cover, est valide.

On dit que  $C$  est un ensemble dépendant minimal si  $\forall j \in C$ ,  $C \setminus \{j\}$  est un ensemble indépendant (i.e.  $\sum_{i \in C \setminus \{j\}} a_i \leq b$ ).

Si  $C$  est un ensemble dépendant minimal l'inégalité de cover étendue :

$$\sum_{i \in E(C)} x_i \leq |C| - 1, \quad (1.2)$$

où  $E(C) = C \cup \{i \in \{1, \dots, n\} \text{ tels que } a_i \geq a_j \forall j \in C\}$ , est valide.

Les inégalités de cover étendues peuvent être des facettes du polyèdre du sac à dos sous certaines conditions. En particulier, si  $C \subset \{1, \dots, N\}$  est partitionné en deux ensembles  $C_1$  et  $C_2$  tels que  $C_1 \neq \emptyset$  les inégalités de cover sont des facettes de :

$$S(C_1, C_2) = \{x \in \{0, 1\}^n : ax \leq b, x_i = 0 \forall i \notin C, x_i = 1 \forall i \in C_2\}.$$

Ce résultat permet de déduire qu'en lifant (voir paragraphe 2.6.1) l'inégalité de cover sur  $C_2$  et  $\{1, \dots, n\} \setminus C$  on obtient une facette du polyèdre du sac à dos.

Il est important de noter que l'on peut chercher une inégalité de cover séparant un point pour n'importe quelle contrainte d'un problème mixte 0/1, en éliminant les variables continues de la contrainte (en les fixant à leur borne supérieure ou inférieure suivant que leur coefficient est respectivement positif ou négatif), et en prenant le complément de variables binaires dont le coefficient est négatif.

La première implémentation d'un algorithme heuristique de séparation des inégalités de cover dans le cadre d'un branch and bound pour résoudre des problèmes généraux est due à Crowder Johnson et Padberg [21]. Avec cet algorithme, on résout un sac à dos continu pour essayer de trouver un ensemble dépendant  $C$  induisant une cover séparant l'optimum continu courant, puis (si on a trouvé une inégalité séparatrice)  $n - |C|$  problèmes de sac à dos continus pour effectuer un lifting approché. Cet algorithme de séparation est utilisé dans le cadre de problèmes comportant plusieurs contraintes en essayant de séparer une cover pour chaque contrainte du problème prise séparément.

Dans [52], Van Roy et Wolsey utilisent aussi une heuristique pour séparer des covers étendues.

Gabrel et Minoux [27] ont proposé et implémenté un algorithme exact de séparation des cover minimales étendues les plus violées par le critère du ratio, puis lifting pour les problèmes de sac à dos. Avec cet algorithme on doit résoudre plusieurs problèmes de sacs à dos 0/1 de dimensions réduites pour séparer la cover minimale étendue. On peut ensuite effectuer  $n - |E(C)|$  opérations de lifting pour renforcer cette inégalité, mais les expériences de calcul effectués dans [27] montrent que ceci modifie très peu la cover minimale étendue.

Gu, Nemhauser et Padberg [34] ont exploré l'utilisation des inégalités de cover dans le cadre d'algorithmes de branch and cut. Une des originalités de leur approche est d'effectuer le lifting éventuellement sur des variables appartenant à l'ensemble dépendant qui définit la cover.

**(1-k) configurations** Une autre catégorie d'inégalités valides pour le problème du sac à dos sont les (1-k)-configurations mises en évidence par Padberg [49].

Soient  $C \subseteq \{1, \dots, n\}$  non vide et  $z \in \{1, \dots, n\} \setminus C$ . L'ensemble  $C \cup \{z\}$  est une (1-k)-configuration si :

$$\sum_{i \in C} a_i \leq b$$

et  $\{z\} \cup K$  est une cover minimale pour tout  $K \subseteq C$  tel que  $|K| = k$ . Alors, si  $C \cup \{z\}$  est une (1-k)-configuration, l'inégalité :

$$(|C| - k + 1)x_z + \sum_{i \in C} x_i \leq |C|$$

est valide. Remarquons que si  $k = |C|$  l'inégalité est simplement une cover, et que sinon l'inégalité de (1-k)-configuration résulte simplement du lifting de la variable  $z$  pour l'inégalité  $\sum_{i \in C} x_i \leq |C|$ .

Crowder, Johnson et Padberg, dans [21], ont aussi implémenté un algorithme heuristique de séparation des (1-k)-configurations.

### 1.1.3 Contraintes de semi-affectation et de sac à dos

On s'intéresse ici à un ensemble de solutions réalisables constitué d'une contrainte de sac à dos combinée avec des contraintes de semi-affectation. Soit  $I$  l'ensemble des variables, on a une couverture de  $I$  par des sous-ensembles disjoints  $Q_1, Q_2, \dots, Q_k$ . L'espace des solutions réalisables s'exprime alors comme :

$$X = \{x \in \{0, 1\}^n : a^T x \leq b \text{ et } \sum_{j \in Q_i} x_j \leq 1 \forall i = 1, \dots, k\}$$

On appelle GUB-cover un sous-ensemble  $C$  de  $I$  tel que  $C$  est une cover pour la contrainte de sac à dos, et  $C$  ne contient pas plus d'un élément de chaque  $Q_i$ . Pour toute GUB-cover  $C$  l'inégalité :

$$\sum_{i \in C} x_i \leq |I| - 1 \quad (1.3)$$

appelée GUB-cover est valide [56].

### 1.1.4 Coupes de Chvátal-Gomory

Soit un ensemble de point entiers  $X = \{x \in \mathbb{Z}^n : Ax \leq b, x \geq 0\}$ , avec  $A \in \mathbb{Z}^{m \times n}$  et  $b \in \mathbb{Z}^m$  une matrice et un vecteur entiers. Pour tout  $u \in \mathbb{R}^m$  tel que  $u \geq 0$ , l'inégalité

$$\lfloor uA \rfloor x \leq \lfloor ub \rfloor,$$

appelée inégalité de Chvátal, [18], est valide ( $\lfloor uA \rfloor x \leq uAx \leq ub$  est valide comme  $u \geq 0$ , et comme  $\lfloor uA \rfloor x$  est entier on a  $\lfloor uA \rfloor x \leq \lfloor ub \rfloor$ ).

Si on reformule  $X$  en un système d'équations par l'ajout de variables d'écart entières  $x_{n+1}, \dots, x_{n+m}$  comme étant  $\{x \in \mathbb{Z}_+^{n+m}; (A, I)x = b\}$ , pour tout  $v \in \mathbb{R}^m$ , l'inégalité :

$$vb - \lfloor vb \rfloor - (vA - \lfloor vA \rfloor)x + (v - \lfloor v \rfloor)z \leq 0,$$

appelée inégalité de Gomory, [30] [31] [32], est valide (comme le terme gauche de l'inégalité est un entier qui est strictement inférieur à 1).

On peut démontrer que ces deux classes d'inégalités sont équivalentes (cf. [45] par exemple), et nous les appellerons donc dans la suite inégalités de Chvátal-Gomory.

Le problème de séparation des inégalités de Chvátal-Gomory a été récemment démontré comme étant NP-complet par Eisenbrand [24]. Cependant Gomory [30] [31] [32] a donné une méthode pour déterminer une coupe séparant un point extrême du polyèdre de la relaxation continue de  $X$  (les solutions base réalisables) qui n'est pas un point entier. Si, pour une solution base réalisable  $\bar{x}$ , on note  $B$  l'ensemble des indices des variables en base et  $N$  l'ensemble des indices des variables hors base, le système d'équations définissant  $X$  peut se reformuler comme :

$$x^B + \bar{A}^N x^N = \bar{b}.$$

L'équation  $x_i^B + \sum_{j=1}^{|N|} \bar{a}_{ij} x_j^N = \bar{b}_i$  conduit alors à l'inégalité de Chvátal-Gomory :

$$\bar{b}_i - \lfloor \bar{b}_i \rfloor - \sum_{j=1}^{|N|} (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j^N \leq 0$$

Or si  $\bar{x}_i^B$  n'est pas entier cette inégalité coupe  $\bar{x}$  comme on a d'une part  $\bar{x}^N = 0$  et d'autre part  $\bar{x}_i^B = \bar{b}_i$  et  $\bar{b}_i - \lfloor \bar{b}_i \rfloor > 0$ .

L'algorithme pour la programmation entière de Gomory [32] permet de trouver la solution optimale d'un programme linéaire en nombres entiers en un nombre fini d'applications de la méthode de séparation décrite ci-dessus (utilisée de manière appropriée et en combinaison avec une version modifiée de l'algorithme du simplexe).

Dans l'étude des coupes de Chvátal-Gomory, on s'intéresse à la fermeture élémentaire de ces coupes. Pour un polyèdre  $Q \subset \mathbb{R}_+^n$  la fermeture élémentaire des coupes de Chvátal-Gomory notée  $Q'$  est l'ensemble des points vérifiant toutes les coupes de Chvátal-Gomory directement issues du système d'inégalités définissant  $Q$ . Par exemple si  $Q = \{x \in \mathbb{Z}^n : Ax \leq b, x \geq 0\}$  est le polyèdre résultant de la relaxation continue de  $X$ , on a

$$Q' = \{x \in \mathbb{R}^n : \forall u > 0, \lfloor uA \rfloor x \leq \lfloor ub \rfloor\}$$

Schrijver [53] a démontré que :

**Théorème 1.1** *Si  $Q$  est un polyèdre rationnel,  $Q'$  est un polyèdre rationnel*

Si on définit maintenant une suite de polyèdres dont chacun est la fermeture élémentaire du précédent, en partant de la relaxation continue :

$$\begin{aligned} Q_0 &= \{x \in \mathbb{R}^n : Ax \leq b\} \\ Q_1 &= Q'_0 \\ Q_{t+1} &= Q'_t \end{aligned}$$

on peut démontrer le théorème suivant dû à Chvátal[18] pour les polytopes et généralisé par Schrijver[53] pour les polyèdres :

**Théorème 1.2** *Il existe un entier  $k$  tel que :  $Q_k = X$*

Le plus petit  $k$  tel que  $Q_k = X$  est communément appelé rang de Chvátal. Celui-ci est utilisé pour étudier les polyèdres combinatoires, et la qualité de certaines coupes.

Malgré le grand nombre de résultats théoriques qui on pu être dérivés grâce aux coupes de Chvátal-Gomory, celles-ci sont peu utilisées pour la résolution algorithmique de programmes en nombres entiers (ni au travers de l'algorithme de Gomory, ni comme plans coupants ajoutés à la relaxation). Une des causes en est que ces coupes ne sont pas applicables aux programmes en nombres entiers mixtes (comportant des variables fractionnaires). Les coupes mixtes de Gomory que nous allons maintenant étudier constituent une famille de coupes plus fortes et qui permettent de résoudre ce problème.

### 1.1.5 Coupes mixtes de Gomory

On considère ici un ensemble  $Q$  de points de  $\mathbb{R}^n$  défini par des contraintes mixtes :  $Q = \{(x, y) \in \mathbb{R}^n : Ax + By = b \text{ et } x_i \in \mathbb{Z}\}$ . Pour tout  $u \in \mathbb{R}^m$ , on note  $f_0$  la partie fractionnaire de  $ub$  ( $f_0 = ub - \lfloor ub \rfloor$ ) et  $f_i$  la partie fractionnaire de  $(uA)_i$ . L'inégalité :

$$\sum_{i:f_i \leq f_0} f_i x_i + \frac{f_0}{1-f_0} \sum_{i:f_i > f_0} (1-f_i)x_i + \sum_{i:(uB)_i \geq 0} (uB)_i y_i + \frac{f_0}{1-f_0} \sum_{i:(uB)_i < 0} (uB)_i y_i \geq f_0 \quad (1.4)$$

appelée coupe mixte de Gomory est valide[29]. Les coupes mixtes de Gomory mixtes sont plus fortes et contiennent les coupes de Chvátal-Gomory dans le cas des problèmes entiers. Comme pour celles-ci le problème général de séparation est NP-complet, mais est soluble en temps polynomial pour les solutions de base de la relaxation linéaire.

Notons que les coupes générées pour les solutions de bases par la méthode classique de Gomory sont équivalentes aux coupes disjonctives simples renforcées que nous étudierons au chapitre 4, et que les coupes générales sont équivalentes aux coupes disjonctives et aux coupes MIR [44].

Comme pour le cas entier, Gomory a pu établir un algorithme de résolution exacte des problèmes mixtes basé sur la séparation de ces coupes dans les années 60 . Cependant c'est surtout depuis une dizaine d'années que ces coupes ont été utilisées pour la résolution de problème par branch & cut. Suite aux premiers travaux de Balas *et al.*, étudiant l'utilisation de ces coupes dans un tel cadre [6], de nombreux travaux théoriques et pratiques ont été conduits pour améliorer qualitativement l'efficacité des algorithmes de séparation [38], [15], [20]. Les coupes dérivées de solutions de bases sont incluses dans la plupart des solveurs de programmation mixte.

## 1.2 Exemples de problèmes d'application

Nous décrivons dans ce paragraphe quelques problèmes provenant d'applications diverses pouvant s'exprimer naturellement comme des programmes mathématiques mixtes et utilisant les deux structures de base introduites précédemment (contraintes de type "sac à dos", contraintes de type "stable") et des structures de type "semi-affectation" et "set packing".

### 1.2.1 Problème de planification des arrêts de sites de production à EDF

Ce problème consiste à planifier la production électrique et les arrêts de production pour entretien de  $n$  sites de production électrique sur une période de temps donnée  $T$ . Les contraintes du problème sont de satisfaire la demande d'électricité à chaque pas de temps, de satisfaire un certain nombre de conditions portant sur l'espacement des arrêts pour entretien, et de tenir compte des ressources limitées pour effectuer l'entretien des sites de productions.

Il existe plusieurs modélisations plus ou moins fines de ce problème par programmation mathématique, dont une a été présentée par F. Fourcade, E.L. Johnson et al dans [26]. La version que nous allons donner ici comporte plus de contraintes que celle présentée dans [26] mais est légèrement simplifiée par rapport au problème réel dans le sens où chaque site de production s'arrête une fois et une seule durant l'étude pendant  $\delta$  pas de temps consécutifs. Le problème peut se modéliser comme un programme linéaire mixte avec des variables continues  $u_i(t) \in [0, 1]$  donnant la production du site  $i$  au pas de temps  $t$  en pourcentage de sa puissance nominale, et des variables booléennes  $a_i(t)$  valant 1 si le site  $i$  commence un arrêt au pas de temps  $t$ . Des contraintes de semi-affectation indiquent qu'un arrêt et un seul a lieu pour chaque site :

$$\sum_{t=1}^T a_i(t) = 1, \forall i = 1, \dots, n \quad (1.5)$$

La contrainte imposant que la production soit nulle pendant la période de  $\delta$  pas de temps suivant le

début de l'arrêt est donnée par une contrainte de type borne supérieure généralisée :

$$u_i(t) + \sum_{t'=t-\delta}^t a_i(t') \leq 1, \forall t = 1, \dots, T \quad (1.6)$$

(où  $\delta$  est la durée de l'arrêt) La contrainte de demande est modélisée par une contrainte portant uniquement sur les variables continues du problème :

$$\sum_{i=1}^n u_i(t) \geq D_t, \forall t = 1, \dots, T \quad (1.7)$$

Une contrainte bornant la puissance indisponible (la puissance cumulée des tranches arrêtées) à chaque pas de temps donne une série de contraintes de type sac à dos :

$$\sum_i \sum_{t'=t-\delta}^t p_i a_i(t') \leq P, \forall t = 1, \dots, T \quad (1.8)$$

La quasi-totalité des contraintes de ressources portant sur le placement des arrêts se modélise comme un ensemble de contraintes de type stable données dans la formulation de base du problème comme des contraintes de cliques (non nécessairement maximales) de la forme :

$$\sum_{i,t} a_i(t) \leq 1 \quad (1.9)$$

Quelques-unes des contraintes de ressources ont une forme plus générale de type "set-packing" :

$$\sum_{i,t} a_i(t) \leq k \quad (1.10)$$

Une instance réelle de ce problème peut comporter typiquement de l'ordre de 15000 variables et 5000 contraintes. Expérimentalement, on observe que le traitement algorithmique fait par le solveur CPLEX avec l'activation de toutes les coupes ne permet de générer qu'un faible nombre de coupes de cliques.

## 1.2.2 Problème de dimensionnement de réseau avec fonctions de coûts en escaliers

Un réseau est donné par un graphe non orienté  $G = (N, U)$ , et des demandes  $d_k$  de communications entre des couples de nœuds  $(s(k), t(k))$  du graphe. Le problème consiste à installer des capacités sur les arêtes du graphe de manière à rendre possible l'acheminement des communications entre tous les couples de sommets. Le coût total du réseau correspond à la somme des coûts d'installation de capacité sur chaque arête ; ce coût est donné dans le cas présent par une fonction croissante discrète en escalier  $\phi_u$  ayant  $q_u$  points de discontinuité. Le problème de dimensionnement de réseau est de trouver une installation de capacités suffisante pour qu'il existe un routage de toutes les demandes de communication et de coût minimum.

Nous allons donner une des modélisations possibles de ce problème comme un programme linéaire en variables binaires (pour plus de précisions sur la modélisation et la résolution de ces problèmes on peut se reporter à [35], [36],[42]). Pour chaque arête du graphe la fonction de coût est donnée par un ensemble ordonné de points de discontinuité  $V_u = \{v_u^0, v_u^1, \dots, v_u^{q_u}\}$  (tels que  $0 = v_u^0$  et  $v_u^i < v_u^{i+1}$ ) et par la valeur de la fonction sur l'intervalle  $]v_u^i, v_u^{i+1}]$ , soit  $\gamma_u^i$ .

Une modélisation possible utilise des variables de décision binaires  $x_u^i$  définies pour chaque arête du

graphe  $u \in U$  et chaque point de discontinuité  $i = 0, \dots, q_u$  qui indiquent l'installation de la capacité  $v_u^i$  sur l'arête  $u$ . La fonction de coût à minimiser est donc :

$$\sum_{u \in U} \sum_{i=0}^{q_u} \gamma_u^i x_u^i.$$

On a des contraintes de semi-affectation portant sur les variables de chaque arête :

$$\sum_{i=0}^{q_u} x_u^i = 1, \forall u \in U.$$

L'expression du fait que l'installation de capacité est suffisante pour le routage des demandes s'effectue au travers de contraintes (correspondant à des rayons extrêmes du cône métrique) de la forme :

$$\sum_{u \in U} \lambda_u \left( \sum_{i=0}^{q_u} v_u^i x_u^i \right) \geq \theta(\lambda)$$

(où  $\lambda_u \geq 0$  et  $\theta(\lambda)$  est le coût du réseau de coût minimal lorsque les fonctions de coût sont linéaires de coefficient  $\lambda_u$  sur chaque arête). Notons que, dans ces contraintes, tous les coefficients sont positifs ce qui induit une structure similaire à celle de contraintes de sac à dos (il suffit de compléter toutes les variables pour retrouver la structure de sac à dos).

### 1.2.3 Problème de l'affectation des personnels navigants dans une compagnie aérienne

Pour effectuer un programme de vols sur une période de temps donnée (une semaine, un mois) une compagnie aérienne doit affecter à chaque vol des personnels navigants. Nous présentons dans ses grandes lignes le modèle proposé dans [59] pour une application correspondant à la société TUNISAIR (pour une présentation plus complète et détaillée on se reportera à l'article cité). On notera  $P$  l'ensemble des personnels à affecter (pilotes et copilotes par exemple) ;  $V$  l'ensemble des vols à assurer ;  $PS$  l'ensemble des périodes de service (sous-ensemble de vols pouvant être assurés par une même personne dans le respect de diverses contraintes règlementaires) ;  $PS(v)$  le sous-ensemble des périodes de service contenant un vol  $v$  donné ;  $PS(s)$  le sous-ensemble des périodes de service relatives à une semaine  $s$  donnée et  $PS(m)$  le sous-ensemble des périodes de service relatives à un mois donné. L'objectif à minimiser est une somme des coûts des heures supplémentaires et des coûts de "mise en place" (un coût de "mise en place" doit être payé par la compagnie chaque fois qu'un personnel doit débiter ou terminer son service en un lieu distinct de sa base de rattachement). L'objectif s'exprime comme une fonction linéaire. La modélisation du problème comporte des variables binaires  $x_{p,v}$  définies pour tout personnel  $p$  et tout vol  $v$  et valant 1 si le personnel  $p$  est affecté au vol  $v$  et 0 sinon,  $X_{p,ps}$  définies pour tout personnel  $p$  et toute période de service  $ps$  et valant 1 si le personnel  $p$  est affecté à la période de service  $ps$  et des variables continues  $HS_p$  donnant le nombre d'heures supplémentaires réalisées par le personnel  $p$ .

Les principales contraintes du problèmes sont les suivantes :

$$\sum_{p \in P} x_{p,v} = 1, \forall v \in V \quad (1.11)$$

$$x_{p,v} - \sum_{ps \in PS(v)} X_{p,ps} = 0, \forall v \in V \text{ et } \forall p \in P \quad (1.12)$$

$$\sum_{ps \in PS(s)} TV_{ps} X_{p,ps} \leq T_{max}, \forall \text{semaine } s \text{ et } \forall p \in P \quad (1.13)$$

$$X_{p,ps'} + X_{p,ps} \leq 1, \forall ps \text{ et } ps' \text{ incompatibles et } \forall p \in P \quad (1.14)$$

$$X_{p',ps} + X_{p,ps} \leq 1, \forall p \text{ et } p' \text{ incompatibles et } \forall ps \in PS \quad (1.15)$$

$$\sum_{ps \in PS(m)} TC_{ps} X_{p,ps} - HS_t \leq T_{stat}, \forall p \in P \text{ et } \forall \text{mois } m \quad (1.16)$$

Les contraintes (1.11) sont des contraintes de semi-affectation exprimant le fait que chaque vol  $v$  doit être affecté à un personnel et un seul.

Les contraintes (1.12) expriment le couplage entre les variables  $X_{p,ps}$  et les variables  $x_{p,v}$  : un personnel  $p$  ne peut être affecté à un vol  $v$  que s'il est affecté à une période de service  $ps$  contenant ce vol.

Les contraintes (1.13) sont de type "sac à dos" et expriment le fait que pour chaque personnel  $p$  le temps de vol hebdomadaire ne doit pas excéder une valeur fixée  $T_{max}$ .

Les contraintes (1.14) et (1.15) sont des contraintes de type "stable". Souvent très nombreuses en pratique, elles expriment soit l'impossibilité d'affecter un même personnel à deux périodes de services incompatibles, soit l'impossibilité d'affecter deux personnels incompatibles à une même période de service.

Enfin les contraintes (1.16) sont celles qui permettent de définir les heures supplémentaires  $HS_p$  pour tous les personnels  $p \in P$ . Les heures supplémentaires interviennent pour un personnel lorsque son temps de travail mensuel dépasse le temps de travail statutaire  $T_{stat}$  donné.

On voit donc que le problème est ainsi formulé comme un programme linéaire mixte dont une part importante des contraintes est une combinaison de contraintes d'affectation, de contraintes de stable et de sac à dos.

#### 1.2.4 Le problème du sac à dos multidimensionnel

Le problème du sac à dos multidimensionnel est la généralisation du problème de sac à dos dans laquelle plusieurs contraintes portent sur les objets à placer dans le sac. En notation matricielle le problème du sac à dos multidimensionnel peut se formuler comme :

$$(MKP) \begin{cases} \max c^T x \\ t.q. \\ Ax \leq b \\ x \in \{0, 1\}^n \end{cases}$$

où  $c \in \mathbb{Z}_+^n$ ,  $A \in \mathbb{Z}_+^{m \times n}$ ,  $b \in \mathbb{Z}_+^m$ .

La matrice  $A$  est souvent pleine et sans structure particulière.

Ce problème a des applications directes comme par exemple la planification de projets sous contraintes de ressources multiples (financières, humaines, ...). Dans un tel contexte  $c_i$  représente le gain espéré pour la réalisation du projet  $i$ . Le terme  $a_{ij}$  de la matrice  $A$  représente la quantité de la ressource  $i$  utilisée pour réaliser le projet  $j$  et  $b_j$  représente la quantité disponible de la ressource  $j$ .

Ce problème apparaît aussi souvent comme une partie non structurée et difficile dans un problème plus grand, par exemple dans les problèmes de planification de production électrique (c.f. paragraphe

1.2.1), de dimensionnement de réseaux (c.f. paragraphe 1.2.2), ou d'affectation d'équipages (c.f. paragraphe 1.2.3) que nous avons présentés.

### 1.2.5 Les problèmes de satisfiabilité (SAT)

Les problèmes de satisfiabilité peuvent se modéliser simplement comme des systèmes d'inégalités linéaires en variables 0/1 comportant des contraintes de type "semi-affectation" et des contraintes de type "set-packing". Pour chaque littéral on a deux variables  $x_i$  et  $\bar{x}_i$  indiquant que la variable est fixée respectivement à 1 et à 0, et une contrainte de semi affectation  $x_i + \bar{x}_i = 1$ . Chaque clause induit une contrainte de set packing. Pour chaque clause du type :

$$\bigvee_{i \in P} x_i \bigvee_{j \in N} \bar{x}_j \quad (1.17)$$

on a une contrainte :

$$\sum_{i \in P} \bar{x}_i + \sum_{j \in N} x_j \leq |P| + |L| - 1 \quad (1.18)$$

### 1.2.6 Les problèmes MAX-SAT

Le problème MAX-SAT consiste à maximiser le nombre de clauses satisfaites dans une instance donnée de problème de satisfiabilité. Il se modélise comme un problème de satisfiabilité auquel on ajoute pour chaque contrainte une variable artificielle  $z_k$  :

$$\sum_{i \in P} \bar{x}_i + \sum_{j \in N} x_j - z_k \leq |P| + |L| - 1 \quad (1.19)$$

Pour un problème MAX-SAT simple l'objectif consiste à minimiser la somme des variables artificielles ; cependant on peut facilement modéliser des problèmes avec des poids quelconques sur les clauses en affectant des poids quelconques positifs aux variables artificielles.

Pour tous les problèmes évoqués ci-dessus, s'il est vrai qu'on dispose de techniques de séparation pour des sous-structures particulières prises individuellement, les inégalités obtenues sont souvent très insuffisantes pour renforcer significativement des modèles faisant intervenir simultanément un grand nombre de contraintes et une combinaison de plusieurs sous-structures. Ainsi la structure de ces modèles n'est peu ou pas prise en compte par les techniques connues de plans coupants.

Parmi les problèmes que nous avons présentés ici, nous nous intéresserons tout particulièrement dans nos expérimentations à celui du sac à dos multidimensionnel (chapitres 3,4 et 6), au problème de MAX-2SAT (chapitres 4 et 5), ainsi qu'à un certain nombre de problèmes mixtes généraux de la MIPLIB<sup>1</sup> (chapitres 4 et 5).

L'objectif du présent travail est d'étudier les diverses techniques disponibles, d'évaluer expérimentalement leur intérêt et leurs limites et de proposer des voies d'amélioration possibles.

---

<sup>1</sup>La MIPLIB est une librairie de programmes entiers et mixtes disponible à <http://www.caam.rice.edu/~bixby/miplib/miplib.html>



## Chapitre 2

# Méthodes de génération d'inégalités valides générales

L'objectif de cette partie est de proposer une présentation des diverses méthodes utilisables pour la génération d'inégalités valides sur un polyèdre  $X$  quelconque dont on ne connaît pas la description complète. Ceci est typiquement la situation rencontrée en programmation en nombres entiers ou mixtes lorsqu'on cherche une "bonne" approximation de l'enveloppe convexe des solutions réalisables. Nous nous efforcerons de donner une présentation générale permettant de placer dans un même cadre les méthodes de séparation par oracle (c.f. chapitre 3) dans l'espace naturel du problème et les méthodes de séparation par reformulation du problème dans un espace de dimension supérieure (c.f. chapitres 4 et 5).

Tout d'abord, nous rappellerons brièvement les principales notions utiles à l'étude des polyèdres. Nous nous intéresserons ensuite à la formulation du problème de séparation de la meilleure coupe relativement à un point extérieur donné. Étant donné un point  $x^* \notin X$ , ce problème consiste à chercher une inégalité d'équation  $\alpha^T x \leq \beta$  valide pour  $X$  et vérifiant  $\alpha^T x^* > \beta$ . Nous discuterons la formulation de ce problème comme un problème d'optimisation en insistant notamment sur les différentes conditions sous lesquelles on peut obtenir la meilleure inégalité coupante.

Dans un troisième temps, nous nous intéresserons au problème du renforcement d'une inégalité valide. Étant connue une inégalité valide coupante  $\alpha^T x \leq \beta$ , celui-ci consiste à transformer  $\alpha^T x \leq \beta$  en une inégalité valide coupante  $\alpha'^T x \leq \beta'$  plus forte.

Dans un quatrième temps, nous nous intéresserons au problème très proche de l'extension d'une inégalité valide. Ici on a une inégalité coupante  $\alpha^T x \leq \beta$  valide uniquement pour une restriction du polyèdre  $X$  qu'on veut utiliser pour obtenir une inégalité valide pour tout  $X$ .

Enfin, dans un dernier temps, nous donnerons deux exemples d'utilisation des techniques développées dans ce chapitre. La première est la technique bien connue du lifting séquentiel. La seconde, est la méthode développée par Aplegate, Bixby, Chvátal et Cook pour séparer des inégalités du voyageur de commerce, qui a la particularité d'utiliser toutes les méthodes développées dans ce chapitre.

## 2.1 Rappels d'algèbre linéaire et de théorie des polyèdres

Nous allons rappeler ici les principales définitions et les principaux résultats nécessaires à l'étude des polyèdres. Nous allons commencer par quelques rappels sur les ensembles convexes.

**Définition** Soit  $C$  un sous-ensemble de  $\mathbb{R}^n$ .  $C$  est un ensemble convexe si pour tout  $x^1$  et  $x^2$  dans  $C$  le segment reliant  $x^1$  à  $x^2$  est contenu dans  $C$  (autrement dit pour tout  $\lambda \in [0, 1]$ , le point  $\lambda x^1 + (1 - \lambda)x^2$  appartient à  $C$ ).

Soit  $(a, b) \in \mathbb{R}^{n+1}$  tel que  $a \neq 0$ , l'ensemble  $H = \{x \in \mathbb{R}^n : a^T x = b\}$  est appelé hyperplan de  $\mathbb{R}^n$ . Pour tout hyperplan  $H$ , on note  $H^+$  l'ensemble  $H^+ = \{x \in \mathbb{R}^n : a^T x \leq b\}$ . Etant donné un ensemble convexe  $C$  un point  $z$  et un hyperplan  $H$ , on dit que  $H$  sépare  $z$  de  $C$  si  $z \in H^+$  et  $C \not\subseteq H^+$  ou si  $z \notin H^+$  et  $C \subseteq H^+$ .

**Théorème 2.1** Soit  $C$  un ensemble convexe fermé. Pour tout  $z \notin C$  il existe un hyperplan séparant  $z$  et  $C$ .

Ainsi, un ensemble convexe fermé peut toujours être décrit par un ensemble (possiblement infini) d'hyperplans. Un polyèdre  $P$  est précisément un ensemble convexe qui peut être décrit par un ensemble fini d'hyperplans.  $P$  est donc un polyèdre si et seulement si il existe une matrice  $A \in \mathbb{R}^{m \times n}$  et un vecteur  $b \in \mathbb{R}^m$  tels que  $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ .

Nous allons maintenant nous intéresser à la définition de la dimension des polyèdres.

**Définition** les points de  $\mathbb{R}^n$   $x^1, \dots, x^k$  sont dits linéairement indépendants si l'unique solution de  $\sum_{i=1}^k \lambda_i x^i = 0$  est  $\lambda = 0$ .

**Définition** les points de  $\mathbb{R}^n$   $x^1, \dots, x^k$  sont dits affinement indépendants si l'unique solution de  $\sum_{i=1}^k \lambda_i x^i = 0, \sum_{i=1}^k \lambda_i = 0$  est  $\lambda = 0$ .

On peut voir que  $x^1, \dots, x^n$  sont affinement indépendants si et seulement si  $x^2 - x^1, \dots, x^n - x^1$  sont linéairement indépendants.

**Définition** Un polyèdre  $P$  est de dimension  $k$  si la cardinalité maximale d'un ensemble de points affinement indépendants de  $P$  est  $k + 1$ . Un polyèdre de dimension  $n$  est dit de pleine dimension.

On peut partitionner les lignes du système linéaire décrivant  $P$  en deux de la façon suivante. Soit  $(A^=, b^=)$  l'ensemble des lignes de  $(A, b)$  telles que pour tout  $x$  dans  $P$  on a  $A^= x = b^=$  et soit  $(A^{\leq}, b^{\leq})$  le reste des lignes du système. Ainsi les lignes de  $(A^=, b^=)$  sont celles correspondant aux inégalités vérifiées à l'égalité par tout point de  $P$ . On peut maintenant relier la dimension du polyèdre  $P$  au rang de la matrice constituée des lignes de  $(A^=, b^=)$ .

**Proposition 2.1** Si  $P \subseteq \mathbb{R}^n$ ,  $\dim(P) = n - \text{rang}(A^=, b^=)$  (par convention la dimension du polyèdre vide étant  $-1$ ).

Un polyèdre peut être décrit soit par un ensemble fini d'inégalités, soit par un ensemble fini de points appartenant au polyèdre. Dans la suite, nous allons d'abord nous intéresser à la description d'un polyèdre par un ensemble minimal d'inégalités, puis à la description par points extrêmes et rayons extrêmes.

Un hyperplan  $H$  est dit valide pour  $P$  si  $P \subseteq H^+$ . Si  $H$  est valide, on appelle  $F = H \cap P$  face de  $P$ . Si  $F$  est non vide on dit que la face est propre. Notons que  $F = H \cap P$  est un polyèdre

(comme c'est l'intersection de deux polyèdres), on peut donc calculer sa dimension qui est inférieure ou égale (elle est égale si l'équation décrivant  $H$  appartient à  $A^-, b^-$ ) à celle de  $P$ . Les faces de  $P$  de dimension  $\dim(P) - 1$  sont appelées facettes de  $P$ . La proposition suivante permet de déterminer à partir de la dimension des faces les lignes de  $(A, b)$  non nécessaires à la description de  $P$ .

**Proposition 2.2** Soit  $(a, b)$  une ligne de  $(A, b)$  et  $F$  la face issue de l'hyperplan décrit par  $(a, b)$ . Si  $\dim(F) < \dim(P) - 1$ ,  $(a, b)$  peut être retirée de la description matricielle de  $P$

D'après cette proposition, on peut voir qu'un système minimal d'inégalités et d'égalités décrivant  $P$  est donné par les inégalités décrivant chaque facette de  $P$ , et un ensemble minimal d'égalités décrivant  $(A^-, b^-)$ .

Nous allons maintenant nous intéresser à la description minimale de  $P$  par ses points réalisables.

**Définition** L'enveloppe convexe de  $X$  notée  $\text{conv}(X)$  est le plus petit ensemble convexe contenant  $X$ .

**Définition** Etant donné  $X \subseteq \mathbb{R}^n$ , le point  $x \in \mathbb{R}^n$  est une combinaison convexe de points de  $X$  si il existe un ensemble fini de points  $\{x_1, \dots, x_k\} \subseteq X$  et un vecteur  $\lambda \in \mathbb{R}_+^k$  tels que :  $\sum_{i=1}^k \lambda_i = 1$  et  $x = \sum_{i=1}^k \lambda_i x_i$ .

Si  $X$  est un ensemble fini,  $\text{conv}(X)$  est l'ensemble des points combinaisons convexes des points de  $X$ .

**Définition** Etant donné un polyèdre  $P$ ,  $x \in R^n$  est un point extrême de  $P$  si et seulement si il n'existe pas  $x^1$  et  $x^2 \in X$  avec  $x^1 \neq x^2$  tels que  $x$  est une combinaison convexe de  $x^1$  et  $x^2$ .

**Définition** Soit  $P^0 = \{r \in R^n : Ar \leq 0\}$ . Si  $P \neq \emptyset$ ,  $r \in P^0 \setminus \{0\}$  est un rayon de  $X$ .

**Définition** Etant donné un polyèdre  $P$ ,  $r \in R^n$  est un rayon extrême de  $P$  si et seulement c'est un rayon de  $P$  et qu'il n'existe pas de rayons de  $P$   $r^1$  et  $r^2$  avec  $r^1 \neq \lambda r^2$  (pour tout  $\lambda > 0$ ) tels que  $r$  est sur le segment reliant  $r^1$  à  $r^2$ .

**Proposition 2.3**  $x$  est un point extrême de  $P$  si et seulement si c'est une face de dimension 0.  
 $r$  est un rayon extrême de  $P$  si et seulement si c'est une face de dimension 1 de  $X^0$ .

**Théorème 2.2 (Minkowsky)** Si  $P$  est un polyèdre non vide et  $\text{rang}(A) = n$

$$P = \text{conv}(\{x^1, x^2, \dots, x^k\}) + \text{cone}(\{r^1, \dots, r^l\})$$

où  $(\{x^1, x^2, \dots, x^k\})$  est l'ensemble des points extrêmes de  $X$ ,  $(\{r^1, \dots, r^l\})$  l'ensemble des rayons extrêmes, et  $\text{cone}(X)$  est l'ensemble des combinaisons positives de points de  $X$ .

## 2.2 Séparation d'inégalités valides générales

Nous allons commencer par donner la forme générique des polyèdres sur lesquels on va considérer le problème de séparation. On veut séparer des inégalités valides renforçant la relaxation linéaire d'un problème combinatoire de la forme :

$$(I) \begin{cases} \max c^T x \\ t.q. : \\ x \in X \end{cases}$$

(où  $X \subset \mathbb{R}^n$ ). La séparation s'effectue relativement à une relaxation du problème dont on connaît une description polyédrique complète dans un espace de dimension supérieure :

$$Q = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^q : Ax + By \leq b\} \quad (2.1)$$

(où  $A \in \mathbb{R}^{m \times n}$ ,  $B \in \mathbb{R}^{m \times q}$ , et  $b \in \mathbb{R}^m$ ). La relaxation de  $X$  étant donc donnée par le problème :

$$(II) \begin{cases} \max c^T x \\ t.q. : \\ x \in P_x(Q) \end{cases}$$

où  $X \subset P_x(Q)$  est la projection de  $Q$  sur l'espace des variables  $x$  :

$$P_x(Q) = \{x \in \mathbb{R}^n : \exists y \in \mathbb{R}^q \text{ tel que } Ax + By \leq b\}.$$

Ainsi le problème de séparer un point  $x^*$  est ramené à celui de trouver une inégalité valide pour  $P_x(Q)$  et coupant  $x^*$ .

Dans la suite, nous considérerons comme exemple en particulier la séparation exacte pour  $X$  (ou sur un ensemble combinatoire contenant  $X$ ). Dans ce cas, si on note  $\{x^1, \dots, x^k\}$  l'ensemble des points extrêmes de  $X$  et  $\{r^{k+1}, \dots, r^l\}$  l'ensemble des rayons extrêmes, une description complète dans un espace de dimension supérieure est donnée (par application du théorème de Minkowski-Weil) par le système suivant :

$$\begin{aligned} x - \sum_{i=1}^k y_i x^i - \sum_{i=k+1}^l y_i r^i &= 0 \\ \sum_{i=1}^k y_i &= 1 \\ y_i &\geq 0 \end{aligned} \quad (2.2)$$

D'autres exemples, qu'on traitera dans les chapitres suivants, de séparations à l'aide d'autres relaxations s'exprimant comme des polyèdres de dimensions supérieures sont la programmation disjonctive (chapitre 4) et la relaxation de type "Reformulation-Linéarisation" (chapitre 4.3.3).

### 2.2.1 Formulation du problème de séparation

Étant donné un point  $x^* \in \mathbb{R}^n$ , le problème de séparation d'une inégalité valide s'obtient à partir du problème de réalisabilité suivant : existe-t-il  $y \in \mathbb{R}^q$  vérifiant

$$By \leq b - Ax^* \quad (2.3)$$

Si ce problème n'a pas de solution alors  $x^*$  peut être coupé. Par le lemme de Farkas (voir par ex.[45]), ce système n'a pas de solution si et seulement si il existe  $u \in \mathbb{R}_+^m$  tel que

$$\begin{aligned} u^T b &< u^T Ax^* \\ u^T B &= 0 \\ u &\geq 0 \end{aligned} \quad (2.4)$$

et dans ce cas, si on prend  $\alpha = u^T A$  et  $\beta = u^T b$ , l'inégalité  $\alpha^T x \leq \beta$  est coupante (d'après l'expression du système) et valide (comme l'inégalité  $u^T Ax + u^T By \leq u^T b$  est trivialement valide et comme  $u^T B = 0$  d'après l'expression du système). Dans la suite on notera  $\Pi(B)$  le cône  $\{u \in \mathbb{R}^m : u^T B = 0, u \geq 0\}$ , appelé cône de projection sur  $x$  de  $Q$ , et  $\Pi(Q) = \{(\alpha, \beta) \in \mathbb{R}^{n+1} :$

$\exists u \in \mathbb{R}^m$  avec  $\alpha = u^T A, \beta = u^T b, u \in \Pi(B)$  l'ensemble des inégalités valides pour  $Q$  issues du cône de projection.

Considérons l'exemple de la séparation sur  $X$  avec la formulation (2.2). Comme la matrice  $(A, b)$  est l'identité, on a  $\Pi(B) = \Pi(Q)$  et le problème de séparation consiste à trouver  $\alpha \in \mathbb{R}^n$  et  $\beta \in \mathbb{R}$  tels que :

$$\begin{cases} \alpha^T x^* > \beta \\ \alpha^T x \leq \beta & \forall x \text{ point extrême de } X \\ \alpha^T r \leq 0 & \forall r \text{ rayon extrême de } X \end{cases} \quad (2.5)$$

Dans ce cas, on retrouve donc le problème de séparation sur le cône polaire.

### 2.2.2 Transformation du problème de séparation en un problème d'optimisation

Nous allons maintenant construire des problèmes d'optimisation dont la résolution donne des solutions de (2.4). La contrainte  $u^T b < u^T Ax^*$  peut être prise en compte par le biais de divers critères d'optimisation. Nous en étudierons principalement deux. Le premier exprime la différence entre le premier et le second membre de l'inégalité, le second le ratio des deux membres de l'inégalité.

En utilisant le premier critère on obtient le problème d'optimisation :

$$\begin{cases} \max u^T (Ax^* - b) \\ t.q. : \\ u \in \Pi(B). \end{cases}$$

Ce problème est, bien sur, soluble par un algorithme de programmation linéaire. Cependant, si  $x^*$  peut être séparé, le problème est non borné et si l'on veut obtenir une coupe maximisant le critère (l'inégalité la "plus" coupante) par des algorithmes de programmation linéaire il faut imposer une *condition de normalisation* restreignant l'ensemble des  $u$  admissibles à un sous-ensemble borné  $\mathcal{N}$ .

On obtient alors le problème d'optimisation :

$$\begin{cases} \max u^T (Ax^* - b) \\ t.q. : \\ u \in \Pi(B) \cap \mathcal{N}. \end{cases} \quad (2.6)$$

Nous discuterons dans le paragraphe 2.2.3 différentes conditions de normalisation possibles.

Nous allons maintenant nous intéresser au second critère que nous appellerons *critère du ratio*. Le problème de ce critère est que suivant le signe de  $u^T Ax^*$  les inégalités coupantes sont soit celles de ratio  $\frac{u^T b}{u^T Ax^*} < 1$  (si  $u^T Ax^* > 0$ ) soit celles de ratio  $\frac{u^T b}{u^T Ax^*} > -1$  (si  $u^T Ax^* < 0$ ), et que si  $u^T Ax^* = 0$  le problème n'est pas défini. Cependant on peut dériver des conditions fortes pour que le critère soit bien défini.

Supposons d'abord que  $u^T Ax^* > 0$ . Dans ce cas le problème d'optimisation à résoudre pour séparer  $x^*$  est :

$$\begin{cases} \min \frac{u^T b}{u^T Ax^*} \\ t.q. : \\ u \in \Pi(B). \end{cases}$$

On voit que l'avantage de ce critère par rapport au précédent est que dans cette formulation le coût objectif est le même pour tous les  $u$  situés sur un même rayon de  $\Pi(B)$ .

Comme toutes les inégalités sont identiques et ont même coût objectif à un coefficient multiplicateur positif près, on peut transformer ce problème en :

$$\left\{ \begin{array}{l} \min u^T b \\ \text{t.q. :} \\ u \in \Pi(B) \\ u^T A x^* = 1 \end{array} \right. \quad (2.7)$$

Nous allons examiner les conditions générales pour que ce problème soit bien défini. Nous étudierons d'abord les conditions sous lesquelles le problème (2.7) est réalisable et donne lieu à une inégalité coupante à l'optimum, ensuite les conditions sous lesquelles le problème est borné.

Une condition nécessaire et suffisante de réalisabilité du problème et d'existence d'une inégalité coupante est donnée par la proposition suivante.

- Proposition 2.4** 1. *Le problème (2.7) est réalisable si et seulement s'il existe  $M$  tel que,  $\forall \lambda \geq M$ ,  $\lambda x^* \notin X$ .*  
 2. *La solution optimale de (2.7) est coupante s'il existe  $M < 1$  tel que,  $\forall \lambda \geq M$ ,  $\lambda x^* \notin X$ .*

**Preuve :** 1. ( $\Rightarrow$ ) Soit  $u$  solution réalisable de (2.7). En choisissant  $M > u^T b$ , on a  $\forall \lambda \geq M$   $u^T A(\lambda x^*) = \lambda > \beta$ .

( $\Leftarrow$ ) Soit  $M$  tel que  $\forall \lambda \geq M$ ,  $\lambda x^* \notin X$ . Soit  $Y = \{x = \lambda x^* : \lambda \in \mathbb{R}\}$ . Clairement l'inégalité  $\frac{x^{*T} x}{\|x^*\|} \leq M$  est valide pour  $X \cap Y$ . D'après le théorème 2.7 (c.f. paragraphe 2.4 ci-après) sur l'extension d'inégalité valide, il existe une inégalité  $\alpha^T x \leq \beta$  telle que sur  $Y$   $\{x \in Y : \frac{x^{*T} x}{\|x^*\|} \leq M\} = \{x \in Y : \alpha^T x \leq \beta\}$ , et  $\beta = \alpha^T M x^*$ . Si  $\alpha^T x^* < 0$ ,  $\forall \lambda < M$ ,  $\alpha^T x^* \lambda > \alpha^T x^* M$ , ce qui contredit  $\{x \in Y : \frac{x^{*T} x}{\|x^*\|} \leq M\} = \{x \in Y : \alpha^T x \leq \beta\}$ . Si  $\alpha^T x^* = 0$ ,  $\forall \lambda$   $\alpha^T x^* \lambda > \alpha^T x^* M$ , ce qui contredit aussi l'équivalence de  $\alpha^T x \leq \beta$  et  $\frac{x^{*T} x}{\|x^*\|} \leq M$  sur  $Y$ .

2. ( $\Rightarrow$ ) L'inégalité est coupante si et seulement si  $\beta < 1$ . On peut donc prendre  $\beta < M < 1$ .

( $\Leftarrow$ ) Si  $M < 1$ , l'inégalité  $\frac{x^{*T} x}{\|x^*\|} \leq M$  est coupante sur  $Y$ . Il résulte du théorème 2.7 que l'inégalité  $\alpha^T x \leq \beta$  est coupante. ■

En supposant que  $x^*$  est le maximum pour une relaxation du problème, la proposition précédente permet de déduire une condition forte pour la réalisabilité de (2.7).

**Corollaire 2.1** *Si  $c^T x^* > 0$  et si  $x^*$  est l'optimum sur une relaxation contenant  $P_x(Q)$  le problème (2.7) est réalisable et son optimum, si il existe, est coupant.*

**Preuve :** En effet, s'il n'existe pas de  $M < 1$  tel que pour tout  $\lambda \geq M$  on a  $\lambda x^* \notin P_x(Q)$ , il existe  $\hat{\lambda} > 1$  tel que  $\hat{\lambda} x^* \in P_x(Q)$  et  $\hat{\lambda} c^T x^* > c^T x^*$ , ce qui contredit l'optimalité de  $x^*$ . ■

Le problème (2.7) est borné si et seulement s'il n'existe pas  $\hat{u} \in \Pi(B)$  tel que  $\hat{u}^T Ax^* = 0$  et  $\hat{u}^T Ax^* > \hat{u}^T b$  (i.e.  $\hat{u}^T b$  négatif). Si (2.7) est non borné, il suffit de trouver une solution du système d'inégalités suivant dont toutes les solutions ont un coût infini par le critère du ratio) :

$$\begin{cases} u^T Ax^* = 0 \\ u^T b = -1 \\ u \in \Pi(B) \end{cases}$$

**Remarque 2.1 :** Lorsque  $c^T x^* < 0$ , une inégalité coupante peut être obtenue par la résolution du problème :

$$\begin{cases} \max u^T b \\ t.q. : \\ u \in \Pi(B). \\ u^T Ax^* = -1 \end{cases}$$

||

### 2.2.3 Qualité de l'inégalité générée et conditions de normalisation

Nous allons étudier ici la qualité des inégalités générées lors de la résolution du problème (2.6) du point de vue théorique. En particulier nous allons nous intéresser aux conditions qui permettent de garantir que l'on génère des facettes de  $P_x(Q)$  (la projection du polyèdre  $Q$  sur les variables  $x$ ).

Nous rappellerons dans un premier temps la condition suffisante sur  $Q$  pour que les inégalités issues des rayons extrêmes de  $\Pi(B)$  soient des facettes de  $P_x(Q)$  donnée par Balas dans [5].

Comme on l'a vu, si l'on veut trouver une facette optimale selon le critère de la différence en utilisant un algorithme de programmation linéaire classique, il faut imposer une condition de normalisation. Nous nous intéresserons donc dans un deuxième temps aux différentes conditions de normalisation connues dans la littérature en étudiant plus particulièrement deux aspects :

- Les points extrêmes de  $\Pi(B) \cap \mathcal{N}$  sont-ils sur des rayons extrêmes de  $\Pi(B)$  (i.e. les solutions obtenues par l'optimisation de (2.6) peuvent-elles correspondre à des facettes de  $P_x(Q)$ ) ?
- Tout  $\Pi(B)$  est-il représenté, à un coefficient multiplicateur près, dans  $\Pi(B) \cap \mathcal{N}$  (i.e. pour tout rayon  $r$  de  $\Pi(B)$  existe-t-il un point  $x$  de  $\Pi(B) \cap \mathcal{N}$  tel que  $x = \lambda r$  avec  $\lambda \geq 0$ ) ?

#### Condition suffisante sur $Q$ pour que les rayons extrêmes de $\Pi(B)$ donnent des facettes de $P_x(Q)$

On appelle cône polaire de  $P_x(Q)$  l'ensemble des inégalités valides pour  $P_x(Q)$ . Les facettes de  $P_x(Q)$  sont données par les rayons extrêmes de son cône polaire. Nous allons chercher ici à déterminer sous quelles conditions pour un rayon extrême  $\hat{u}$  de  $\Pi(B)$  le point donné par  $(\hat{u}^T A, \hat{u}^T b)$  est un rayon extrême du cône polaire. Pour cela nous allons commencer par établir que le cône polaire est exactement l'ensemble des inégalités valides issues du cône de projection.

**Théorème 2.3**  $\Pi(Q)$  est le cône polaire de  $P_x(Q)$

**Preuve :** Soit  $(\alpha, \beta) \in \Pi(Q)$ . Comme on l'a montré au paragraphe précédent  $\alpha^T x \leq \beta$  est alors une inégalité valide pour  $P_x(Q)$ . Il nous suffit donc de montrer que si  $\alpha^T x \leq \beta$  est valide pour  $P_x(Q)$ ,  $(\alpha, \beta) \in \Pi(Q)$ .

Si  $\alpha^T x \leq \beta$  est valide pour  $P_x(Q)$ , l'inégalité  $\alpha^T x + 0y \leq \beta$  est valide pour  $Q$  et d'après le lemme de Farkas il existe donc  $v \geq 0$  tel que :

$$\begin{aligned}\alpha &= v^T A \\ 0 &= v^T B \\ \beta &= v^T b\end{aligned}$$

et  $(\alpha, \beta)$  appartient bien à  $P_x(Q)$ . ■

Donc  $(\hat{u}^T A, \hat{u}^T b)$  représente une facette si et seulement si c'est un rayon extrême de  $\Pi(Q)$ . On peut donc donner une condition nécessaire pour que les rayons extrêmes de  $\Pi(B)$  permettent d'obtenir des facettes. Le théorème qui suit énonce sous une forme légèrement différente un résultat établi dans Balas [5] :

**Théorème 2.4** *Si  $(A, b)$  (la matrice composée des colonnes de  $A$  et de  $b$ ) est de plein rang sur les lignes, alors l'inégalité  $(\hat{u}^T A, \hat{u}^T b)$  est une facette si et seulement si  $\hat{u}$  est un rayon extrême de  $\Pi(B)$ .*

**Preuve :** Commençons par montrer que si  $(\hat{u}^T A, \hat{u}^T b)$  est un facette  $\hat{u}$  est un rayon extrême de  $\Pi(B)$ . En effet dans ce cas  $(\hat{u}^T A, \hat{u}^T b)$  est un rayon extrême de  $\Pi(Q)$ , mais si  $\hat{u}$  n'est pas un rayon extrême de  $\Pi(B)$  il existe  $u^1 \neq \lambda \hat{u}$  ( $\lambda \geq 0$ ) et  $u^2$  dans  $\Pi(B)$  tels que  $u^1 + u^2 = \hat{u}$ . On a donc :

$$(\hat{u}^T A, \hat{u}^T b) = (u^{1T} A, u^{1T} b) + (u^{2T} A, u^{2T} b)$$

et  $u^{1T}(A, b) \neq \lambda \hat{u}^T(A, b)$  (sinon comme  $(A, b)$  est de plein rang sur les lignes  $u^1 = \lambda \hat{u}$ ). Ceci contredit donc le fait que  $(\hat{u}^T A, \hat{u}^T b)$  est un rayon extrême.

Supposons maintenant que  $\hat{u}$  est un rayon extrême de  $\Pi(B)$ . Supposons qu'il existe  $(\alpha^1, \beta^1) \neq \lambda(\hat{u}^T A, \hat{u}^T b)$  ( $\lambda \geq 0$ ) et  $(\alpha^2, \beta^2)$  dans  $\Pi(Q)$  tels que :

$$(\hat{u}^T A, \hat{u}^T b) = (\alpha^1, \beta^1) + (\alpha^2, \beta^2).$$

Comme  $(\alpha^1, \beta^1)$  est dans  $\Pi(Q)$ , il existe  $u^1$  dans  $\Pi(B)$  tel que  $(\alpha^1, \beta^1) = u^{1T}(A, b)$  (avec  $u^1 \neq \hat{u}$ ) et de même pour  $(\alpha^2, \beta^2)$  il existe  $u^2$  dans  $\Pi(Q)$  tel que  $(\alpha^2, \beta^2) = u^{2T}(A, b)$  et donc :

$$(\hat{u}^T A, \hat{u}^T b) = (u^{1T} A, u^{1T} b) + (u^{2T} A, u^{2T} b).$$

Maintenant comme  $(A, b)$  est de plein rang sur les lignes on a  $\hat{u} = u^1 + u^2$  ce qui contredit le fait que  $\hat{u}$  soit un rayon extrême de  $\Pi(B)$ . ■

### Conditions de normalisation

Différentes conditions de normalisation ont été étudiées dans la littérature dans le contexte de la programmation disjonctive [7] [4]. Nous allons reprendre ici ces conditions dans le contexte plus général de la séparation sur  $Q$ .

Nous en étudierons particulièrement quatre :

1. imposer  $\|u\|_1 \leq 1$
2. imposer  $\|u\|_\infty \leq 1$

3. borner  $\beta$

4. borner le produit scalaire  $\alpha^T(x^* - \bar{x})$  (où  $x^*$  est le point à couper et  $\bar{x}$  est un point de  $P_x(Q)$ ).

Pour introduire les conditions de normalisation, nous allons passer par le problème de réalisabilité (2.3). En fait, l'introduction de conditions de normalisation, correspond à la transformation de ce problème toujours irréalisable si  $x \notin P_x(Q)$  en un problème d'optimisation par l'introduction d'une ou plusieurs variables artificielles. Rappelons que l'on appelle variables artificielles des variables dont l'ajout permet de rendre non vide le polyèdre, et dont le coût est tel que le problème original est réalisable si et seulement si l'optimum du problème avec variables artificielles annule toutes les variables artificielles à l'optimum.

**normalisation**  $\|u\|_1 \leq 1$  La première condition de normalisation correspond à l'ajout d'une unique variable artificielle  $z$  :

$$\begin{cases} \min z \\ t.q. : \\ By - z\mathbf{1} \leq b - Ax^* \\ z \geq 0 \end{cases}$$

(où  $\mathbf{1}$  est le vecteur dont toutes les composantes sont 1). Ce problème est toujours réalisable et son optimum est nul s'il existe  $y$  tel que  $By \leq b - Ax^*$ . Le problème de séparation associé à ce problème de réalisabilité est son dual :

$$\begin{cases} \max u^T(b - Ax^*) \\ t.q. : \\ t.q. \\ u^T B = 0 \\ \sum_{i=1}^m u_i \leq 1 \\ u \geq 0 \end{cases}$$

La condition de normalisation est donc que la somme des multiplicateurs  $u_i$  soit inférieure ou égale à 1 (comme  $u \geq 0$  il s'agit d'une normalisation au sens de la norme  $L_1$  de  $u$ ). Sous cette condition les points extrêmes sont tous sur des rayons extrêmes étant donné que le cône  $\Pi(B)$  est intersecté par un hyperplan unique. De plus ce problème est toujours borné puisque son dual est toujours réalisable.

**Remarque 2.2 :** Comme, pour tout point extrême différent de 0, l'inégalité  $\sum_{i=1}^m u_i \leq 1$  est saturée on peut remplacer celle-ci par une égalité. ||

**normalisation**  $\|u\|_\infty \leq 1$  La seconde condition de normalisation correspond à l'introduction d'une variable artificielle par contrainte :

$$\begin{cases} \min \sum_{i=1}^m z_i \\ t.q. : \\ By - z \leq b - Ax^* \\ z \geq 0 \end{cases}$$

Là aussi, le problème est toujours réalisable et son optimum est nul s'il existe  $y$  tel que  $By \leq b - Ax^*$ . Le problème dual correspond au problème de séparation associé à ce problème de réalisabilité :

$$\begin{cases} \max u^T(b - Ax^*) \\ t.q. : \\ u^T B = 0 \\ 0 \leq u \leq 1 \end{cases}$$

La condition de normalisation est donc que chaque multiplicateur doit être inférieur à 1 (il s'agit donc d'une condition de normalisation au sens de la norme  $L_\infty$ ). Le problème est toujours borné, par contre ses points extrêmes ne sont pas forcément des rayons extrêmes du cône  $\Pi(B)$ . Par exemple, si  $\Pi(B)$  est l'ensemble des  $u \geq 0$  (c'est à dire si  $Q$  est juste une relaxation continue d'une formulation de  $X$ ), le vecteur composé uniquement de 1 est un point extrême de  $\Pi(B) \cap \mathcal{N}$  mais n'est pas un rayon extrême de  $\Pi(B)$ .

**normalisation**  $\beta \leq 1$  Les deux conditions de normalisation que nous venons de présenter correspondent à l'introduction de variables artificielles classiques dans la résolution d'un problème d'optimisation. Par contre les deux dernières que nous allons maintenant présenter sont véritablement spécifiques à la résolution de problèmes de séparation.

La première de ces conditions résulte du problème d'optimisation suivant :

$$\begin{cases} \min z \\ t.q. : \\ By - bz \leq b - Ax^* \\ z \geq 0 \end{cases} \quad (2.8)$$

L'optimum de ce problème est nul si et seulement si  $x^* \in P_x(Q)$ . Par contre, lorsque  $x^* \notin P_x(Q)$  ce problème n'est pas toujours réalisable. Le lemme suivant donne une condition nécessaire et suffisante pour que ce problème soit réalisable.

**Lemme 2.1** *Le problème (2.10) est réalisable si et seulement s'il existe  $\gamma \in ]0, 1]$  tel que  $\gamma x^* \in P_x(Q)$*

**Preuve :**

( $\Leftarrow$ ) Soit  $\gamma \in ]0, 1]$  tel que  $\gamma x^* \in P_x(Q)$  ; alors il existe  $\bar{y}$  tel que :

$$A\gamma x^* + B\bar{y} \leq b$$

Comme  $\gamma > 0$  on a :

$$Ax^* + B\frac{\bar{y}}{\gamma} \leq \frac{b}{\gamma}.$$

Maintenant comme  $\frac{1}{\gamma} \geq 1$  on peut prendre  $z$  tel que  $\frac{1}{\gamma} = 1 + z$  et on a :

$$Ax^* + B\frac{\bar{y}}{\gamma} - zb \leq b$$

qui est bien une solution réalisable de (2.10).

( $\Rightarrow$ ) Soit  $y^*$  et  $z^*$  constituant une solution réalisable de (2.10). On a  $\frac{x^*}{1+z^*} \in P_x(Q)$  et  $\frac{1}{1+z^*} \in ]0, 1]$  (comme  $z^* \geq 0$ ). ■

Le problème (2.10) est donc en particulier toujours réalisable si le point 0 est réalisable et strictement à l'intérieur du polyèdre (comme dans ce cas il existe toujours  $\epsilon > 0$  tel que  $\epsilon x^* \in P_x(Q)$ ). Ceci permet de déduire le résultat suivant, plus utile pour les problèmes combinatoires, cité par Balas *et al.*[7].

**Corollaire 2.2** [7] *En particulier si  $P_x(Q) \subset \mathbb{R}_+^n$  (c'est à dire si  $P_x(Q)$  est inclus dans l'orthant positif ce qui est par exemple en général le cas lorsque c'est la relaxation d'un problème 0/1), le problème (2.10) est toujours réalisable si  $P_x(Q)$  peut être remplacé par son antidominant  $\{x : \exists \bar{x} \in P_x(Q) \text{ avec } .x \leq \bar{x}\}$ .*

Le dual de (2.10) permettant de séparer est :

$$\begin{cases} \max u^T(b - Ax^*) \\ \text{t.q. :} \\ u^T B = 0 \\ -u^T b \leq 1 \\ u \geq 0 \end{cases} \quad (2.9)$$

La condition de normalisation consiste donc à borner supérieurement le second membre  $\beta$  de l'inégalité générée ( $\beta = -u^T b$ ). Les points extrêmes sont toujours sur des rayons extrêmes puisque le cône est intersecté par un hyperplan unique.

**Remarque 2.3 :** Une condition de normalisation similaire à la précédente consiste à considérer le problème :

$$\begin{cases} \min z \\ \text{t.q. :} \\ By + bz \leq b - Ax^* \\ z \geq 0 \end{cases} \quad (2.10)$$

Dans ce cas, le problème est réalisable s'il existe  $\gamma \in [1, +\infty[$  tel que  $\gamma x^* \in P_x(Q)$ . L'équivalent du corollaire est que le problème est réalisable si  $P_x(Q)$  peut être remplacé par son dominant  $\{x : \exists \bar{x} \in P_x(Q) \text{ avec } x \geq \bar{x}\}$ . La condition de normalisation dans le dual est :  $-u^T b \geq -1$ . ||

**normalisation**  $\alpha^T(\bar{x} - x^*) \leq 1$  Pour introduire la quatrième condition de normalisation nous avons besoin de connaître un point réalisable de  $P_x(Q)$  que nous notons  $\bar{x}$ . Dans ce cas, on introduit une variable artificielle  $z$  pour résoudre le problème suivant :

$$\begin{cases} \min z \\ \text{t.q. :} \\ By + z(A\bar{x} - Ax^*) \leq b - Ax^* \\ z \geq 0 \end{cases} \quad (2.11)$$

**Lemme 2.2** *Si  $\bar{x} \in P_x(Q)$ , le problème (2.11) est toujours réalisable et son optimum a un coût nul si et seulement si  $x^* \in P_x(Q)$ .*

**Preuve :** Le problème est toujours réalisable du fait que pour  $z = 1$  on obtient le système :

$$By + A\bar{x} \leq b$$

qui est toujours réalisable du fait que  $\bar{x} \in P_x(Q)$ .

L'optimum du problème est donné par le point combinaison convexe de  $x^*$  et  $\bar{x}$ ,  $x = (1 - z)x^* + z\bar{x}$  appartenant à  $P_x(Q)$  et le plus proche de  $x^*$ . Donc en particulier si  $x^* \in P_x(Q)$ , ce point est  $x^*$  lui-même. ■

En passant au dual on obtient le problème de séparation associé :

$$\begin{cases} \max u^T(b - Ax^*) \\ t.q. : \\ \quad u^T B = 0 \\ \quad u^T A(\bar{x} - x^*) \leq 1 \\ u \geq 0 \end{cases} \quad (2.12)$$

La condition de normalisation correspond donc à borner la projection du vecteur directeur de l'inégalité générée  $\alpha = u^T A$  dans la direction  $\bar{x} - x^*$ .

## 2.2.4 Résolution algorithmique du problème de séparation

Comme on l'a vu, le problème de séparation peut être en général résolu directement par programmation linéaire. Cependant, dans certains cas (comme par exemple l'optimisation directe sur  $X$  par l'intermédiaire de (2.2)), la matrice  $B$  du polyèdre  $Q$  comporte un nombre trop important de colonnes pour être écrite explicitement. Dans ce cas, si on dispose d'un oracle permettant de résoudre un problème d'optimisation sur  $Q$  où la fonction de coûts porte uniquement sur l'espace des variables  $x$ , le problème de séparation peut être résolu par décomposition de Benders en résolvant le problème-maître :

$$\begin{cases} \max u^T(b - A\bar{x}) \\ t.q. : \\ \quad u^T(Ax - b) \leq 0 \quad \forall x \in P_x(Q) \\ u \geq 0 \end{cases} \quad (2.13)$$

Avec comme sous-problème (pour  $u = \bar{u}$  optimum du problème maître) :

$$\begin{cases} \max_x \bar{u}^T(Ax - b) \\ t.q. : \\ \quad Ax + By \leq b \end{cases} \quad (2.14)$$

Par exemple, dans le cas de la séparation sur  $\text{conv}(X)$  par l'intermédiaire de la formulation (2.2), si  $P$  est un sous-ensemble des points extrêmes, et  $V$  un sous-ensemble des rayons extrêmes de  $\text{conv}(X)$ , le problème-maître est le programme linéaire suivant :

$$(SPP') \begin{cases} \max \alpha^T x^* - \beta \\ t.q. : \\ \quad \alpha^T x \leq \beta \quad \forall x \in P \\ \quad \alpha^T r \leq 0 \quad \forall r \in V \\ (\alpha, \beta) \in \mathcal{N} \end{cases}$$

**Algorithme 1** séparation sur  $\text{conv}(X)$  par décomposition

Soit  $P$  un sous-ensemble des points extrêmes de  $\text{conv}(X)$ .

Soit  $V$  un sous-ensemble des rayons extrêmes de  $\text{conv}(X)$ .

**répéter**

Trouver  $\alpha$  et  $\beta$  solution optimale de (SPP').

Résoudre  $\max \alpha^T x$  pour  $x \in X$

**si**  $\max \alpha^T x$  est non borné et que  $r$  est le rayon extrême déterminé **alors**

Ajouter le rayon extrême  $r$  à  $V$

**sinon**

Soit  $\hat{x}$  la solution optimale du problème  $\max\{\alpha^T x : x \in X\}$

**si**  $\alpha^T \hat{x} > \beta$  **alors**

ajouter  $\hat{x}$  à  $P$ .

**fin si****fin si**

**tant que** Un point ou un rayon extrême a été ajouté à la dernière itération

et on peut trouver l'inégalité solution de (SPP') en appliquant l'algorithme 1 ci-dessous.

Si, par l'utilisation de cette méthode, on peut économiser l'énumération complète des points et rayons extrêmes, la résolution efficace reste subordonnée à la facilité de résoudre le sous-problème  $\max \alpha^T x$  sur  $X$ . Remarquons ici que si ce sous-problème était facile à résoudre pour tout  $\alpha$ , alors le problème de départ (avec  $\alpha = c$ ) serait aussi facile à résoudre.

## 2.3 Renforcement d'une inégalité coupante

Une autre façon d'obtenir une inégalité valide coupante générale est de partir d'une inégalité coupante ne représentant pas une facette et d'augmenter progressivement sa dimension jusqu'à l'obtention d'une facette. D'après un théorème de Minkowski ([41] Section 19), ceci est théoriquement toujours possible.

**Théorème 2.5** Soit  $X$  un polytope de  $\mathbb{R}^d$ , soit  $x^* \notin X$  un point de  $\mathbb{R}^d$ , et soit  $\alpha^T x \leq \beta$  séparant  $x^*$  de  $X$ . Soit  $F = \{x \in X : \alpha^T x = \beta\}$ . Si  $\dim(F) \leq \dim(X) - 1$  (i.e.  $\alpha, \beta$  n'induit pas une facette du polytope), alors il existe une inégalité  $\alpha'^T x \leq \beta'$  séparant  $x^*$  définissant une facette telle que  $F \subset F' = \{x \in X : \alpha'^T x = \beta'\}$ .

Lorsque l'on a la description complète du polyèdre sur lequel on désire séparer ou sa représentation dans un espace de dimension supérieure, le problème du renforcement d'inégalité valide est trivial, nous considérerons donc uniquement ici le cas où l'on doit passer par l'utilisation d'un oracle pour obtenir les points extrêmes du polyèdre.

Nous allons tout d'abord formuler le problème du renforcement d'une inégalité de manière très générale, puis nous allons décrire plusieurs méthodes connues pour le résoudre.

### 2.3.1 Formulation du problème du renforcement

Pour plus de simplicité dans l'exposé, nous allons considérer uniquement le cas des polytopes de pleine dimension cependant la plupart des résultats sont généralisables aux polyèdres de dimension

quelconque. Soit  $X$  un polytope de pleine dimension dans  $\mathbb{R}^d$ ,  $x^* \notin X$  un point de  $\mathbb{R}^d$ ,  $\alpha^T x \leq \beta$  séparant  $x^*$  de  $X$ , et  $F = \{x \in X : \alpha^T x = \beta\}$  tel que  $\dim(F) < \dim(X) - 1$ . Le but du problème de renforcement d'une inégalité valide est de modifier  $\alpha$  et  $\beta$  pour ajouter des sommets de  $X$  à  $F$  de manière à augmenter sa dimension. Pour effectuer cela, nous allons procéder par étapes de manière à ce que, à chaque étape, la dimension de l'inégalité soit augmentée de 1. Le problème que l'on doit résoudre est donc :

Étant donné  $X$ ,  $x^*$  et  $\alpha^T x \leq \beta$  définis comme plus haut trouver  $\alpha'^T x \leq \beta'$  telle que  $F = \{x \in X : \alpha^T x = \beta\} \subset F' = \{x \in X : \alpha'^T x = \beta'\}$  et  $\dim(F') \geq \dim(F) + 1$ .

Pour résoudre ce problème, on suppose que l'on a préalablement déterminé un hyperplan d'équation  $a^T x = b$  tel que  $F \subset \{x \in X : a^T x = b\}$ . Une nouvelle inégalité de dimension supérieure peut être trouvée soit par combinaison linéaire positive de  $\alpha^T x \leq \beta$  et  $a^T x \leq b$  soit par combinaison linéaire positive de  $\alpha^T x \leq \beta$  et  $-a^T x \leq -b$  soit par les deux ; la condition d'existence d'une nouvelle inégalité par combinaison linéaire positive étant que  $a^T x \leq b$  (respectivement  $a^T x \geq b$ ) ne soit pas une inégalité valide pour  $X$ .  $a$  et  $b$  étant supposés connus, ces inégalités peuvent être trouvées respectivement en résolvant les problèmes :

$$\begin{cases} \max y \\ \text{t.q. :} \\ (\alpha + ya)^T x \leq \beta + yb \quad \forall x \in X \end{cases} \quad (2.15)$$

et :

$$\begin{cases} \min -y \\ \text{t.q. :} \\ (\alpha - ya)^T x \leq \beta - yb \quad \forall x \in X \end{cases} \quad (2.16)$$

Nous avons laissé de côté pour l'instant le problème de savoir si la nouvelle inégalité obtenue coupait le point  $x^*$  ; remarquons simplement que si  $a^T x^* < b$  (respectivement  $a^T x^* > b$ ), toute combinaison linéaire positive des deux inégalités coupe trivialement le point  $x^*$ .

### 2.3.2 Résolution du problème de renforcement d'une inégalité

Évidemment une approche simple est de résoudre le problème de renforcement par énumération de tous les points extrêmes du polytope. Nous allons donner ici deux approches moins naïves. La première est basée sur une preuve constructive originale du théorème d'existence d'une inégalité de plus grande dimension donnée de manière algorithmique dans [1], que nous avons reprise ici. La seconde est une méthode donnée dans [51] pour la programmation disjonctive que nous adaptons, ici, au cas général.

**Remarque 2.4 :** En fait la première méthode correspond à la résolution par génération de contraintes du problème de renforcement tel que formulé plus haut en l'initialisant avec un seul point  $x^1 = \arg \max\{a^T x : x \in X\}$ . ||

Pour la première méthode, on va donner une preuve constructive du lemme suivant.

**Propriété 2.1** Soient  $X$ ,  $x^*$ ,  $\alpha$ ,  $\beta$  et  $F$  définis comme dans le théorème 2.5. Alors il existe une inégalité  $\alpha'^T x \leq \beta'$  telle que  $F \subset F' = \{x \in X : \alpha'^T x \leq \beta'\}$  et  $\dim(F') = \dim(F) + 1$ .

Démontrons auparavant le lemme suivant ;

**Lemme 2.3** Soit  $X \subset \mathbb{R}^n$  un polytope de dimension  $n - k$ . Soit  $x^* \notin X$ . Soit un polytope  $F \subset X$  tel que  $\dim(F) = n - k - j \leq n - k - 1$  et  $x^* \notin \text{aff}(F)$  (où  $\text{aff}(F)$  est l'enveloppe affine des points de  $F$ ). Alors, il existe  $v \in \mathbb{R}^n$  et  $w \in \mathbb{R}$  tels que :

- (i)  $v^T x = w, \forall x \in F$  ;
- (ii) il existe  $\bar{x} \in X$  avec  $v^T \bar{x} \neq w$  ;
- (iii)  $v^T x^* \neq w$

**Preuve :** D'après les dimensions respectives de  $F$ ,  $X$  et  $\mathbb{R}^n$  :

Il existe  $x^1, \dots, x^j$  appartenant à  $\text{aff}(F)$  affinement indépendants entre eux et avec tout  $x \in \text{aff}(F)$ .

Il existe  $y^1, \dots, y^k$  appartenant à  $\mathbb{R}^n$  affinement indépendants entre eux et avec tout  $x \in X$ .

Prenons  $F' = \text{aff}(F \cup \{x^2, \dots, x^j, y^1, \dots, y^k\})$ . Clairement  $\dim(F') = n - 1$ , c'est donc un hyperplan de  $\mathbb{R}^n$  pouvant être défini par  $v^T x = w$ .

$\dim(F' \cap X) \leq n - k - 1$  donc (ii) est vérifié. Si  $x^* \in \text{aff}(X)$ , on peut prendre  $x^* = x^1$  et (iii) est vérifié. Si  $x^* \notin \text{aff}(X)$ ,  $x^* = \sum_{i=1}^j \lambda^i x^i + \sum_{i=1}^k \mu^i y^i + \bar{x}$  avec  $\bar{x} \in F$ . On prend  $x^1$  tel que  $\lambda^1 > 0$ , on a  $x^* - \lambda^1 x^1 \in F'$  et  $x^1 \notin F'$  donc  $x^* \notin F'$ . ■

**Preuve de la propriété 2.1 :** Par hypothèse  $\dim(F) < \dim(X) - 1$ , il existe donc  $x^0 \in X \setminus F$ . De plus, il existe par application du lemme 2.3,  $v^0$  et  $w^0$  tels que :

- (i)  $v^{0T} x = w^0 \forall x \in F \cup \{x^0\}$ .
- (ii) il existe  $x \in X$  tel que  $v^{0T} x \neq w^0$
- (iii)  $v^{0T} x^* > w^0$

Pour  $i \geq 1$  on définit la suite :

$$x^i = \text{argmax}\{v^{i-1T} x : x \in X, \alpha^T x \neq b\}$$

où  $v^i$  et  $w^i$  sont donnés par :

$$v^i = \mu^i v^{i-1} + \lambda^i \alpha$$

$$w^i = \mu^i w^{i-1} + \lambda^i \beta$$

et  $\lambda^i$  et  $\mu^i$  sont donnés par :

$$\lambda^i = v^{i-1T} x^i - w^{i-1}$$

$$\mu^i = \beta - \alpha^T x^i$$

Dans la suite nous allons montrer que la suite des  $x^i$  devient stationnaire à partir d'un certain moment, et que l'inégalité  $v^{iT} x \leq w^i$  alors obtenue est valide et de la dimension souhaitée.

Par définition de  $x^i$  et comme  $\alpha^T x \leq \beta$  est valide,  $\mu^i > 0$

Pour  $i \geq 0$   $v^{iT} x = w^i \forall x \in F$ . Cela est évidemment vrai pour  $i = 0$ , supposons que cela est vrai pour  $i$  alors :

$$\begin{aligned} v^{i+1T} x &= \mu^{i+1} v^{iT} x + \lambda^{i+1} \alpha^T x \\ &= \mu^{i+1} w^i + \lambda^{i+1} \beta = w^{i+1}. \end{aligned}$$

Pour  $i \geq 0$   $v^{iT} x^i = w^i$  :

$$\begin{aligned}
v^{iT} x^i &= \mu^i v^{i-1T} x^i + \lambda^i \alpha^T x^i \\
&= (\beta - \alpha^T x^i) v^{i-1T} x^i + (v^{i-1T} x^i - w^{i-1}) \alpha^T x^i \\
&= \beta v^{i-1T} x^i - w^{i-1} \alpha^T x^i \\
&= \beta v^{i-1T} x^i - w^{i-1} \alpha^T x^i - \beta w^{i-1} + \beta w^{i-1} \\
&= \beta (v^{i-1T} x^i - w^{i-1}) + w^{i-1} (\beta - \alpha^T x^i) \\
&= \lambda^i \beta + \mu^i w^{i-1}
\end{aligned}$$

Ceci implique que  $\lambda^i \geq 0$  pour tout  $i$ .

$\forall j < i$  on a  $v^{iT} x^j - w^i \leq 0$ . En effet :

$$\begin{aligned}
v^i &= \mu^i v^{i-1} + \lambda^i \alpha \\
&= \mu^i (\mu^{i-1} v^{i-2} + \lambda^{i-1} \alpha) + \lambda^i \alpha \\
&= \mu^i \mu^{i-1} v^{i-2} + (\lambda^i + \mu^i \lambda^{i-1}) \alpha \\
&= \mu^i \mu^{i-1} \dots \mu^{j+1} v^j + (\lambda^i + \mu^i \lambda^{i-1} + \dots + \mu^i \mu^{i-1} \dots \mu^{j+2} \lambda^{j+1}) \alpha
\end{aligned}$$

Soit en posant :

$$\lambda = (\lambda^i + \mu^i \lambda^{i-1} + \dots + \mu^i \mu^{i-1} \dots \mu^{j+2} \lambda^{j+1}) \quad (2.17)$$

et

$$\mu = \mu^i \mu^{i-1} \dots \mu^{j+1} \quad (2.18)$$

(clairement  $\lambda \geq 0$  et  $\mu \geq 0$ )

$$v^i = \mu v^j + \lambda \alpha. \quad (2.19)$$

De même :

$$\begin{aligned}
w^i &= \mu^i \mu^{i-1} \dots \mu^{j+1} w^j + (\lambda^i + \mu^i \lambda^{i-1} + \dots + \mu^i \mu^{i-1} \dots \mu^j + 2\lambda^j + 1) \beta. \\
&= \mu w^j + \lambda \beta
\end{aligned}$$

Donc :

$$\begin{aligned}
v^{iT} x^j - w^i &= \mu (v^{jT} x^j - w^j) + \lambda (a^T x^j - b) \\
&= \lambda (a^T x^j - b) \leq 0
\end{aligned}$$

comme  $\lambda \geq 0$ .

Si  $x^i = x^j$  et  $i > j$  on a  $v^{iT} x^j - w^i = 0$ . Comme  $v^{iT} x^j - w^i = \lambda (a^T x^j - b)$ ,  $a^T x^j < b$  et  $\lambda \geq 0$  on a  $\lambda = 0$  donc  $\lambda^i = 0$  et  $v^{iT} x^i = v^{iT} x^{i-1}$ .

De même pour  $i' \in \{j+1, \dots, i\}$  on a  $\lambda^{i'} = 0$

Il en résulte que tous les  $w^{i'}$  et  $v^{i'}$  sont identiques à un coefficient multiplicateur près et donc  $v^{jT} x \leq w^j$  est une inégalité valide.

Comme  $X$  est un polytope et comme les  $x^i$  sont des points extrêmes de ce polytope, les  $x^i$  différents sont en nombre fini et il existe toujours  $i$  et  $j$  tels que  $x^i = x^j$ . La suite aboutit donc toujours à une inégalité valide en un nombre fini d'itérations. Notons  $\alpha' = v^k$ ,  $\beta' = w^k$  cette inégalité valide obtenue.

$x^i$  n'appartient jamais à  $\text{aff}(F)$  (sinon  $a^T x^j = b$ ) et donc  $\dim(F') = \dim(F) + 1$ .

Il reste à montrer que  $\alpha'^T x^* > \beta'$ . On démontre par récurrence que ceci est vrai pour tout  $i \geq 0$  et tout  $v^i$  et  $w^i$ . Pour  $i = 0$  cela est vrai par hypothèse.

Maintenant, supposons que le caractère coupant est conservé au rang  $i$ , alors :

$$v^{i+1T} x^* = \mu^{i+1} v^{iT} x^* + \lambda^{i+1} \alpha'^T x^* > w^{i+1}$$

■

Nous allons maintenant décrire brièvement la méthode donnée dans [51]. La formulation donnée dans [51] porte sur des polyèdres décrits par programmation disjonctive. Nous allons en donner ici une formulation plus générale en adaptant la démonstration de Perregaard et Balas.

**Théorème 2.6 ([51])** *Le programme suivant :*

$$\left\{ \begin{array}{l} \max \alpha^T x - \beta x_0 \\ t.q. : \\ \left( \begin{array}{c} x \\ x_0 \end{array} \right) \in \text{ext}(X) \\ a^T x - b x_0 = 1 \end{array} \right. \quad (2.20)$$

*est équivalent à (2.15).*

**Preuve :** Commençons par écrire le dual de (2.15). Supposons que les points extrêmes de  $X$  sont numérotés de 1 à  $k$  ( $X = \text{conv}(\{x^1, x^2, \dots, x^k\})$ ). Le dual de (2.15) est donné par :

$$\left\{ \begin{array}{l} \min \sum_{j=1}^k (\beta - \alpha^T x^j) w_j \\ t.q. : \\ \sum_{j=1}^k (a^T x^j - b) w_j = 1 \\ w_j \geq 0 \end{array} \right. \quad (2.21)$$

On pose  $x_0 = \sum_{j=1}^k w_j$  et  $x = \sum_{j=1}^k x^j w_j$ .  $\frac{x}{x_0}$  est une combinaison convexe des  $x^j$  donc  $\frac{x}{x_0} \in \text{conv}(X)$ .

Le dual (2.21) est donc équivalent à :

$$\left\{ \begin{array}{l} \max \alpha^T x - \beta x_0 \\ t.q. : \\ \left( \begin{array}{c} x \\ x_0 \end{array} \right) \in X \\ a^T x - b x_0 = 1 \\ w_j \geq 0 \end{array} \right. \quad (2.22)$$

Il reste à montrer l'équivalence entre (2.22) et (2.20). Notons :

$$A = \text{ext} \left( \left\{ (x, x_0) : \begin{pmatrix} x \\ x_0 \end{pmatrix} \in X, a^T x - bx_0 = 1 \right\} \right)$$

et

$$B = \left\{ (x, x_0) : \begin{pmatrix} x \\ x_0 \end{pmatrix} \in \text{ext}(X), a^T x - bx_0 = 1 \right\}$$

Si  $A = B$ , les deux problèmes (2.22) et (2.20) sont équivalents.

Un point  $(x, x_0)$  est dans  $A$  si et seulement si c'est un rayon extrême de  $\begin{pmatrix} x \\ x_0 \end{pmatrix} \in X$ . En effet, il n'existe pas de  $(y, y_0)$  et  $(z, z_0)$  dans  $A$  tels que  $(x, x_0) = \frac{(y, y_0) + (z, z_0)}{2}$  si et seulement s'il n'existe pas de rayons de  $\begin{pmatrix} x \\ x_0 \end{pmatrix} \in X$  ( $y', y'_0 = \mu(y, y_0)$  et  $(z', z'_0) = \nu(z, z_0)$  tels que  $(x, x_0) = \frac{(y', y'_0) + (z', z'_0)}{2}$ ).

Tout rayon extrême de  $\begin{pmatrix} x \\ x_0 \end{pmatrix} \in X$  correspondant à un point extrême de  $X$ ,  $A$  est bien égal à  $B$ .

■

Dans le cas de la résolution d'un problème 0/1 mixte, le problème (2.20) s'exprime comme :

$$\begin{cases} \max \alpha^T x - \beta x_0 \\ \text{t.q. :} \\ Ax - bx_0 \leq 0 \\ x_i \in \{0, x_0\} \quad i \in I \\ a^T x - bx_0 = 1 \end{cases}$$

(où  $I$  est l'ensemble des variables 0/1 du problème).

## 2.4 Extension d'une inégalité valide

Nous allons nous intéresser ici à un problème plus général que le problème de renforcement qui peut-être résolu d'une manière similaire. La méthode peut être considérée comme une généralisation du lifting.

Ici, contrairement au cas du renforcement, l'inégalité  $\alpha^T x \leq \beta$  n'est pas valide pour le problème mais seulement pour une restriction de celui-ci. Nous allons montrer que quelle que soit la restriction l'inégalité peut être transformée en facette de manière séquentielle.

Nous supposons ici que l'on a une inégalité  $\alpha^T x \leq \beta$  valide pour une restriction du polyèdre de départ  $X \cap Y$  (où  $Y$  est une variété affine), telle que la solution continue que l'on cherche à séparer  $x^*$  appartient à  $Y$ . En particulier, par exemple si  $X$  est l'ensemble des solutions d'un problème de PLNE 0-1,  $Y$  peut être la restriction où l'on a fixé certaines variables à leur borne ; on est alors dans le cas du lifting séquentiel qu'on rappellera au paragraphe 2.6.1. Soit  $a^T x = b$  l'équation d'un hyperplan contenant  $Y$ , autrement dit  $Y \subset \{x : a^T x = b\}$ , soit  $Y'$  tel que  $Y = Y' \cap \{x : a^T x = b\}$  (i.e. la description de  $Y'$  par des égalités est celle de  $Y$  à laquelle on a retiré l'égalité  $a^T x = b$ ) et soit  $X' = Y' \cap X$ . On va considérer le problème d'optimisation suivant :

$$\begin{cases} \max \lambda \\ \text{t.q. :} \\ (\alpha + \lambda a)^T x \leq \beta + \lambda b \quad \forall x \in X' \end{cases} \quad (2.23)$$

Remarquons que ce problème est identique au problème de renforcement sauf que la condition de positivité sur  $\lambda$  a été retirée (en fait comme on le verra  $\lambda$  est garanti positif si l'inégalité  $\alpha^T x \leq \beta$  est valide pour tout  $X'$ ) et que l'on recherche des points sur  $X'$ .

Comme pour le problème de renforcement, on peut définir le problème correspondant à l'autre sens de rotation :

$$\begin{cases} \min \lambda \\ \text{t.q. :} \\ (\alpha + \lambda a)^T x \leq \beta + \lambda b \quad \forall x \in X' \end{cases} \quad (2.24)$$

**Lemme 2.4** *Le problème (2.23) est toujours réalisable. De plus si l'inégalité  $a^T x \leq b$  n'est pas valide pour  $X'$ , le problème est borné.*

**Preuve :** (réalisabilité) Notons  $U = \{x \in X' : \alpha^T x > \beta\}$  l'ensemble des points de  $X'$  ne vérifiant pas l'inégalité. Remarquons que  $U \cap \{x : a^T x = b\} = \emptyset$  (par hypothèse) et que donc  $U \cap \{x : a^T x < b\} = U \cap \{x : a^T x \leq b\}$  et  $U \cap \{x : a^T x > b\} = U \cap \{x : a^T x \geq b\}$ . On va démontrer qu'il existe une solution réalisable par l'absurde. On va démontrer la réalisabilité en énumérant les trois cas :  $U = \emptyset$ ,  $U \cap \{x : a^T x < b\} \neq \emptyset$ ,  $U \cap \{x : a^T x > b\} \neq \emptyset$

1. Si  $U = \emptyset$ ,  $\alpha^T x \leq \beta$  est valide pour  $X'$  le problème est réalisable pour  $\lambda = 0$ .
2. Supposons que  $U \cap \{x : a^T x \leq b\} \neq \emptyset$ . Pour tout  $\lambda \geq 0$  et  $x \in X' \cap \{x : a^T x \leq b, \alpha^T x \leq \beta\}$  on a  $(\alpha + \lambda a)^T x \leq \beta + \lambda b$ . De plus, pour  $\lambda \geq 0$  suffisamment grand tous les points de  $U \cap \{x : a^T x \leq b\}$  vérifient l'inégalité (rappelons que  $U \cap \{x : a^T x \leq b\} = U \cap \{x : a^T x < b\}$ ). On peut donc trouver  $M$  solution optimale de :

$$\begin{cases} \min \lambda \\ \text{t.q.} \\ (\alpha + \lambda a)^T x \leq \beta + \lambda b \quad \forall x \in \{x : a^T x \leq b\} \\ \lambda \geq 0 \end{cases} \quad (2.25)$$

Pour ce  $M$ , on a forcément un  $y \in U \cap \{x : a^T x \leq b\}$  saturant l'inégalité. On va démontrer par l'absurde que l'inégalité  $(\alpha + Ma)^T x \leq \beta + Mb$  est satisfaite par tout  $x \in X'$  et est donc une solution réalisable de (2.23). Supposons donc qu'il existe  $z \in \{x \in X' : a^T x > b\}$  tel que  $(\alpha + Ma)^T z > \beta + Mb$ .

Alors il existe  $\mu \in [0, 1]$  tel que  $x' = \mu y + (1 - \mu)z \in X' \cap \{x : a^T x = b\} = X \cap Y$ . Pour ce  $x'$  on a d'une part :

$$(\alpha + Ma)^T x' = \alpha^T x' + Mb \leq \beta + Mb$$

(comme  $a^T x' = b$  et  $\alpha^T x \leq \beta$  est valide pour  $X \cap Y$ ).

et d'autre part :

$$(\alpha + Ma)^T x' = (\alpha + Ma)^T (\mu y + (1 - \mu)z) = \mu(\beta + Mb) + (1 - \mu)((\alpha + Ma)^T z) > \beta + Mb$$

(comme  $y$  sature l'inégalité et  $z$  ne la vérifie pas). Ce qui mène à une contradiction. L'inégalité obtenue avec  $\lambda = M$  est donc bien solution réalisable de (2.23).

3. Le cas  $U \cap \{x : a^T x \geq b\} \neq \emptyset$  peut-être démontré de la même manière.

(problème borné) Supposons que (2.23) est non borné, alors il existe  $M$  tel que pour tout  $\lambda \geq M$  l'inégalité  $(\alpha + \lambda a)^T x \leq (\beta + \lambda b)$  est valide et donc l'inégalité  $(\frac{1}{\lambda'}\alpha + a)^T x \leq (\frac{1}{\lambda'}\beta + b)$  aussi. En faisant tendre  $\lambda'$  vers  $+\infty$  on obtient que  $a^T x \leq b$  est une inégalité valide. ■

**Corollaire 2.3** Si ni  $a^T x \leq b$  ni  $a^T x \geq b$  ne sont des inégalités valides alors les problèmes (2.23) et (2.24) ont tous deux une solution optimale et sont bornés.

**Remarque 2.5 :** Si  $X \cap Y$  ne contient aucun point extrême de  $X$ , (2.23) et (2.24) ont même solution optimale. ||

On va maintenant s'intéresser à la dimension de l'inégalité générée par la résolution de (2.23). Comme on l'a vu dans la démonstration du lemme 2.4, la nouvelle inégalité va saturer au moins un point  $\bar{x}$  de  $X'$ .

**Lemme 2.5** Soit  $k = \dim\{x \in X \cap Y : \alpha^T x = \beta\}$ . L'inégalité obtenue par la résolution de (2.23) a pour dimension  $k + 1$  dans  $X'$ .

**Preuve :** Par construction  $\bar{x} \in X'$  ne vérifie pas  $a^T x = b$ , il est donc affinement indépendant avec tous les points de  $\{x \in X \cap Y : \alpha^T x = \beta\} \subset \{a^T x = b\}$ . ■

En particulier si  $\alpha^T x \leq \beta$  est une facette de  $X \cap Y$ , en effectuant séquentiellement l'extension à l'aide de (2.23) on obtient une facette de  $X'$ .

On peut déduire de tout cela un résultat un peu plus général que le théorème 2.5 :

**Théorème 2.7** Soit  $X$  polytope et  $Y$  une variété affine de  $\mathbb{R}^d$ , soit  $x^*$  un point de  $\mathbb{R}^d \cap Y$ , et soit  $\alpha^T x \leq \beta$  séparant  $x^*$  de  $X \cap Y$ . Soit  $F = \{x \in X \cap Y : \alpha^T x = \beta\}$ . Il existe une inégalité  $\alpha'^T x \leq \beta'$  définissant une facette de  $X$  telle que  $F \subset F' = \{x \in X : \alpha'^T x = \beta'\}$  coupant  $x^*$ .

**Preuve :** Si l'inégalité de départ n'est pas une facette de  $X \cap Y$ , on peut la renforcer jusqu'à l'obtention d'une facette de  $X \cap Y$  coupant  $x^*$ . Ensuite, par des résolutions successives de (2.23), on aboutit à une facette de  $X$ . Comme l'inégalité n'est pas modifiée sur  $X \cap Y$ , elle coupe toujours  $x^*$ . ■

## 2.5 Transformation d'une inégalité en une facette

Comme on l'a vu dans la partie précédente, il est possible de transformer une inégalité valide pour une restriction en une facette pour le polyèdre complet en appliquant séquentiellement différentes méthodes. Ces méthodes requièrent de résoudre  $n - k$  fois le problème (2.15) ou son dual ; à chaque résolution on augmente d'une unité au moins la dimension de la face induite. Nous allons étudier ici la possibilité d'augmenter systématiquement de plusieurs unités la dimension d'une inégalité valide en une unique résolution d'un problème similaire à (2.15). On s'intéressera en particulier au cas du passage à une facette en une seule étape de résolution.

On va supposer que l'on connaît une inégalité valide  $\alpha^T x \leq \beta$  saturant un unique point de  $X$  noté  $\bar{x}$ ,

et que pour toute inégalité valide  $\beta > 0$ .

On va chercher à exprimer l'ensemble des inégalités valides saturant  $\bar{x}$ .

Complétons  $\alpha$  avec les vecteurs  $a_1, a_2, \dots, a_{n-1}$ , tels que  $a_i^T \bar{x} = b_i$  et  $X \cap \{a_i^T x = b_i\} \cap \{x : \alpha^T x < \beta\} \neq \emptyset$ , en une base de  $\mathbb{R}^n$ .

L'ensemble des inégalités saturant  $\bar{x}$  peut être alors exprimé par :

$$Y = \{(\alpha', \beta') \in \mathbb{R}^{n+1} : \alpha'^T x \leq \beta' \forall x \in X \text{ et } \alpha' = \alpha + \lambda_1 a_1 + \dots + \lambda_{n-1} a_{n-1}, \beta' = \beta + \lambda_1 b_1 + \dots + \lambda_{n-1} b_{n-1}\}.$$

En effet, soit  $a^T x \leq b$  une inégalité valide saturant  $\bar{x}$  (avec  $b > 0$  par hypothèse).  $\alpha, a_1, a_2, \dots, a_{n-1}$  formant une base de  $\mathbb{R}^n$ , il existe des  $\lambda_0, \lambda_1, \dots, \lambda_{n-1}$  tels que  $a = \lambda_0 \alpha + \sum_{i=1}^{n-1} \lambda_i a_i$ .

De plus, comme l'inégalité sature  $\bar{x}$  on a :

$$b = a^T \bar{x} = \lambda_0 \beta + \sum_{i=1}^{n-1} \lambda_i b_i$$

Il nous reste à montrer que  $\lambda_0 > 0$ . Supposons  $\lambda_0 \leq 0$ , soit  $x \in X \cap \{a_i^T x = b_i\} \cap \{x : \alpha^T x < \beta\}$ , on a :

$$a^T x = \lambda_0 \alpha^T x + \sum_{i=1}^{n-1} \lambda_i a_i^T x = \lambda_0 \alpha^T x + \sum_{i=1}^{n-1} \lambda_i b_i > \lambda_0 \beta + \sum_{i=1}^{n-1} \lambda_i b_i.$$

Ce qui montre que dans ce cas l'inégalité n'est pas valide.

En prenant  $\alpha' = \frac{a}{\lambda_0}$  on retrouve bien l'ensemble des inégalités valides donné plus haut.

De plus, remarquons que si  $\alpha'$  est un point extrême de  $Y$ ,  $\alpha'^T x \leq \beta'$  est une facette du polyèdre.

Le problème de trouver une facette saturant  $\bar{x}$  peut donc être reformulé comme :

$$\left\{ \begin{array}{l} \max \sum_{i=1}^{n-1} c_i \lambda_i \\ t.q. : \\ (\alpha + \sum_{i=1}^{n-1} \lambda_i a_i)^T x \leq \beta \quad \forall x \in X \end{array} \right. \quad (2.26)$$

pour un vecteur de coût  $c$  à choisir.

Le problème (2.26) est toujours réalisable d'après le théorème 2.5. De plus, d'après le lemme de Farkas, ce problème est borné si et seulement si il n'existe pas de  $\lambda_i$  tels que  $\sum_{i=1}^{n-1} \lambda_i c_i \geq 0$  et

$$\sum_{i=1}^{n-1} \lambda_i a_i^T x \leq 0 \text{ soit une inégalité valide}$$

**Remarque 2.6 :** Si le problème (2.26) est toujours réalisable, il ne garantit aucunement le fait que la solution optimale soit coupante. ||

Nous allons maintenant chercher à exprimer le dual de (2.26). On suppose que les points extrêmes

de  $X$  sont indexés ( i.e.  $X = \{x^1, x^2, \dots, x^j, \dots, x^k\}$ ).

$$\left\{ \begin{array}{l} \min \sum_{j=1}^k (\beta - \alpha^T x^j) w_j \\ t.q. : \\ \sum_{j=1}^k (a_i^T x^j - b_i) w_j = c_i \quad i = 1, \dots, n-1 \\ w_j \geq 0 \end{array} \right. \quad (2.27)$$

Comme dans le théorème 2.6, en posant  $x_0 = \sum_{j=1}^k w_j$  et  $x = \sum_{j=1}^k x^j w_j$ , on a  $\frac{x}{x_0}$  est une combinaison linéaire convexe positive des  $x^j$  donc  $\frac{x}{x_0} \in \text{conv}(X)$ . Le dual (2.27) est donc équivalent à :

$$\left\{ \begin{array}{l} \max \alpha^T x - \beta x_0 \\ t.q. \\ \frac{x}{x_0} \in \text{conv}(X) \\ a_i^T x - b_i x_0 = c_i \quad i = 1, \dots, n-1 \\ w_j \geq 0 \end{array} \right. \quad (2.28)$$

**Remarque 2.7 :** Dans le cas du renforcement simple, le théorème 2.6 montre que le programme équivalent a 2.27 peut encore être simplifié en cherchant les  $\frac{x}{x_0}$  sur  $X$  au lieu de  $\text{conv}(X)$ . Un tel résultat ne peut être obtenu ici, ce qui limite l'intérêt du passage au dual puisqu'il faut alors connaître la description de l'enveloppe convexe. Néanmoins on peut utiliser cette méthode par exemple dans le cas où l'on a fixé un certain nombre de variables du problème pour chercher une inégalité valide. Dans ce cas on peut obtenir très rapidement une inégalité valide pour la relaxation continue du problème ||

## 2.6 Applications de l'extension d'inégalités valides

L'approche générale présentée au paragraphe 2.4 a déjà été appliquée dans des cas particuliers où son utilisation est simplifiée. Dans ce paragraphe, nous allons donner deux exemples d'applications en essayant dans chaque cas de mettre en évidence les simplifications effectuées.

Nous commençons par une remarque sur les exemples connus d'applications, et la possibilité d'utiliser l'extension d'inégalité valide.

Dans tous les exemples d'applications à notre connaissance, les hyperplans  $a^T x = b$  définissant  $Y$  sont des faces du polyèdre  $X$ . En effet, dans ce cas,  $\text{ext}(X \cap Y) = \text{ext}(X) \cap Y$  (pour une preuve cf. [7]) ce qui simplifie le problème de trouver une inégalité valide séparatrice de  $\text{ext}(X \cap Y)$ .

Par contre dans le cas général en programmation 0/1, où  $X$  est l'enveloppe convexe d'un sous-ensemble de  $\{0, 1\}^n$ , la recherche des points extrêmes de  $X \cap Y$  est aussi difficile que celle des points extrêmes de  $X$ . Et donc la séparation d'une première inégalité sur  $X \cap Y$  est aussi difficile que celle d'une inégalité sur  $X$ .

### 2.6.1 Le cas du lifting

Un cas spécial du problème d'extension d'une inégalité valide pour la programmation 0/1 est celui où  $Y$  est défini par des inégalités  $x_i = 0$  ou 1. La méthode utilisée classiquement dans ce cas pour

effectuer l'extension de l'inégalité est appelée "lifting". Cette méthode a été suggérée entre autres par Padberg [50], Wolsey [58], Zemel [60] et Balas et Zemel[11] et a été largement utilisée dans des implémentations d'algorithmes de plans coupants notamment par Crowder et al.[21], Van Roy et Wolsey [52]. Nous allons montrer ici que les algorithmes de lifting séquentiel sont l'adaptation de la résolution des problèmes (2.23) et (2.24) au cas particulier mentionné ci-dessus.

Soit donc  $I = \{i : x_i^* \in \{0, 1\}\}$  (où  $x^*$  est la solution continue du problème); pour plus de simplicité supposons que  $I = \{1, \dots, k\}$ . On définit  $Y$  comme  $\{x : x_i = x_i^* \ i \in I\}$ . Supposons que l'on a séparé l'inégalité valide  $\alpha^T x \leq \beta$  sur  $X \cap Y$ . On peut donc appliquer la méthode exposée plus haut en éliminant un par un les hyperplans  $x_i = x_i^*$  pour  $i = 1, \dots, k$  de la description de  $Y$ . Nous allons montrer que l'élimination d'un de ces hyperplans  $x_i = x_i^*$  va conduire à l'algorithme de lifting.

Supposons que  $x_i^* = 0$  (le cas  $x_i^* = 1$  se traite de façon analogue). L'inégalité  $x_i \leq 0$  est clairement non valide. On peut donc appliquer le lemme 2.4 et trouver une inégalité valide pour  $X' = X \cap \{x : x_j = x_j^* \ j \in I \setminus \{i\}\}$  de dimension supérieure en résolvant le problème (2.23). Or dans ce cas précis ce problème devient :

$$\begin{cases} \max \lambda \\ t.q. : \\ \alpha^T x \leq \beta \quad \forall x \in X \cap \{x : x_i = 0\} \\ \alpha^T x + \lambda \leq \beta \quad \forall x \in X \cap \{x : x_i = 1\} \end{cases} \quad (2.29)$$

Dans ce problème, la première série de contraintes peut être éliminée (elles correspondent aux points de  $X \cap \{x_i = 0\}$ ).  $\alpha^T x$  et  $\beta$  ne dépendant pas de  $\lambda$ , le  $\lambda$  maximal est déterminé par le problème :

$$\begin{cases} \max \alpha^T x \\ t.q. : \\ x \in X \cap \{x : x_i = 1\} \end{cases} \quad (2.30)$$

Étant donnée  $\bar{x}$  la solution optimale de ce problème, la solution optimale de (2.29) est donnée par  $\bar{\lambda} = \beta - \alpha^T \bar{x}$ .

### 2.6.2 Séparation d'inégalités génériques pour le voyageur de commerce

Comme deuxième exemple d'application nous allons prendre la méthode de séparation pour le problème de voyageur de commerce de Applegate Bixby *et al.*[1] que nous allons rappeler dans le paragraphe suivant.

#### Algorithmes de résolution du voyageur de commerce par génération d'inégalités valides

L'utilisation d'algorithmes de génération d'inégalités valides pour la résolution de problèmes d'optimisation en nombres entiers remonte aux travaux de Dantzig, Fulkerson et Johnson [23] sur le voyageur de commerce.

La méthode, devenue depuis classique, consiste, à partir d'une formulation du problème où seules sont présentes les contraintes sur les degrés des sommets, à résoudre ce problème à l'optimum et, si la solution obtenue n'est pas un cycle hamiltonien, à ajouter une inégalité valide permettant d'éliminer cette solution.

La méthode de base utilise comme inégalités valides les inégalités de Gomory et les inégalités de

sous-tours.

Beaucoup d'autres classes d'inégalités valides ont été développées depuis, par exemple les inégalités de "peigne" [33].

Applegate *et al.*[1] ont développé une nouvelle méthode de séparation d'inégalités valides pour le voyageur de commerce sortant délibérément de l'utilisation de classes d'inégalités valides pré-définies. L'approche utilisée consiste à générer une inégalité valide coupante, la meilleure existante, sur un sous-espace du voyageur de commerce initial ; puis à rendre cette inégalité, par diverses méthodes, valide pour le problème de départ tout en lui conservant son caractère coupant.

La méthode s'est avérée très efficace et ce en particulier sur des problèmes de grande taille jugés difficiles. Nous allons brièvement exposer ici la méthode générale exposée dans [1] en essayant de faire apparaître les idées susceptibles de s'adapter à des problèmes peu structurés (tels que le sac à dos multidimensionnel).

### Réduction de l'espace de recherche

Le but de la réduction de l'espace de recherche est de pouvoir chercher la meilleure inégalité valide dans un espace de dimension suffisamment petite pour que la résolution du problème (SPP') (c.f. paragraphe 2.2.4) soit rapide.

La première méthode pour réduire la dimension de l'espace de recherche consiste à projeter  $x^*$  (la solution continue) et  $X$  (l'espace des solutions réalisables) dans un espace de dimension inférieure à l'aide d'une application linéaire  $\phi$ . On choisit ensuite  $\mathcal{T}$ , une relaxation de  $\phi(X)$  sur laquelle l'optimisation d'une fonction objectif est plus aisée.

Si une inégalité  $\alpha^T \bar{x} \leq \beta$  est trouvée séparant  $\phi(x^*)$  de  $\mathcal{T}$ , il suffit de substituer  $\phi(x)$  à  $\bar{x}$  dans l'inégalité pour trouver une inégalité valide pour  $X$ .

Applegate *et al.*, dans leur approche pour le voyageur de commerce, utilisent pour  $\phi$  une application linéaire regroupant les sommets du graphe de manière à obtenir un graphe  $(\bar{V}, \bar{E})$  d'une trentaine de sommets pour  $\phi(X)$ .  $\mathcal{T}$  est l'ensemble des vecteurs  $x$  entiers indexés par  $\bar{E}$  (les arêtes du nouveau graphe) tels que toutes les composantes de  $x$  sont positives et que le sous-graphe induit par les composantes strictement positives de  $x$  soit connexe et ait tous ses sommets de degré pair. Notons que  $\mathcal{T}$  est bien une relaxation de  $\phi(X)$  car tout sous-graphe connexe de degré paire de  $(\bar{V}, \bar{E})$  n'est pas l'image par  $\phi$  d'un cycle hamiltonien du graphe initial. Cette projection et cette relaxation ont déjà été utilisées [37], [48] dans le cadre du voyageur de commerce pour la séparation de classes d'inégalités valides classiques.

L'utilisation d'une application linéaire pour rendre le problème de séparation plus simple n'est pas une technique limitée dans la littérature au voyageur de commerce. Dans [12] Borndörfer et Weismantel proposent des applications  $\phi$  pour plusieurs problèmes combinatoires tels que le problème du stable maximum, l'arbre de Steiner, le sac à dos multiple ; récemment dans [22] un algorithme de séparation d'inégalités valides pour le problème du stable utilise une application linéaire  $\phi$  construite à partir de concentrations de sommets.

Il faut remarquer ici que  $\phi(x^*)$  peut être à l'intérieur de  $\mathcal{T}$  même si  $x^*$  est à l'extérieur de  $X$ .

Applegate *et al.* utilisent une méthode supplémentaire permettant à la fois de tester rapidement l'appartenance de  $\phi(x^*)$  à  $\mathcal{T}$  et de réduire la dimension de l'espace de recherche de la meilleure inégalité valide.

Soit  $\mathcal{F}$  une famille d'inégalités  $v^T x \leq w$  valides pour  $\mathcal{T}$  saturées par  $\phi(x^*)$ . Soit  $\mathcal{T}'$  l'ensemble des éléments de  $\mathcal{T}$  saturant toutes les inégalités de  $\mathcal{F}$ , on a  $\phi(x^*) \in \mathcal{T}$  si et seulement si  $\phi(x^*) \in \mathcal{T}'$ . Dans le cas d'un problème en variables 0/1, des familles d'inégalités valides élémentaires pour  $\mathcal{F}$  sont les inégalités bornant les variables du problème. En considérant ces inégalités, on peut ainsi supprimer les variables non fractionnaires de  $x^*$  dans la recherche d'inégalité valide.

Dans le cas du voyageur de commerce,  $\mathcal{F}$  est constitué des inégalités :

$$\bar{x}_e \geq 0 \quad \forall e \text{ tel que } x_e^* = 0 \quad (2.31)$$

$$\sum_{e:u \in e} \bar{x}_e \geq 2 \quad \forall u \text{ tel que } \sum_{e:u \in e} x_e^* = 2 \quad (2.32)$$

$$\bar{x}(e, \bar{V} - e) \geq 2 \quad \forall e \text{ tel que } x^*(e, \bar{V} - e) = 2. \quad (2.33)$$

Après toute ces réductions on sépare une facette sur le polaire (tel qu'exposé au paragraphe 2.2) de l'espace de dimension réduite  $\mathcal{T}'$ , on trouve ainsi une inégalité  $a^T x \leq b$  valide pour  $\mathcal{T}'$  séparant  $\phi(x^*)$ .

### Remontée de l'inégalité valide

La phase qu'on appelle ici la remontée consiste à transformer l'inégalité  $a^T x \leq b$  en une inégalité valide coupante pour  $X$ . Ceci s'effectue en deux étapes :

1. Transformer l'inégalité en facette de  $\mathcal{T}$ ,
2. Rendre l'inégalité valide pour  $X$ .

Comme on l'a déjà vu, l'étape 2 est très simple. Nous allons donc nous concentrer sur la transformation de l'inégalité en facette de  $\mathcal{T}$

Dans l'étape 1, il s'agit donc de remplacer une inégalité  $a^T x \leq b$  valide pour la variété affine saturant  $v^T x = w$  pour tout  $(v, w) \in \mathcal{F}$  en une inégalité valide pour les  $x$  vérifiant  $v^T x \leq w$ .

Ce problème est exactement celui de l'extension d'inégalité valide étudié en détail au paragraphe 2.4. L'inégalité est donc étendue séquentiellement à un polyèdre de plus en plus grand en supprimant à chaque étape une des inégalités  $(v, w)$  de  $\mathcal{F}$ .

Suivant le type des  $(v, w)$  la méthode diffère. Pour les  $(v, w)$  "simples", du type  $x_i = 0$  ou  $x_i = 1$ , la méthode utilisée est le lifting exposé au paragraphe 2.6.1.

Pour les  $(v, w)$  plus complexes (correspondant aux inégalités de degrés) l'extension n'est pas effectuée par la résolution directe d'un des problèmes (2.23), (2.24). La méthode utilisée procède en deux étapes pour aboutir à une solution d'un de ces deux problèmes.

La première de ces étapes est de rendre l'inégalité valide, la seconde de renforcer cette inégalité valide jusqu'à obtenir une facette en résolvant le problème de renforcement d'inégalités valides 2.15.

**Rendre l'inégalité valide pour  $\mathcal{T}$**  On remarque que comme  $v^T x \leq w$  ( $\forall v, w \in \mathcal{F}$ ) est valide, pour  $M$  suffisamment grand, l'inégalité :

$$a^T x - M \left( \sum_{v, w \in \mathcal{F}} (v^T x - w) \right) \leq b$$

est valide pour  $\mathcal{T}$  et coupe  $\phi(x^*)$ .

Il suffit de prendre :

$$M \geq \frac{a^T x - b}{\sum_{v, w \in \mathcal{F}} v^T x - w}$$

Dans l'approche utilisée par Applegate *et al.*, compte tenu des  $v$  et  $w$ , il est facile de trouver un tel  $M$  (comme en programmation 0/1 si  $\sum v^T x - w \geq 1$  il suffit de prendre  $M = \|a\|_1 - b$ ). Remarquons que toutes les  $(v, w)$  de même type sont éliminées avant d'effectuer le renforcement.

**Transformer l'inégalité valide en facette de  $\mathcal{T}$**  Le renforcement de l'inégalité est effectué à l'aide d'une méthode d'énumération inspirée par la preuve constructive du théorème 2.5 donnée au paragraphe 2.3, que nous ne détaillons pas ici.

**Remarque 2.8 :** Un aspect, que nous n'avons pas mentionné, de la méthode développée par Applegate *et al.* est le souci de générer des inégalités dont les coefficients sont entiers. ||

Dans cette méthode de séparation, les techniques présentées dans le présent chapitre (de séparation, de renforcement et d'extension d'inégalités valides) sont utilisées pour déterminer une facette coupante de  $\mathcal{T}$ . Contrairement aux techniques classiques de séparation d'inégalités valides, le problème de séparation est pris ici dans sa généralité et résolu par des techniques algorithmiques qui peuvent paraître a priori trop coûteuses en temps. Le succès de la méthode réside évidemment dans les restrictions faites au problème, particulièrement grâce à l'application  $\phi$  et à la restriction  $\mathcal{T}'$  dont la combinaison permet une réduction considérable du problème initial de séparation.

Le caractère général de la méthode laisse espérer que des méthodes semblables pourraient être utilisées pour la séparation d'inégalités sur d'autres polyèdres que celui du voyageur de commerce, si l'on est capable de trouver des réductions appropriées au problème traité.

Dans le prochain chapitre, nous présentons des résultats d'expériences effectuées sur des problèmes de sac à dos multidimensionnels par des méthodes qui ont en commun avec celle-ci de passer par la séparation par des approches polaires.

## Chapitre 3

# Étude comparative de méthodes de séparation par oracles : application au sac à dos multidimensionnel

Dans cette partie, nous présentons des résultats d'expériences comparatives de différentes techniques de recherche d'inégalités valides pour des problèmes de sac à dos multidimensionnels 0/1 de la forme :

$$\left\{ \begin{array}{l} \max c^T x \\ t.q. : \\ Ax \leq b \\ x \in \{0, 1\}^n \end{array} \right. \quad (3.1)$$

(où tous les éléments de  $A$ ,  $b$  et  $c$  sont des entiers positifs). Dans la suite, on notera  $X$  l'ensemble des solutions réalisables :  $\{x \in \{0, 1\}^n : Ax \leq b\}$ .

Les problèmes traités sont les benchmarks de Chu et Beasley [17]. Pour les expériences nous avons retenu les instances Mknpcb7 et 8 qui apparaissent comme étant difficiles à résoudre par CPLEX (version 7). Le tableau 3 donne un récapitulatif des caractéristiques de ces problèmes tests. Pour chacune des contraintes de chaque problème, le second membre  $b_i$  est déterminé grâce au coefficient de "tightness"  $\zeta$  comme étant  $b_i = \zeta(\sum_j a_{ij})$ .

Noms des fichiers		nombre de variables	nombre de contraintes	coefficient de "tightness"
Série	numéro			
7.	00 à 09	100	30	0.25
	10 à 19			0.5
	20 à 29			0.75
8.	00 à 09	250	30	0.25
	10 à 19			0.5
	20 à 29			5

FIG. 3.1 – Caractéristiques des problèmes tests de Chu et Beasley[17]

Toutes les expériences faites dans ce chapitre (et les suivants) sont effectuées sur un Pentium3 400 MHz avec 191 Mo de RAM. Tous les programmes linéaires et les sous-problèmes entiers ou mixtes sont résolus avec CPLEX 7.0.

Nous étudierons tout d'abord la séparation exacte par le critère du ratio (c.f. paragraphe 3.1) qui est celle produisant la meilleure coupe possible et que nous considérerons par la suite comme notre référence pour évaluer comparativement les autres approches. Nous comparerons ensuite d'autres méthodes de séparation basées sur l'utilisation d'heuristiques (c.f. paragraphe 3.2), de relaxations (c.f. paragraphe 3.3) puis sur l'utilisation d'approches par "restriction et lifting" (c.f. paragraphe 3.4). Nous nous attacherons en particulier à comparer les qualités des inégalités obtenues en terme de renforcement de la relaxation continue et de ratio.

### 3.1 Séparations des meilleurs inégalités suivant le critère du ratio

Le tableau 3.2 résume des résultats de calcul pour la séparation de la meilleure inégalité valide par le critère du ratio au travers de la résolution exacte du problème (SPP') :

$$(SPP') \left\{ \begin{array}{l} \max \alpha^T x^* - \beta \\ t.q. : \\ \alpha^T x \leq \beta \quad \forall x \in X \\ (\alpha, \beta) \in \mathcal{N} \end{array} \right.$$

à l'aide de l'algorithme 1 (c.f. paragraphe 2.2.4 page 38), la condition de normalisation  $(\alpha, \beta) \in \mathcal{N}$  est  $\beta = 1$ .

Pour chacun des 10 problèmes considérés, on donne la valeur de la relaxation linéaire avant séparation et le temps de séparation en secondes de temps CPU de l'inégalité coupant la solution continue du problème. On donne ensuite deux valeurs dont le but est de mesurer la qualité de l'inégalité générée. La première est le ratio de l'inégalité trouvée  $\alpha^T x \leq \beta$  donné par  $\mu = \frac{\beta}{\sum_{i=1}^n \alpha^T x^*}$  (rappelons que  $x^*$  est le point à couper et qu'ici la condition de normalisation impose  $\beta = 1$ ). Si le ratio est de 1 l'inégalité n'est pas coupante. Dans le cas d'un problème de sac à dos multidimensionnel le ratio est toujours positif et plus celui-ci est petit plus l'inégalité coupe profondément le point  $x^*$ .

Comme il a été vu au chapitre 2, l'interprétation géométrique du ratio est que le point combinaison convexe de 0 et de  $x^*$  vérifiant  $\alpha^T x \leq \beta$  le plus proche de  $x^*$  est  $0 \times (1 - \mu) + \mu x^*$ . Cette mesure est donnée dans l'avant-dernière colonne du tableau.

Le second critère que nous prenons en compte est le renforcement relatif de la relaxation induit par la coupe ajoutée. Cette mesure est calculée à partir du saut d'intégrité qui est la différence relative entre la valeur de l'optimum relaxé et de l'optimum entier du problème. Le renforcement de la relaxation (que l'on pourrait aussi appeler part du saut d'intégrité comblée) est le pourcentage du saut d'intégrité du problème original comblé par l'ajout de la coupe (de manière plus formelle si  $c^{opt}$  est la valeur de l'optimum entier,  $c^{rel}$  la valeur de la relaxation continue et  $c^{ren}$  la valeur de la relaxation renforcée le renforcement est  $: 1 - \frac{c^{ren} - c^{opt}}{c^{rel} - c^{opt}}$ ). Cette mesure est donnée dans la dernière colonne du tableau.

Le tableau 3.3 présente des résultats de calcul pour une variante de l'algorithme 1. Dans l'algorithme 1, à chaque itération on ajoute une nouvelle contrainte au problème de séparation correspondant

Nom du prob	Relaxation continue	Temps de séparation	Ratio de l'inégalité	Relaxation continue après ajout	Renforcement (en %)
7.00	22579.1	1772.5	0.949180	22544.2	5.51%
7.01	22367.80	4636.9	0.953499	22349.1	2.87%
7.02	21270.50	1269.6	0.953137	21244.5	5.03%
7.03	22049.60	2150.0	0.954305	22041.4	1.40%
7.04	22531.50	1844.3	0.952176	22522.8	1.27%
7.05	22910.10	7840.7	0.955903	22900.0	1.38%
7.06	22530.10	8423.8	0.942735	22513.7	2.24%
7.07	22088.80	4036.7	0.950278	22075.5	1.92%
7.08	23201.20	12321.2	0.952813	23188.5	1.88%
7.09	21524.80	1418.6	0.952304	21492.9	5.89%
Bilan		4571.43	0.951633	22287.3	2.94%

FIG. 3.2 – Meilleure inégalité par génération simple

à l'optimum du sous problème entier à résoudre. La variante utilisée ici consiste à ajouter plusieurs contraintes à chaque itération, une pour chacun des points entiers ne satisfaisant pas l'inégalité courante trouvés lors de la résolution exacte par CPLEX du sous-problème.

Les inégalités trouvées étant identiques à celles du tableau 3.2, on donne juste ici le temps de séparation et le facteur d'accélération par rapport à la résolution par génération simple. Le nombre

Nom du prob	Temps de séparation	Facteur d'accélération
7.00	794,06	2,2
7.01	2736,75	1,7
7.02	1059,25	1,2
7.03	954,2	2,3
7.04	1398,69	1,3
7.05	4373	1,8
7.06	8457,94	1,0
7.07	2572,35	1,6
7.08	5714,21	2,2
7.09	1386,49	1,0
Bilan(moyenne)	2944,69	1,62

FIG. 3.3 – Meilleure inégalité par génération multiple

moyen de points générés à chaque itération pour ces problèmes est de 1,74, le facteur d'accélération moyen est de 1,62.

Afin de mieux apprécier la qualité et l'impact des meilleurs coupes trouvées, nous avons fait des expériences portant sur la résolution complète à l'optimum des problèmes présentés dans le tableau 3.4. Pour chacun des problèmes, on a effectué une résolution complète avec CPLEX, puis une autre

en ajoutant à la description du problème la meilleure coupe. On indique les temps de calcul par chacune des méthodes sans inclure le temps de génération de la coupe, puis la différence relative de temps de calcul de nombre de nœuds de l'arbre de recherche et de nombre total de pivots du simplexe.

Une première constatation est que les temps de calcul de la meilleure coupe sont tout à fait prohibitifs par rapport au temps de résolution total puisque dans tous les problèmes il faut plus de temps pour séparer la meilleure coupe que pour effectuer une résolution complète.

L'impact de la coupe ajoutée varie selon les problèmes. Dans trois des problèmes traités (7.02, 7.03 et 7.04) la coupe a un impact négatif assez fort à la fois sur le temps de résolution, le nombre de nœuds et le nombre de pivots (le temps de résolution augmente en moyenne de 12% sur ces trois problèmes). Dans les sept autres problèmes le temps de résolution diminue en moyenne de 6% avec un seul problème où la diminution est vraiment sensible (près de 25%).

Les mesures de qualité des coupes données dans le tableau 3.2 ne semblent pas permettre d'expliquer les résultats du tableau 3.4. Il est difficile d'avancer une explication satisfaisante comme presque tous les choix dans la résolution exacte sont effectués par CPLEX sans qu'on puisse les contrôler ou même les connaître précisément. On peut penser que les coupes ajoutées agissent parfois négativement sur la résolution parce que les choix dans les branchements deviennent moins bons ou les heuristiques moins efficaces. Par exemple, pour les problèmes 2, 3 et 4, si les choix de branchement effectués étaient les mêmes avec ou sans coupe on devrait avoir un nombre de nœuds explorés au moins inférieur lorsqu'une coupe est ajoutée au problème.

Nom du prob	Optimum	Temps résolution brute	Temps résolution avec coupes	Différence temps	Différence nombre noeuds	Différence itérations simplexe
7.00	21946	635.8	607.3	-4.694%	-8.032%	-4.186%
7.01	21716	3532.1	3480.0	-1.499%	-6.150%	-0.109%
7.02	20754	1459.9	1645.5	11.279%	8.719%	9.726%
7.03	21464	2058.9	2278.2	9.624%	6.012%	7.930%
7.04	21844	8059.9	9645.6	16.440%	13.927%	13.118%
7.05	22176	9169.4	9046.3	-1.362%	-4.962%	-2.598%
7.06	21799	9644.3	9030.2	-6.800%	-9.210%	-6.251%
7.07	21397	6913.2	6504.1	-6.288%	-8.494%	-4.908%
7.08	22525	24774.1	24107.2	-2.766%	1.502%	0.589%
7.09	20983	1229.3	990.8	-24.077%	-18.690%	-25.293%
<b>Bilan</b>		<b>6747.69</b>	<b>6733.5</b>	<b>-1.014%</b>	<b>-2.538%</b>	<b>-1.198%</b>

FIG. 3.4 – Résolution complète par CPLEX après ajout de l'inégalité la plus forte par le critère du ratio

Comme on l'a vu les temps de calcul de la meilleure coupe sont prohibitifs. C'est pourquoi nous étudierons dans les paragraphes suivants diverses méthodes permettant de les réduire, évidemment au prix d'une perte de qualité. Nous verrons en particulier que la méthode proposée au paragraphe 3.4 fondée sur une procédure de lifting originale permet d'obtenir des qualités d'inégalités très proches

de celles des inégalités optimales avec des facteurs d'accélération très significatifs.

### 3.2 Séparation approchée par résolution heuristique du sous-problème

Dans cette partie nous allons donner des résultats d'expériences obtenus en résolvant (SPP').

Dans l'algorithme 1,  $\alpha$  étant fixé au lieu de résoudre de manière exacte le sous-problème  $\max\{\alpha^T x : x \in X\}$ , celui-ci est résolu de manière approchée à l'aide d'une méthode heuristique gloutonne adaptée au sac à dos multidimensionnel décrite dans [14]. Lorsque la solution donnée par l'heuristique ne permet pas d'invalider l'inégalité courante du problème, on conserve le vecteur directeur de l'inégalité  $\alpha$ . Puis on résout le sous-problème  $\max\{\alpha^T x : x \in X\}$  de manière exacte (une seule fois) à l'aide du logiciel CPLEX pour déterminer un second membre valide pour l'inégalité.

Dans le tableau 3.5, on donne le temps de séparation et le ratio des inégalités trouvées. On donne ensuite trois éléments de comparaison avec les meilleures inégalités valides par le critère du ratio.

Le premier est le facteur d'accélération ( $= \frac{\text{Temps de séparation de la meilleure inégalité}}{\text{Temps de séparation de l'inégalité approchée}}$ ).

Le second compare l'efficacité de la meilleure coupe et de la coupe générée du point de vue de la valeur de la relaxation continue après ajout de la coupe,  $\Delta$  étant le gain en valeur objective avec la méthode de séparation heuristique et  $\Delta_{opt}$  le gain avec la meilleure coupe ; dans la deuxième colonne on donne  $\frac{\Delta}{\Delta_{opt}}$ .

Le troisième compare les deux inégalités obtenues du point de vue de leurs ratios. Comme nous l'avons vu plus haut le ratio  $\mu$  d'une inégalité indique que le point  $\frac{x^*}{\mu}$  est le point combinaison convexe de 0 et  $x^*$  vérifiant l'inégalité le plus proche de  $x^*$ . La distance entre ce point et  $x^*$  est donnée par  $(1 - \mu)\|x^*\|$ . Si  $\mu_{opt}$  est le ratio de l'inégalité optimale, on peut ainsi évaluer la qualité de l'inégalité de ratio  $\mu$  en comparant la distance de  $x^*$  à respectivement  $\frac{x^*}{\mu}$  et  $\frac{x^*}{\mu_{opt}}$ . Pour cela il suffit d'évaluer le rapport  $\frac{1-\mu}{1-\mu_{opt}}$  donné dans la dernière colonne du tableau.

On constate que les temps de calculs diminuent sensiblement et sont assez réguliers suivant les problèmes. Cependant la perte de qualité aussi bien en terme de ratio que de renforcement de la relaxation continue est assez grande.

### 3.3 Séparation exacte sur des relaxations mixtes

Soit  $I$  l'ensemble des indices des variables qui sont différentes de 0 et de 1 dans la solution continue du problème. Alors, quel que soit  $I' \subset I$  tel que  $I' \neq \emptyset$ , il existe une inégalité séparatrice sur le polyèdre :

$$\hat{X}^{I'} = \{x : Ax \leq b, x_i \in \{0, 1\} \text{ } i \in I', x_i \in [0, 1] \text{ } i \notin I'\}.$$

On peut donc chercher une inégalité séparatrice sur  $\hat{X}^{I'}$  à l'aide de l'algorithme 1. Dans ce cas le sous-problème est un problème mixte pouvant être résolu de manière exacte par CPLEX.

Nom du prob	Temps de séparation	Ratio de l'inégalité	Facteur d'accélération	Part de la relaxation gagnée	Fraction du ratio maximal
7.00	297,54	0,97236	6,0	67,05%	54,39%
7.01	195,47	0,98505	23,7	29,41%	32,16%
7.02	192,74	0,98199	6,6	17,69%	38,44%
7.03	226,83	0,97545	9,5	41,46%	53,72%
7.04	195,48	0,97567	9,4	39,08%	50,87%
7.05	191,4	0,98108	41,0	33,66%	42,90%
7.06	199,27	0,97984	42,3	23,17%	35,20%
7.07	206	0,96794	19,6	66,17%	64,49%
7.08	239,76	0,97405	51,4	20,47%	55,00%
7.09	233,21	0,97351	6,1	39,50%	55,54%
<b>Bilan (moyenne)</b>	<b>217,77</b>	<b>0,98</b>	<b>21,55</b>	<b>37,77%</b>	<b>48,27%</b>

FIG. 3.5 – Inégalité par génération à l'aide d'une heuristique

Dans les tableaux 3.6, 3.7 et 3.8 nous présentons des résultats d'expériences réalisées pour différents choix de  $I'$ . Pour chaque série d'expériences, on donne le nombre de variables contenues dans  $I'$  (i.e. le nombre de variables dont l'intégrité est prise en compte dans le sous-problème), le temps de séparation, le ratio des inégalités trouvées, le facteur d'accélération par rapport au temps pour trouver la meilleure inégalité sur  $X$ , la part du saut d'intégrité fermée relativement au gain obtenu en ajoutant la meilleure inégalité, la variation relative du ratio par rapport à la meilleure inégalité (voir explications du tableau 3.5).

On constate que les temps de calcul diminuent substantiellement en même temps que le nombre de variables entières (c.a.d  $|I'|$ ) dans le sous-problème. La qualité des inégalités générées décroît également légèrement lorsque l'effort de calcul diminue mais est dans l'ensemble nettement meilleure qu'avec la méthode utilisant une heuristique présentée au chapitre précédent.

### 3.4 Résolution du problème de séparation sur une restriction du problème et lifting

Soit  $x^*$  la solution optimale de la relaxation continue. Sans perte de généralité nous allons supposer que les composantes de  $x^*$  sont ordonnées de manière à ce que les  $p$  premières composantes sont fractionnaires, les  $p'$  suivantes égales à 1 et les  $n - p - p'$  dernières égales à 0. Soient :

$$\begin{aligned}
 I &= \{i : x_i^* \in ]0, 1[ \} = \{1, \dots, p\} \\
 U &= \{i : x_i^* = 1\} = \{p + 1, \dots, p'\} \\
 L &= \{i : x_i^* = 0\} = \{p' + 1, \dots, n\} \\
 \tilde{X} &= \{x : Ax \leq b, x_i \in \{0, 1\} \ i \in I, x_i = x_i^* \ i \notin I\}. \\
 \hat{X} &= \{x : Ax \leq b, x_i \in \{0, 1\} \ i \in I, x_i \in [0, 1] \ i \notin I\}.
 \end{aligned}$$

Dans ce paragraphe nous étudions une approche consistant à effectuer la séparation en résolvant le problème (SPP') sur l'ensemble de points réalisables défini par  $\tilde{X}$  à l'aide de l'algorithme 1 du

Nom du prob	I	Temps de séparation	Ratio de l'inégalité	Facteur d'accélération	Part de la relaxation gagnée	Fraction du ratio maximal
7.00	23	127.75	0.96782	13.9	67.62%	63.32%
7.01	20	143.65	0.96926	32.3	79.68%	66.12%
7.02	21	105.82	0.97030	12.0	73.46%	63.37%
7.03	21	132.09	0.96796	16.3	70.73%	70.12%
7.04	22	128.08	0.96707	14.4	65.52%	68.87%
7.05	23	177.46	0.96746	44.2	82.18%	73.79%
7.06	24	207.13	0.96712	40.7	45.12%	57.42%
7.07	23	249.2	0.96463	16.2	59.40%	71.14%
7.08	27	812.74	0.96461	15.2	55.91%	75.01%
7.09	20	108.5	0.97022	13.1	54.55%	62.44%
<b>Bilan(moyenne)</b>	<b>22.4</b>	<b>219.242</b>	<b>0.97</b>	<b>21.81</b>	<b>65.42%</b>	<b>67.16%</b>

FIG. 3.6 – Résultats de séparation sur un polyèdre mixte  $\hat{X}^{I'}$  avec  $I' = I = \{i : x_i^* \notin \{0, 1\}\}$ 

Nom du prob	I'	Temps de séparation	Ratio de l'inégalité	Facteur d'accélération	Part de la relaxation gagnée	Fraction du ratio maximal
7.00	18	66.38	0.96964	26.7	44.70%	59.75%
7.01	18	95.88	0.96982	48.4	62.03%	64.91%
7.02	20	97.84	0.97107	13.0	65.77%	61.73%
7.03	20	121.78	0.96861	17.7	70.73%	68.69%
7.04	20	109.8	0.96874	16.8	54.02%	65.37%
7.05	22	145.54	0.96865	53.9	80.20%	71.10%
7.06	21	141.71	0.96855	59.4	44.51%	54.91%
7.07	21	211.53	0.96484	19.1	54.89%	70.71%
7.08	27	814.65	0.96461	15.1	55.91%	75.01%
7.09	19	108.5	0.97022	13.1	54.55%	62.44%
<b>Bilan(moyenne)</b>	<b>20.6</b>	<b>191.361</b>	<b>0.97</b>	<b>28.31</b>	<b>58.73%</b>	<b>65.46%</b>

FIG. 3.7 – Résultats de séparation sur un polyèdre mixte  $\hat{X}^{I'}$  avec  $I' = \{i : x_i^* \in [0.05, 0.95]\}$

Nom du prob	$ I' $	Temps de séparation	Ratio de l'inégalité	Facteur d'accélération	Part de la relaxation gagnée	Fraction du ratio maximal
7.00	13	60.7	0.97255	29.2	37.54%	54.01%
7.01	17	85.55	0.97075	54.2	58.29%	62.89%
7.02	18	83.07	0.97307	15.3	66.92%	57.46%
7.03	20	132.05	0.96861	16.3	70.73%	68.69%
7.04	19	114.73	0.96967	16.1	57.47%	63.43%
7.05	20	132.49	0.97027	59.2	61.39%	67.42%
7.06	19	115.6	0.96995	72.9	36.59%	52.48%
7.07	20	156.93	0.96551	25.7	51.13%	69.37%
7.08	24	500.16	0.96681	24.6	45.67%	70.33%
7.09	18	100.95	0.97129	14.1	32.60%	60.20%
<b>Bilan(moyenne)</b>	<b>18.8</b>	<b>148.223</b>	<b>0.97</b>	<b>32.75</b>	<b>51.83%</b>	<b>62.63%</b>

FIG. 3.8 – Résultats de séparation sur un polyèdre mixte  $\hat{X}^{I'}$  avec  $I' = \{i : x_i^* \in [0.1, 0.9]\}$

paragraphe 2.2.4, puis à transformer l'inégalité trouvée en une inégalité valide de  $X$  par une procédure nouvelle de lifting que nous appellerons lifting relaxé. L'originalité de l'approche est que le lifting est effectué séquentiellement sur un polyèdre qui dépend de l'ordre dans lequel est effectué le lifting. La procédure est exposée en détail dans le paragraphe qui suit.

### 3.4.1 Présentation de la procédure de lifting relaxé

On suppose qu'on a une inégalité  $\alpha^T x \leq \beta$  valide pour  $\tilde{X}$  coupant  $x^*$  et ayant pour support  $I$ .

On va décrire l'opération permettant de lifter  $\alpha^T x \leq \beta$  en une inégalité valide pour  $X$  coupant  $x^*$ .

Pour cela nous allons commencer par décrire le polyèdre (contenant  $X$  et contenu dans  $\hat{X}$ ) sur lequel l'inégalité doit être liftee.

Pour tout  $j > p$  définissons :

$$\begin{aligned} X^j &= \left\{ x : Ax \leq b \ x_i \in \{0, 1\} \ i \in I, \ x_i \in \{0, 1\} \ i = p+1, \dots, j-1, \ x_j = 1 - x_j^*, \ x_i = x_i^* \ i > j \right\}, \\ Q^j &= \left\{ x : Ax \leq b \ x_i \in \{0, 1\} \ i \in I, \ x_i \in [0, 1] \ i = p+1, \dots, j-1, \ x_j = 1 - x_j^*, \ x_i = x_i^* \ i > j \right\}, \\ \hat{X}^j &= \left\{ x : Ax \leq b \ x_i \in \{0, 1\} \ i \in I, \ x_i \in [0, 1] \ i = p+1, \dots, j-1, \ x_j \in [0, 1], \ x_i = x_i^* \ i > j \right\}, \\ X^p &= Q^p = \hat{X}^p = \tilde{X}. \end{aligned}$$

#### Proposition 3.1

$$X \subseteq \left( \bigcup_{j=p}^n Q^j \right) \subseteq \hat{X} \quad (3.2)$$

**Preuve :**

$$\begin{aligned} X^j &\subseteq Q^j \subseteq \hat{X}^j && \forall j = p, \dots, n \\ \bigcup_{i=p}^n X^i &\subseteq \bigcup_{i=p}^n Q^i \subseteq \bigcup_{i=p}^n \hat{X}^i \end{aligned}$$

Or on a :  $\left(\bigcup_{i=p}^n X^j\right) = X^n = X$  et  $\hat{X}^n = \tilde{X}$ .

■

On note  $\tilde{X}^j$  le polyèdre conv  $\left(\bigcup_{i=p}^j Q^i\right)$  sur lequel le lifting va être effectué.

**Proposition 3.2** *Pour  $j = p, \dots, p' - 1$  (tel que  $x_{j+1}^* = 1$ ), soit  $\alpha^T x \leq \beta$  une inégalité valide pour  $\tilde{X}^j$  ayant pour support  $\{i : 1 \leq i \leq j\}$ . Alors si  $\alpha' \in \mathbb{R}^n$  et  $\beta' \in \mathbb{R}$  sont tels que  $\alpha'_i = \alpha_i$  si  $i \neq j+1$ ,  $\alpha'_{j+1} = \max_{x \in Q^{j+1}} \alpha^T x - \beta$  et  $\beta' = \beta + \alpha'_{j+1}$ . L'inégalité  $\alpha' x \leq \beta'$  est valide pour  $\tilde{X}^{j+1}$ .*

*De plus si  $\alpha^T x \leq \beta$  définit une face de dimension  $q$  de  $\tilde{X}^j$ ;  $\alpha'^T x \leq \beta'$  définit une face de dimension  $q + 1$  de  $\tilde{X}^{j+1}$*

**Preuve :**  $\tilde{X}^{j+1} = \text{conv} \left( \left( \bigcup_{i=p}^j Q^i \right) \cup (Q^{j+1}) \right)$ .

Pour montrer que une inégalité est valide pour  $\left(\tilde{X}^{j+1}\right)$ , il suffit de montrer qu'elle est valide pour  $\text{conv} \left( \bigcup_{i=p}^j Q^i \right)$  et pour  $\text{conv} (Q^{j+1})$ .

Si  $x \in \text{conv} \left( \left( \bigcup_{i=p}^j Q^i \right) \right)$  alors  $x_{j+1} = 1$  donc  $\alpha'^T x = \alpha'_{j+1} + \alpha^T x \leq \alpha'_{j+1} + \beta$ . De plus si  $\alpha^T x = \beta$ ,  $\alpha'^T x = \beta$ .

Si  $x \in \text{conv} (Q^{j+1})$  alors  $x_{j+1} = 0$  donc  $\alpha'^T x = \alpha^T x \leq \beta + \max_{x \in Q^{j+1}} \alpha^T x - \beta = \beta'$ .

De plus si  $x = \arg \max_{x \in Q^{j+1}} \alpha^T x$ , l'inégalité est satisfaite à l'égalité et comme  $x_{j+1} = 0$ ,  $x$  est affinement indépendant des points de  $\tilde{X}^j$  et en particulier de ceux saturant l'inégalité. On a donc  $q + 2$  points affinement indépendants saturant la nouvelle inégalité. ■

**Remarque 3.1 :** Notons que bien que  $\tilde{X}^n$  soit contenu dans  $\hat{X}$  l'opération de lifting séquentielle est plus simple sur  $\tilde{X}^n$ . En effet, si l'on utilise l'algorithme d'extension d'inégalité valide présenté au paragraphe 2.4 pour effectuer le lifting sur le polyèdre mixte  $\hat{X}$ , on obtient un problème d'optimisation paramétré :

$$(LIFT(\lambda)) \begin{cases} \max \alpha^T x \\ \text{t.q. :} \\ x \in \hat{X} \cap \{x : x_i = \lambda\} \\ \lambda \in \mathbb{R} \end{cases}$$

A contrario, le lifting sur  $X$  va demander de résoudre des problèmes ayant de plus en plus de variables entières et est donc aussi plus complexe.

||

Le lifting des variables fixées à 0 s'effectue de manière analogue comme le montre la proposition suivante :

**Proposition 3.3** Pour  $j = p' - 1, \dots, n - 1$  (tel que  $x_{j+1}^* = 0$ ), si  $\alpha^T x \leq \beta$  est valide pour  $\tilde{X}^j$  et a pour support  $\{i : 1 \leq i \leq p'\}$ ,  $\alpha^T x \leq \beta$  est valide pour  $\tilde{X}^{j+1}$ , alors si  $\alpha' \in \mathbb{R}^n$  et  $\beta' \in \mathbb{R}$  sont tels que  $\alpha'_i = \alpha_i$  si  $i \neq j + 1$ ,  $\alpha'_{j+1} = \beta - \max_{x \in Q^{j+1}} \alpha^T x$  et  $\beta' = \beta$ . L'inégalité  $\alpha' x \leq \beta'$  est valide pour  $\tilde{X}^{j+1}$ .

La preuve de cette proposition est similaire à celle de la proposition précédente.

Dans le cas du sac à dos multidimensionnel le lifting des variables fixées à 0 n'est cependant pas nécessaire comme le montre la proposition suivante.

**Proposition 3.4** Si  $X$  est tel que  $\forall x \in X, \{x' \in \mathbb{R}^n : x' \leq x\} \subseteq X$ . Alors, si  $\alpha^T x \leq \beta$  est valide pour  $\tilde{X}^{p'}$  et a pour support  $\{i : 1 \leq i \leq p'\}$ ,  $\alpha^T x \leq \beta$  est valide pour  $X$ .

**Preuve :** (par l'absurde) Soit  $x \in X$  et  $x'$  sa projection sur  $\{x : x_i = 0, i = p' + 1, \dots, n\}$ . Par définition  $x' \in \tilde{X}^{p'}$  et  $\alpha^T x' = \alpha^T x$ . Donc si  $\alpha^T x > \beta$ ,  $\alpha^T x' > \beta$  aussi ce qui contredit la validité de l'inégalité sur  $\tilde{X}^{p'}$ . ■

---

**Algorithme 2** Procédure de lifting relaxé pour le sac à dos multidimensionnel

---

Soit  $\alpha \in \mathbb{R}_+^n$  tel que  $\alpha_i = 0 \forall i \notin I$  et  $\beta \in R_+$  tels que  $\alpha^T x \leq \beta$  est valide pour  $\tilde{X}$ .

2: **pour**  $j = p + 1, \dots, p'$  **faire**

$$\alpha_j := \max_{x \in \tilde{X}^j} \alpha^T x - \beta$$

4:  $\beta := \beta + \alpha_j$ .

**fin pour**

---

Intéressons nous maintenant au caractère coupant de l'inégalité liftée obtenue. Soient  $(\alpha^1, \beta^1), \dots, (\alpha^{p-p'}, \beta^{p-p'})$  les inégalité obtenues en liftant l'inégalité  $\alpha^T x \leq \beta$  successivement sur les variables  $p + 1, \dots, p'$ .

Si  $\frac{\beta}{\alpha^T x^*} < 1$  alors  $\frac{\beta^1}{\alpha^{1^T} x^*} < 1$ .

En effet, on a  $\frac{\beta}{\alpha^T x^*} < 1 = \frac{\alpha^1_{p+1}}{\alpha^1_{p+1}}$  donc  $\frac{\beta^1}{\alpha^{1^T} x^*} = \frac{\beta + \alpha^1_{p+1}}{\alpha^T x^* + \alpha^1_{p+1}} < \frac{\alpha^1_{p+1}}{\alpha^1_{p+1}} = 1$ .

De la même manière on peut montrer que si  $\frac{\beta^j}{\alpha^{j^T} x^*} < 1$  alors  $\frac{\beta^{j+1}}{\alpha^{j+1^T} x^*} < 1$ .

Ceci montre que si l'inégalité de départ  $\alpha^T x < \beta$  est coupante alors en effectuant le lifting séquentiellement on obtient une inégalité qui reste coupante.

**Remarque 3.2 :** L'inégalité obtenue en appliquant l'algorithme de lifting dépend de l'ordre des variables sur lesquelles le lifting est effectué séquentiellement. Nous avons réalisé des expériences sur les problèmes de sac à dos multidimensionnels pour tester l'influence de l'ordre des variables, mais celles-ci n'ont pas permis de dégager de grandes différences suivant l'ordre dans lequel les variables sont prises (nous n'avons cependant pas pu tester tous les ordres étant donné leur nombre). ||

### 3.4.2 Résultats expérimentaux

Les tableaux suivants donnent les résultats de séparation obtenus avec la méthode de lifting relaxé ; les ratios des inégalités obtenues sont comparés avec ceux de la séparation exacte donnés au paragraphe 3.1.

### 3.4. RÉSOLUTION DU PROBLÈME DE SÉPARATION SUR UNE RESTRICTION DU PROBLÈME ET LIFTING63

Le tableau 3.9 donne les résultats pour la séparation exacte sur la restriction du problème  $\tilde{X}$  puis lifting. Les données indiquées sont identiques à celles du tableau 3.6.

Les résultats sont bien meilleurs qu'avec les méthodes précédentes pour approcher la meilleure inégalité. On obtient des facteurs d'accélération nettement supérieurs tout en approchant mieux le ratio optimum. Notons que la majeure partie du temps de calcul est consacrée à la recherche de la meilleure inégalité sur la restriction du problème et qu'une faible part seulement est consacrée à la procédure de lifting.

Nom du prob	I	Temps de séparation	Ratio de l'inégalité	Facteur d'accélération	Part de la relaxation gagnée	Fraction du ratio maximal
7.00	23	30.45	0.95992	58.2	86.53%	78.86%
7.01	20	19	0.96261	244.0	85.56%	80.41%
7.02	21	41.65	0.95975	30.5	90.00%	85.90%
7.03	21	24.51	0.96046	87.7	87.80%	86.52%
7.04	22	37.82	0.95881	48.8	104.60%	86.13%
7.05	23	74.7	0.96555	105.0	91.09%	78.13%
7.06	24	70.98	0.95841	118.7	70.73%	72.62%
7.07	23	97.28	0.95682	41.5	90.23%	86.85%
7.08	27	298.49	0.95720	41.3	78.74%	90.69%
7.09	20	20.89	0.96507	67.9	75.86%	73.23%
<b>Bilan (moyenne)</b>	<b>22.4</b>	<b>71.577</b>	<b>0.96</b>	<b>84.35</b>	<b>86.11%</b>	<b>81.94%</b>

FIG. 3.9 – Résultats de la séparation sur  $\tilde{X}$  puis lifting relaxé sur les problèmes tests de Chu et Beasley [17]

Au vu de ces résultats plutôt bons, nous avons effectué, comme au paragraphe 3.1, des expériences portant sur la résolution complète des problèmes relatées dans les tableaux 3.10 et 3.11.

Dans la première série d'expériences nous avons (comme au paragraphe 3.1) ajouté la première coupe trouvée puis comparé les temps de résolutions, le nombre de nœuds de l'exploration et le nombre de pivot du simplexe. La seule différence est qu'ici, pour la comparaison des temps, nous avons inclus le temps de génération de la coupe.

Les résultats obtenus sont en moyenne meilleurs que ceux du tableau 3.4, même en prenant en compte le temps de génération de l'inégalité. Le temps de recherche de la coupe représente en moyenne 1,5% du temps total de résolution et est compris selon les problèmes entre 0.2% et 5% du temps de résolution. Par contre la comparaison entre les deux tableaux n'est pas uniforme suivant les problèmes : pour les problèmes 7.02, 7.05 et 7.07 les résultats sont moins bons avec l'inégalité liftée même si on ne prend pas en compte le temps de génération comme dans le tableau 3.4.

Ici encore, les ratios et les renforcements peinent à expliquer les différences. Cependant l'analyse des coefficients directeurs des inégalités trouvées nous permet d'avancer une tentative d'explication. Tout d'abord, nous avons utilisé une procédure simple qui transforme les inégalités de manière à ce que les coefficients directeurs soient entiers (à  $\epsilon = 10^{-4}$  près) et les plus petit possibles (en conservant

Nom du prob	Optimum	Temps résolution brute	Temps résolution avec coupes	Temps résolution avec coupes + Temps génération	Différence temps	Différence nombre noeuds	Différence itérations simplexe
7.00	21946	635.8	536.0	564.5	-12.630%	-19.155%	-12.952%
7.01	21716	3532.1	3064.5	3080.8	-14.650%	-13.718%	-9.386%
7.02	20754	1459.9	2098.7	2137.4	31.698%	23.274%	26.022%
7.03	21464	2058.9	2020.1	2049.5	-0.458%	0.840%	-0.442%
7.04	21844	8059.9	8299.9	8325.4	3.190%	1.391%	1.797%
7.05	22176	9169.4	10771.6	10829.3	15.328%	14.726%	17.167%
7.06	21799	9644.3	7752.2	7808.1	-23.516%	-25.541%	-18.497%
7.07	21397	6913.2	8161.1	8257.8	16.283%	18.455%	16.363%
7.08	22525	24774.1	23932.3	24208.0	-2.339%	1.750%	0.587%
7.09	20983	1229.3	976.2	992.2	-23.897%	-26.135%	-31.618%
Bilan		6747.69	6761.3	6825.3	-1.099%	-2.411%	-1.096%

FIG. 3.10 – Résolution complète par CPLEX après ajout d'une inégalité liftée

la validité des inégalités c'est à dire en effectuant uniquement des multiplications par des nombres positifs). Dans six des dix problèmes les inégalités ainsi transformées ne comportent que des coefficients 0, 1 et 2 et à l'exception du problème 7.05 pour tous ces problèmes l'ajout de l'inégalité accélère la résolution par CPLEX. Dans un seul problème (le problème 7.08), on a des coefficients variant entre 0 et 3 et là aussi l'inégalité permet d'accélérer. Dans les trois problèmes restants (7.03, 7.04 et 7.07), le plus grand coefficient de l'inégalité est au minimum de 14 pour 7.03 et supérieur à mille pour les deux autres (sur lesquels on a une augmentation de respectivement 3% et 16% du temps de calcul). Les expériences sont insuffisantes pour pouvoir valider sérieusement l'hypothèse, mais on peut cependant penser que les inégalités ayant des coefficients faibles et peu variés sont mieux prises en compte dans les stratégies de branchement par CPLEX.

Dans la seconde série d'expériences, nous avons ajouté cinq coupes trouvées en itérant la séparation. Dans ce tableau (3.11), nous n'avons pas cumulé les temps de séparation et les temps de résolution car les temps de séparation sont trop élevés (le temps de séparation augmentent assez nettement au fur et à mesure que des inégalités sont ajoutées au problème). Dans le tableau 3.11, on indique les temps de séparation en secondes, la diminution du saut d'intégrité obtenue, les temps de résolution avec la formulation initiale, puis avec la formulation renforcée par les inégalités trouvées puis les différences relatives des temps de résolution, de nombre de nœuds explorés et de nombre de pivots.

Comme on l'a déjà dit, les temps de résolutions sont assez importants. Le saut d'intégrité a tout de même diminué en moyenne de 5%. La comparaison des résolutions complètes est décevante puisqu'à l'exception de deux problèmes, le temps de résolution est nettement plus important par rapport à la résolution par CPLEX avec ou sans l'ajout d'une seule inégalité au problème. En étudiant les coefficients directeurs des inégalités ajoutées aux problèmes, on peut remarquer que dans le cas du problème 7.01 (le seul où la résolution par CPLEX est nettement plus rapide) quatre des inégalités ajoutées ont leur plus grand coefficient inférieur ou égal à 2 et la dernière a des coefficients inférieurs à 10. Seul le problème 7.06 est aussi dans ce cas ; pour tous les autres problèmes on a au moins une inégalité dont le plus grand coefficient est supérieur à 100.

Une autre explication pourrait venir de l'étude des angles formés entre elles par les inégalités trouvées. Les inégalités trouvées successivement sont en effet proches d'être parallèles (le cosinus entre deux inégalités est en moyenne supérieur à 0.9 et n'est jamais inférieur à 0.8) et aussi de plus en parallèles

### 3.4. RÉOLUTION DU PROBLÈME DE SÉPARATION SUR UNE RESTRICTION DU PROBLÈME ET LIFTING65

Nom du prob	Temps de séparation	Renforcement	Temps résolution brute	Temps résolution avec coupes	Différence temps	Différence nombre noeuds	Différence itérations simplexe
7.00	1376.44	8.78%	635.8	923.7	31.163%	19.058%	28.384%
7.01	533.570	9.01%	3532.1	2544.7	-38.802%	-37.290%	-26.122%
7.02	311.390	6.18%	1459.9	2143.5	31.892%	26.720%	29.850%
7.03	304.950	3.53%	2058.9	3151.4	34.666%	30.139%	32.499%
7.04	1197.630	4.93%	8059.9	8451.8	4.637%	3.869%	3.449%
7.05	726.670	6.03%	9169.4	12270.7	25.274%	21.141%	25.365%
7.06	493.840	4.71%	9644.3	11095.4	13.079%	13.730%	11.156%
7.07	1324.310	4.51%	6913.2	12314.0	43.859%	35.870%	48.726%
7.08	4434.990	3.58%	24774.1	23386.0	-5.936%	-11.090%	-9.036%
7.09	112.110	7.35%	1229.3	1466.2	16.154%	13.972%	15.286%
<b>Bilan</b>		<b>5.86%</b>	<b>6747.69</b>	<b>7774.7</b>	<b>15.599%</b>	<b>11.612%</b>	<b>15.956%</b>

FIG. 3.11 – Résolution complète par CPLEX après ajout de cinq inégalités liftées

à fonction objective.

Nous allons maintenant donner divers résultats où l'on combine le lifting et une résolution approchée du problème de séparation pour essayer d'accélérer encore les temps de séparation.

Le tableau 3.12 donne les résultats pour la séparation sur la restriction du problème  $\tilde{X}$  en utilisant une heuristique de la même manière que dans le paragraphe 3.2 puis lifting. Les résultats montrent là aussi une accélération très grande par rapport à la méthode exacte ou oar rapport à la méthode approchée. Par contre la qualité des inégalités en ratio comme en renforcement est assez nettement moins bonne.

Les deux tableaux 3.13 et 3.14 donnent les résultats pour la séparation sur une relaxation mixte du

Nom du prob	Temps de séparation	Ratio de l'inégalité	Facteur d'accélération	Part de la relaxation gagnée	Fraction du ratio maximal
7.00	9.1	0.98361	194.8	27.79%	32.25%
7.01	8.51	0.98026	544.9	37.97%	42.46%
7.02	6.72	0.98198	188.9	16.54%	38.46%
7.03	5.44	0.99735	395.2	1.22%	5.79%
7.04	10.25	0.97957	179.9	26.44%	42.71%
7.05	11.03	0.97991	710.9	35.64%	45.55%
7.06	10.52	0.97801	800.7	16.46%	38.41%
7.07	8.93	0.97330	452.0	15.04%	53.69%
7.08	23.61	0.98299	521.9	13.39%	36.05%
7.09	7.48	0.97948	189.7	25.08%	43.02%
<b>Bilan(moyenne)</b>	<b>10.159</b>	<b>0.98</b>	<b>417.89</b>	<b>21.56%</b>	<b>37.84%</b>

FIG. 3.12 – Résultats de la séparation à l'aide d'un heuristique sur  $\tilde{X}$  puis lifting relaxé sur les problèmes tests de Chu et Beasley [17]

problème  $\tilde{X}$  établie de la même manière que dans le paragraphe 3.3 puis lifting. Notons, ici aussi, que, comparativement à ceux obtenus au paragraphe 3.3, les résultats sont meilleurs en terme d'accélération et de renforcement. Notons aussi que les résultats du tableau 3.13 sont en moyenne moins bons (en temps et en ratio) que ceux obtenus avec la méthode de séparation exacte puis lifting ; ceci peut s'expliquer par le fait que dans le cadre de la séparation sur un polyèdre mixte on a en général plus de points extrêmes à générer et que la méthode demande plus d'itérations même si chaque itération est plus rapide. Il semble donc que l'utilisation d'un polyèdre mixte n'est pas intéressante si l'intégrité de peu de variables sont relaxées.

Nom du prob	I	Temps de séparation	Ratio de l'inégalité	Facteur d'accélération	Part de la relaxation gagnée	Fraction du ratio maximal
7.00	18	20.78	0.96557	85.3	47.56%	67.74%
7.01	18	26.46	0.96374	175.2	69.52%	77.97%
7.02	20	22.53	0.96539	56.4	54.23%	73.85%
7.03	20	32.01	0.96228	67.2	96.34%	82.55%
7.04	20	33.77	0.96340	54.6	77.01%	76.52%
7.05	22	61.83	0.96268	126.8	93.07%	84.64%
7.06	21	108.17	0.96160	77.9	59.76%	67.06%
7.07	21	94.59	0.95711	42.7	51.88%	86.26%
7.08	27	290.87	0.95720	42.4	78.74%	90.69%
7.09	19	26.53	0.96386	53.5	48.59%	75.78%
<b>Bilan(moyenne)</b>	<b>20.6</b>	<b>71.754</b>	<b>0.96</b>	<b>78.19</b>	<b>67.67%</b>	<b>78.31%</b>

FIG. 3.13 – Résultats de la séparation sur un polyèdre mixte avec  $I' = \{i : x_i^* \in [0, 05, 0, 95]\}$  puis lifting relaxé sur les benchmark de Chu et Beasley [17]

Nom du prob	I	Temps de séparation	Ratio de l'inégalité	Facteur d'accélération	Part de la relaxation gagnée	Fraction du ratio maximal
7.00	13	18.88	0.96902	93.9	48.71%	60.95%
7.01	17	23.2	0.96545	199.9	72.19%	74.31%
7.02	18	20.49	0.96860	62.0	68.08%	67.01%
7.03	20	32.29	0.96228	66.6	96.34%	82.55%
7.04	19	33.46	0.96403	55.1	64.37%	75.22%
7.05	20	49.5	0.96576	158.4	74.26%	77.65%
7.06	19	55.38	0.96374	152.1	35.98%	63.32%
7.07	20	56.03	0.95998	72.0	29.32%	80.49%
7.08	24	181.85	0.96225	67.8	52.76%	80.00%
7.09	18	27.29	0.96526	52.0	37.93%	72.83%
<b>Bilan(moyenne)</b>	<b>18.8</b>	<b>49.837</b>	<b>0.96</b>	<b>97.97</b>	<b>57.99%</b>	<b>73.43%</b>

FIG. 3.14 – Résultats de la séparation sur un polyèdre mixte avec  $I' = \{i : x_i^* \in [0, 1, 0, 9]\}$  puis lifting relaxé sur les benchmark de Chu et Beasley [17]

Dans cette partie, nous avons essayé de générer les meilleures inégalités possibles pour des sacs à dos multidimensionnels, puis de voir comment approcher celles-ci par diverses méthodes. Parmi celles-ci, la méthode par restriction puis lifting apparaît comme celle permettant d'obtenir le meilleur compromis entre qualité et temps de calcul, et peut être encore accélérée par l'utilisation d'heuristique ou de relaxations mixtes.

Les temps de calcul restent cependant élevés, trop pour pouvoir envisager la génération de dizaines d'inégalités par ces méthodes, et l'effet de l'ajout d'un petit nombre d'inégalités semble incertain. Le problème auquel on se heurte ici pour pouvoir accélérer encore la génération de coupe par des approches polaires est le manque de structures sur lesquelles s'appuyer pour restreindre encore plus le problème initial de séparation ou le projeter dans un espace de plus petite dimension (comme cela peut être fait pour un problème structuré comme le voyageur de commerce (c.f. paragraphe 2.6.2)). La qualité des coupes générées est néanmoins très bonne et parfois une seule coupe peut diminuer le temps de résolution de 20%. Une question qui se pose néanmoins est de savoir comment les inégalités ajoutées influencent positivement ou négativement les stratégies de branchements.

Dans les chapitres suivants, nous allons considérer une autre manière d'aborder le problème de séparation, non plus sur l'espace naturel des variables du problème, mais à l'aide de relaxations exprimées par des polyèdres continus dans des espaces de dimension supérieure.



## Chapitre 4

# Séparation et renforcement par la programmation disjonctive - Expérimentation

Dans les deux prochains chapitres, nous allons porter notre intérêt sur des méthodes de séparation basées sur la formulation de relaxations des problèmes d'optimisation dans des espaces de dimension supérieure à celle du problème original.

Dans ce chapitre nous considérerons des méthodes de séparation basées sur la programmation disjonctive. Cette technique, développée dans les années 70 par Balas [4], a été utilisée avec succès depuis le début des années 90 comme méthode de génération de coupes dans des algorithmes de Branch and Cut notamment par Balas, Ceria et Cornuéjols ([7] et [8]), Ceria et Pataki ([16]), Perregaard et Balas ([10] et [51]).

Après avoir fait quelques rappels sur la programmation disjonctive et la méthode classique de séparation par *Lift and Project*, nous proposerons une méthode de décomposition pour réaliser la séparation relativement au polyèdre constitué par l'intersection des polyèdres de *lift and project* (aussi appelée fermeture élémentaire des coupe de lift and project [19]). Nous proposerons une série de résultats comparatifs entre ces méthodes sur des problèmes de sac à dos multidimensionnels, de programmation mixte et de MAX-2SAT.

### 4.1 La programmation disjonctive

On dit qu'un problème d'optimisation est un programme disjonctif si l'ensemble de ses solutions réalisables peut être mis sous la forme d'une union de polyèdres. Reformuler un programme linéaire en nombre entiers pur ou mixte sous forme de programme disjonctif est une des techniques de base intéressantes permettant de renforcer ou de construire des inégalités valides générales.

En nous basant sur [4],[7], [8] et [51], nous allons rappeler ici quelques-uns des résultats de la programmation disjonctive permettant de générer des plans coupants. Bien que beaucoup des résultats mentionnés se généralisent à des problèmes généraux en variables entières, nous nous limiterons dans notre présentation aux problèmes purs ou mixtes en variables 0/1.

Considérons le programme linéaire 0/1 suivant, formulé de manière à ce que les inégalités bornant les valeurs des variables soient incluses dans  $A'$  et  $b'$  :

$$(PL01) \begin{cases} \max c^T x \\ t.q. : \\ A'x \leq b' \\ x \in \mathbb{Z}^n \end{cases}$$

La programmation disjonctive consiste à considérer les solutions réalisables de (PL01) comme appartenant à l'union de plusieurs polyèdres. Soit  $\mathcal{U}$  une famille finie de polyèdres telle que  $\{x \in \mathbb{Z}^n : Ax \leq b\} \subset \bigcup_{U \in \mathcal{U}} U$ ; une relaxation disjonctive de (PL01) correspondant à  $\mathcal{U}$  consiste à optimiser  $c^T x$  sur  $Q = \text{conv}(\{x : A'x \leq b'\} \cap \{\bigcup_{U \in \mathcal{U}} U\})$  ce qui s'exprime sous la forme d'un programme disjonctif :

$$(DP) \begin{cases} \max c^T x \\ t.q. : \\ A'x \leq b' \\ \bigvee_{U \in \mathcal{U}} x \in U \\ x \in \mathbb{R}^{n+} \end{cases}$$

Pour tout  $U \in \mathcal{U}$ , on note  $A^U x \leq b^U$  le système d'inégalités définissant  $\{x : A'x \leq b'\} \cap U$ . L'intérêt de la programmation disjonctive est que la description complète du polyèdre  $Q$  est connue dans un espace de dimension supérieure, ce qui permet de séparer des inégalités valides en utilisant les méthodes exposées au chapitre 2. Le théorème suivant donne la description complète de  $Q$ .

**Théorème 4.1 ([4])**  $Q$  est l'ensemble des  $x \in \mathbb{R}^n$  tel qu'il existe des vecteurs  $y^U \in \mathbb{R}^n$  et  $y_0^U \in \mathbb{R}$ ,  $U \in \mathcal{U}$  tels que :

$$\begin{cases} x - \sum_{U \in \mathcal{U}} y^U = 0 & (\alpha) \\ \sum_{U \in \mathcal{U}} y_0^U = 1 & (\beta) \\ A^U y^U - b y_0^U \leq 0 \quad \forall U \in \mathcal{U} & (v^U) \\ y_0^U \geq 0 \end{cases} \quad (4.1)$$

Pour séparer une inégalité valide sur  $Q$ , il suffit donc en appliquant les résultats du paragraphe 2.2.1 de résoudre le problème en variables continues dont les contraintes sont données par  $\Pi(Q)$  :

$$(SDP) \begin{cases} \max \alpha^T x^* - \beta \\ t.q. : \\ \alpha = v^U A^U \quad \forall U \in \mathcal{U} \\ \beta = v^U b^U \quad \forall U \in \mathcal{U} \end{cases}$$

Notons qu'avec cette formulation la taille du problème de séparation est polynomiale dans le nombre d'éléments de  $\mathcal{U}$  et dans la taille des matrices  $A^U$ . Les solutions extrémales de ce système induisent des facettes de  $Q$ .

L'efficacité de l'utilisation des coupes disjonctives d'un point de vue pratique découle de plusieurs améliorations algorithmiques que nous ne développerons pas ici :

- La possibilité de pouvoir résoudre le problème de séparation dans un espace réduit et d'effectuer ensuite un lifting rapide et efficace à l'aide des coefficients  $v^U$  et  $w^U$ .
- La possibilité de renforcer efficacement les coupes obtenues.

- La possibilité de générer simultanément plusieurs coupes en effectuant des pivots du simplexe supplémentaires dans (DP).

**Remarque 4.1 :** Rappelons que pour résoudre efficacement (SDP) il faut aussi introduire une condition de normalisation, comme il a été discuté au chapitre 2 dans le paragraphe 2.2.3, et que certaines seulement permettent d’obtenir des facettes. ||

Lorsque le nombre de disjonctions prises en compte est grand, il peut être avantageux de ne pas écrire explicitement le problème de séparation mais de passer par une méthode de décomposition comme au paragraphe 2.2.4. Nous allons maintenant montrer que le passage par une méthode de décomposition est équivalent à la séparation d’inégalité valide sur le polaire de  $Q$

Soit  $\Pi(Q)$  le polaire de  $Q$ , et  $(SPP)^Q$  le problème de séparation par le polaire sur  $Q$  :

$$(SPP)^Q \left\{ \begin{array}{l} \max \alpha^T x^* - \beta \\ t.q. : \\ \alpha^T x \leq \beta \quad \forall x \text{ point extrême de } Q \\ \alpha^T r \leq 0 \quad \forall r \text{ rayon extrême de } Q \end{array} \right.$$

On a alors :

**Proposition 4.1** *Le problème  $(SPP)^Q$  est équivalent au problème-maître résultant de l’application de la décomposition de Benders à (SDP).*

**Preuve :** D’après le lemme de Farkas, pour  $\alpha$  et  $\beta$  donnés, il existe des  $v^U$  et  $w^U$  tels que :

$$\alpha \leq v^U A + w^U A^U$$

$$\beta \geq v^U b + w^U b^U$$

si et seulement si pour tous  $x^U \in \mathbb{R}_+^n$  et  $y^U \in \mathbb{R}_+$  définis pour tout  $U \in \mathcal{U}$  et tels que :

$$A'x^U - y^U b \leq 0 \tag{4.2}$$

$$A^U x^U - y^U b^U \leq 0 \tag{4.3}$$

on a :

$$\sum_{U \in \mathcal{U}} (\alpha x^U - \beta y^U) \leq 0 \tag{4.4}$$

Le problème maître de la décomposition de Benders de (SDP) conduit alors à l’ensemble des contraintes :

$$\sum_{U \in \mathcal{U}} (\alpha x^U - \beta y^U) \leq 0 \quad \forall ((x^U, y^U), \dots) \text{ rayon extrême de (4.2) (4.3).}$$

Définissons l’application :

$$\begin{aligned} \phi : \mathbb{R}^{(n+1) \times |\mathcal{U}|} \setminus \{0\} &\rightarrow \mathbb{R}^n \\ ((x^U, y^U), \dots) &\mapsto \frac{\sum_{U \in \mathcal{U}} x^U}{\sum_{U \in \mathcal{U}} y^U} \end{aligned}$$

$x = ((x^U, y^U), \dots)$  vérifie 4.2 et 4.3 si et seulement si  $\phi(x) \in Q = \text{conv} \{A'x \leq b' \cap \{\bigcup_{U \in \mathcal{U}} U\}\}$ .  
Aussi  $x = ((x^U, y^U), \dots)$  vérifie 4.4 si et seulement si  $\alpha^T \phi(x) \leq \beta$ .

La condition nécessaire et suffisante donnée par le lemme de Farkas est donc équivalente à : pour tout  $x \in Q$ ,  $\alpha^T x \leq \beta$ .

La description par point extrêmes de  $Q$  est donc équivalente à la décomposition de Benders de (SDP).



## 4.2 Stratégies de séparation par la programmation disjonctive

Comme on l'a vu, la programmation disjonctive offre un cadre très général pour la séparation systématique d'inégalités valides coupantes. Dans la pratique, de très nombreuses façons d'utiliser la programmation disjonctive sont envisageables. Dans l'implémentation d'un algorithme utilisant la programmation disjonctive, il faut en particulier s'intéresser au choix de la taille des disjonctions à prendre en compte et au nombre de disjonctions différentes à prendre en compte.

Nous allons présenter ici trois stratégies. La première, de séparation sur des disjonctions simples à 2 termes, est la stratégie classique présentée dans [7] sous le nom de "lift and project" sur laquelle de très nombreuses expériences ont été réalisées notamment dans [8]. La seconde est une stratégie où l'on prend en compte des disjonctions de plus grande taille. La troisième consiste à séparer sur l'intersection des polyèdres issus de plusieurs disjonctions différentes.

### 4.2.1 Séparation sur des disjonctions simples

L'utilisation la plus connue des coupes disjonctives consiste à séparer successivement sur des disjonctions simples exprimant l'intégrité de chaque variable du problème (i.e. de la forme  $(x_i = 0) \vee (x_i = 1)$ ). Celle-ci a été proposée et étudiée par Balas Ceria et Cornuéjols dans [7], [8]. La séparation s'effectue au travers de la disjonction  $\mathcal{U} = \{\{x \in \mathbb{R}^n : x_i = 0\}, \{x \in \mathbb{R}^n : x_i = 1\}\}$  qui exprime l'intégrité d'une des variables 0/1 du problème. On obtient ainsi la relaxation disjonctive :

$$\left\{ \begin{array}{l} \max c^T x \\ t.q. \\ A'x \leq b' \\ (x_i = 0) \vee (x_i = 1) \\ x \in \mathbb{R}^{n+} \end{array} \right.$$

qui s'exprime sous forme polyédrale comme :

$$\left\{ \begin{array}{l} x - y^i - z^i = 0 \\ y_0^i + z_0^i = 1 \\ A'y^i - by_0^i \leq 0 \\ y_i^i - y_0^i = 0 \\ A'z^i - bz_0^i \leq 0 \\ z_i^i = 0 \end{array} \right. \forall i \in \{1, \dots, n\}.$$

Le problème de séparation associé est donc un programme linéaire comportant  $(n + 1) + 2(m + 1)$  variables :

$$(L\&P^i) \left\{ \begin{array}{l} \max \alpha^T x^* - \beta \\ t.q. : \\ \alpha = uA' - e_i u_0 \\ \alpha = vA' + e_i v_0 \\ \beta = ub' - u_0 \\ \beta = vb' \\ u \geq 0, v \geq 0, u_0, v_0 \geq 0 \end{array} \right.$$

Les coupes obtenues en séparant sur les disjonctions simples sont généralement appelées coupes de “Lift and Project”.

La séparation des solutions fractionnaires de la relaxation continue du problème s’effectue par *rounds* où l’on sépare la solution courante en exprimant successivement l’intégrité de toutes les variables binaires du problème dont la valeur dans la relaxation continue est fractionnaire. A la fin d’un round on ajoute toutes les inégalités obtenues à la formulation du problème. La relaxation continue de la nouvelle formulation peut alors être utilisée pour effectuer un nouveau round de séparation (sur les polyèdres disjonctifs issus de la nouvelle formulation du problème) ou pour brancher dans un algorithme de résolution par énumération implicite.

Pour l’utiliser cette méthode dans un algorithme de “branch and cut”, de nombreux paramètres sont à prendre en compte tels que le nombre de rounds effectués à chaque séparation (généralement 1 ou 2), la fréquence de générations d’inégalités, le choix de la condition de normalisation. L’impact de ces différents paramètres a été étudié au travers de nombreuses expériences dans [8].

#### 4.2.2 Séparation sur des disjonctions multiples

Une première approche pour améliorer la séparation sur des disjonction simples à deux termes est de prendre en compte des *disjonctions multiples*. Cette approche a été en particulier étudiée par Perregaard et Balas [51], et par Ouzia [47]. Dans la séparation sur des disjonctions multiples, on exprime simultanément l’intégrité de  $k$  variables 0/1  $x_{i_1}, \dots, x_{i_k}$  en énumérant toutes les valeurs que peuvent prendre ces variables par la disjonction à  $2^k$  termes :

$$\bigvee_{\delta \in \{0,1\}^k} (x_{i_1} = \delta_1, \dots, x_{i_k} = \delta_k)$$

Le problème de séparation peut alors être résolu soit par son expression polyédrale complète comme un programme linéaire de  $n + 1 + 2^k(m + 1)$  variables et  $2^k(n + 1)$  contraintes, soit par une méthode de décomposition de Benders au travers de la résolution sur le polaire (les expériences de calculs effectuées aussi bien par Perregaard et Balas [51] que par Ouzia [47] semblent indiquer que pour  $k \geq 3$  la méthode par décomposition devient la plus rapide).

Par rapport à la séparation sur des disjonctions simples, un aspect supplémentaire devant être considéré dans la mise en œuvre d’un algorithme de séparation est le choix des disjonctions sur lesquelles séparer, le nombre de disjonction différentes est  $C_k^n$ . Dans le cas des disjonctions simples, en général toutes les disjonctions (où une partie significative des ces disjonctions simples) sont explorées. Lorsque l’on considère des disjonctions multiples le nombre de disjonctions différentes est trop important pour que toutes soient explorées (même lorsqu’on prend simultanément en compte l’intégrité

de peu de variables). Un choix doit donc être effectué sur les disjonctions “intéressantes”. Ce choix stratégique doit à la fois porter sur le nombre des disjonctions que l’on considère à chaque round et sur la composition de chacune de ces disjonctions.

Dans [51], les disjonctions sont choisies en effectuant un petit branch and bound de taille très réduite. Dans [47], de l’ordre de  $n$  disjonctions sont choisies en se basant sur la solution continue à séparer.

### 4.2.3 Séparation relativement à l’intersection des polyèdres disjonctifs

Une autre approche pour améliorer la séparation des coupes de “lift and project” est de chercher à générer un ensemble d’inégalités issues des polyèdres disjonctifs simples donnant le meilleur renforcement possible de la relaxation. Ceci peut s’effectuer au travers du polyèdre constitué par l’intersection de tous les polyèdres sur les disjonctions simples :

$$P_{L\&P} = \bigcap_{i \in \{1, \dots, n\}} \text{conv}(\{x \in \mathbb{R}^n : A^i x \leq b^i, x_i \in \{0, 1\}\}) \quad (4.5)$$

qui est aussi appelé fermeture élémentaire des coupes de lift and project [19]. Une description polyédrique de cet ensemble est donnée par un système à  $n(n+2)$  variables où sont exprimées successivement chacune des disjonctions simples :

$$\tilde{A}x + \tilde{B}\tilde{y} \leq \tilde{b} \quad (4.6)$$

Dans cette formulation,  $\tilde{y} = (y_0^1, y^1, z_0^1, z^1, \dots, y_0^n, y^n, z_0^n, z^n)$ ,  $\tilde{A}$  est composée de  $n$  blocs  $\tilde{A}^i$  de taille  $(2m+n+3) \times n$  :

$$\tilde{A} = \begin{pmatrix} \tilde{A}^1 \\ \vdots \\ \tilde{A}^n \end{pmatrix}. \quad (4.7)$$

$\tilde{B}$  est une matrice bloc-diagonale où chaque bloc  $\tilde{B}^i$  est de taille  $(2m+n+3) \times (2n+2)$  :

$$\tilde{B} = \begin{pmatrix} \tilde{B}^1 & & & \\ & & 0 & \\ & & \ddots & \\ & 0 & & \tilde{B}^n \end{pmatrix}, \quad (4.8)$$

$\tilde{b}$  peut être décomposé en blocs de  $2n+2$  éléments :

$$\tilde{b} = \begin{pmatrix} \tilde{b}^1 \\ \vdots \\ \tilde{b}^n \end{pmatrix}, \quad (4.9)$$

de manière à ce que le bloc de lignes du système  $\tilde{A}^i x + \tilde{B}^i y^i \leq \tilde{b}^i$  (ou  $y^i$  représente le vecteur de  $\mathbb{R}^n$  de composantes  $y_j^i$ ) soit l'expression du polyèdre issu de la disjonction  $i$  :

$$X^i \begin{cases} x - y^i - z^i = 0 \\ y_0^i + z_0^i = 1 \\ A' y^i - b y_0^i \leq 0 \\ y_i^i - y_0^i = 0 \\ A' z^i - b z_0^i \leq 0 \\ z_i^i = 0 \end{cases} \quad \forall i \in \{1, \dots, n\}. \quad (4.10)$$

Malgré la taille de ce système, on peut optimiser efficacement en tirant parti de la structure bloc-diagonale de la matrice  $\tilde{B}$ . Pour cela, on effectue la résolution par décomposition de Benders au travers de la résolution du problème-maître :

$$\begin{cases} \max c^T x \\ t.q. \\ \alpha^T x \leq \beta \quad \forall (\alpha, \beta) \in \Pi(X^i), \forall i \in \{1, \dots, n\} x \in [0, 1]^n \end{cases} \quad (4.11)$$

et des sous-problèmes donnés par les problèmes de séparation d'inégalité valide sur chacun des  $X^i$  séparément (cf. algorithme 3). Notons qu'ici, à la différence de la séparation par rounds, le problème de séparation n'est pas remis à jour à chaque fois qu'une inégalité est ajoutée..

---

**Algorithme 3** Optimisation sur  $P_{L\&P}$ 


---

**répéter**

**pour**  $i \in \{1, \dots, n\}$  **faire**

    Trouver  $x^*$  l'optimum du problème-maître courant

**si**  $x_i^* \notin \{0, 1\}$  **alors**

    Chercher une inégalité valide  $(\alpha, \beta)$  pour  $X^i$  coupant  $x^*$

**si**  $\alpha^T x^* > \beta$  **alors**

        Ajouter  $\alpha^T x \leq \beta$  à la description du polyèdre

**fin si**

**fin si**

**fin pour**

**tant que** Aucune inégalité n'a été ajoutée à la dernière itération

---

Rappelons qu'ici notre but n'est pas d'ajouter toutes les inégalités trouvées à la formulation du problème. En particulier, on ne désire pas ajouter les inégalités qui ne sont pas des facettes de  $P_{L\&P}$ . Les inégalités générées ne sont en général pas des faces de l'intersection  $P_{L\&P}$  (en effet  $P_{L\&P}$  étant le polyèdre constitué par l'intersection des polyèdres  $X^i$ , toutes les facettes de  $P_{L\&P}$  sont des facettes d'au moins un des  $X^i$ , mais l'inverse n'est pas vrai). Plusieurs méthodes sont envisageable pour générer uniquement des facettes de  $P_{L\&P}$ . Ici nous avons choisi de ne pas tester explicitement la dimension des inégalités générées mais plutôt de conserver pour renforcer la formulation du problème uniquement les inégalités saturées à l'optimum de  $P_{L\&P}$ . On ne peut garantir absolument que ces inégalités sont des facettes, mais on peut raisonnablement penser qu'elles le sont en général (et ce sont au moins des faces), et on est sûr d'ajouter l'ensemble d'inégalités donnant le meilleur renforcement de la relaxation continue. Une fois ces inégalités ajoutées à la formulation du problème, on peut relancer une séparation sur  $P_{L\&P}$  de la nouvelle formulation (précisons bien que celle ci n'est

pas  $P_{L\&P}(P_{L\&P}(X))$ .

Une difficulté qui se présente pour optimiser efficacement sur  $P_{L\&P}$  porte sur le choix des disjonctions sur lesquelles séparer. Dans la séparation par lift-and-project sur les disjonctions simples, à chaque round on sépare sur les variables  $i$  dont la valeur n'est pas entière dans le point à couper  $x^*$ . Comme le point à couper est un point extrême de la relaxation continue, on a la garantie qu'une inégalité va ainsi être générée (autrement dit que  $x^* \notin X^i$ ). Avec l'algorithme 3, le point à couper n'est plus une solution extrême de la formulation sur laquelle sont exprimées les disjonctions, et on ne peut donc garantir qu'une coupe va être générée sans tester explicitement l'appartenance (c'est à dire en générant la coupe). Nous proposerons trois variantes de l'algorithme 3 qui diffèrent par la manière dont sont choisies les disjonctions sur lesquelles on cherche à séparer.

La première est exactement celle de l'algorithme 3, où l'on parcourt les variables et on teste explicitement l'appartenance en résolvant le problème de séparation.

Dans les deux variantes suivantes, on ne résoudra pas le problème d'appartenance pour toutes les disjonctions. Dans la seconde, on parcourt les variables dans l'ordre lexicographique, mais si, pour une variable dont la valeur n'est pas entière, on n'a pas pu générer d'inégalité la dernière fois que l'on a essayé, on ne cherche pas à en générer (on dira dans la suite qu'une telle variable est temporairement inactive). Evidemment pour aller jusqu'à l'optimum, si aucune inégalité n'est trouvée après avoir parcouru toutes les variables actives, il faut reparcourir les variables inactives pour tester l'appartenance du point courant sur les disjonctions correspondantes ; si de nouvelles inégalités sont alors générées les variables correspondantes deviennent à nouveau actives. On effectue cela jusqu'à ce qu'aucune des variables actives ou inactives ne coupe la solution courante.

L'idée de cette stratégie heuristique de choix est qu'au fur et à mesure des itérations le point à couper rentre dans certains des polyèdres disjonctifs et qu'il n'y a plus alors de coupe à générer sur ces disjonctions.

Dans la troisième variante, on a aussi des variables inactives et actives. Par contre, ici les variables ne sont pas parcourues dans l'ordre lexicographique, mais dans l'ordre de leur proximité à  $\frac{1}{2}$ . Après avoir ajouté une coupe et recalculé le point à séparer, on ordonne les variables et on sépare sur la variable active la plus proche de  $\frac{1}{2}$ .

### 4.3 Comparaisons pratiques des différentes stratégies de séparation

Nous avons mené plusieurs séries d'expériences pour comparer les différentes stratégies de génération des coupes disjonctives. Nous allons en présenter les résultats sur trois catégories de problèmes : sacs à dos multidimensionnels (paragraphe 4.3.1), problèmes mixtes généraux de la MIPLIB (paragraphe 4.3.2), et problème de type MAX-2SAT (paragraphe 4.3.3).

Nous nous sommes attaché en particulier à comparer la séparation par "lift and project" classique d'une part et d'autre part l'intersection des polyèdres disjonctifs  $P_{L\&P}$  par l'algorithme 3. Pour rendre possible la comparaison directe entre ces deux méthodes, nous donnerons le renforcement obtenu en effectuant plusieurs rounds de lift and project pour un temps équivalent à celui pris par l'optimisation sur l'intersection des polyèdres.

Nous comparerons aussi entre elles les différentes variantes de l'algorithme 3.

**Remarque 4.2 :** Contrairement au chapitre précédent, nous ne donnons pas ici le détail des ratios des inégalités trouvés. Ils sont en général assez mauvais (en moyenne 0.994 pour les problèmes de sac à dos multidimensionnels) et peu significatifs car, ici, le renforcement ne résulte pas tant de la force individuelle de chacune des inégalités que de la combinaison de l'ensemble des inégalités ajoutées. ||

### 4.3.1 Séparation sur des sacs à dos multidimensionnels

Le tableau 4.1 compare les renforcements obtenus respectivement avec un round de lift and project, l'optimisation sur l'intersection des polyèdres et le lift and project itéré sur 30 problèmes de sacs à dos multidimensionnels à 100 variables et 30 contraintes de Chu et Beasley [17].

Le renforcement est calculé comme la part du gap de la relaxation continue comblée par l'ajout des inégalités (pour plus d'indications sur cette mesure on peut se reporter au paragraphe 3.1). Nous donnons aussi le temps mis pour optimiser sur l'intersection des polyèdres en secondes.

Au vu des résultats du tableau 4.1, l'optimisation sur  $P_{L\&P}$  (l'intersection des polyèdres disjoints) donne des renforcements assez nettement meilleurs qu'un round de lift and project simple. Par ailleurs, on note que le lift and project itéré donne des renforcements un peu moins bons mais comparables.

Dans la seconde série d'expériences sur les sacs à dos multidimensionnels que nous rapportons ici, nous avons répété plusieurs fois la procédure d'optimisation sur  $P_{L\&P}$ . Dans la séparation sur l'intersection des polyèdres, lorsque l'optimum a été atteint nous ajoutons les inégalités saturées par la solution courante (qui sont des faces de l'intersection) à la description du problème et nous cherchons des inégalités sur le nouveau polyèdre obtenu (c'est à dire en ajoutant les inégalités saturées au problème de séparation).

A priori l'approche par Lift and Project classique va permettre d'ajouter plus d'inégalités à la formulation du problème (et à celle du problème de séparation), alors que l'approche relative à  $P_{L\&P}$  ajoutera beaucoup moins d'inégalités mais ajoutera un ensemble d'inégalités permettant à chaque fois le meilleur renforcement.

Le tableau 4.2 rapporte les expériences faites pour comparer ces deux approches. Les mesures donnent des moyennes prises sur les 30 problèmes tests du tableau précédent. Chaque ligne correspond à une optimisation supplémentaire sur l'intersection des polyèdres après ajout des inégalités trouvées à l'itération précédente. Les mesures rapportées sont le temps de calcul cumulé et le renforcement pour l'optimisation sur l'intersection des polyèdres et pour le lift and project itéré pour un temps équivalent. On rapporte aussi le nombre de rounds effectués.

Notons que sur ces problèmes seule la première optimisation sur  $P_{L\&P}$  donne un renforcement conséquent (8% en moyenne), le renforcement donné par chaque itération suivante étant de l'ordre de 0,2% par itération. La comparaison des deux stratégies fait apparaître un avantage léger mais sensible de l'optimisation sur  $P_{L\&P}$  pour ces problèmes.

Nous allons maintenant comparer, entre elles les différentes variantes pour l'optimisation sur  $P_{\{L\&P\}}$ . Le tableau 4.3 indique les résultats de temps de calcul pour les variantes 2 (où on a les

Instance	Renforcement L&P	Renforcement P_(L&P)	Renforcement L&P itéré	Temps P_(L&P)	Temps L&P itéré
7.00	3,84%	11,58%	10,82%	57,57	61,54
7.01	2,32%	9,39%	8,24%	48,83	56,1
7.02	2,63%	6,66%	5,65%	54,5	54,93
7.03	1,35%	6,45%	5,05%	54,03	60,39
7.04	1,79%	5,02%	4,48%	48,08	57,99
7.05	1,98%	7,29%	6,17%	84,74	84,84
7.06	1,76%	6,05%	6,47%	86,01	87,74
7.07	1,68%	8,59%	8,18%	93,19	98,15
7.08	1,40%	4,87%	4,66%	51,84	54,54
7.09	1,99%	6,52%	5,68%	43,27	45,89
7.10	1,67%	7,70%	6,89%	54,16	57,38
7.11	1,93%	8,18%	7,30%	77,11	88,34
7.12	3,32%	9,44%	8,66%	98,99	110,11
7.13	2,63%	8,60%	8,55%	95,22	99,66
7.14	1,02%	6,30%	6,30%	53,89	60,56
7.15	2,18%	6,62%	6,27%	56,85	62,8
7.16	2,11%	8,77%	7,98%	45,56	52,75
7.17	5,10%	10,78%	10,79%	64,87	69,6
7.18	3,49%	10,97%	11,91%	75,55	79,01
7.19	2,86%	7,27%	7,32%	58,61	62,68
7.20	3,69%	8,06%	8,49%	46,64	46,65
7.21	1,35%	6,24%	5,65%	58,19	60,2
7.22	1,18%	6,23%	5,90%	47,66	50,37
7.23	3,88%	10,23%	10,38%	60,03	61,56
7.24	3,78%	9,58%	9,10%	87,94	92,22
7.25	5,07%	9,78%	8,54%	70,37	74,02
7.26	4,03%	9,54%	8,98%	75,85	78,79
7.27	2,24%	9,14%	8,26%	90,75	95,67
7.28	4,57%	10,91%	10,64%	92,56	103,42
7.29	2,79%	9,40%	9,31%	50,43	54,78

FIG. 4.1 – Renforcement par l'optimisation sur  $P_{L\&P}$  (exemples de Chu et Beasley [17])

Num inter	P_(L&P)			L&P itéré		
	Iterations maitre Benders	Temps cumulé	Renforcement	Iterations maitres (rounds)	Temps	Renforcement
1	6,000	28,055	8,20%	5,700	32,293	7,75%
2	8,667	42,960	9,05%	7,533	46,933	8,50%
3	10,300	52,798	9,36%	8,667	56,428	8,85%
4	11,600	61,210	9,57%	9,733	66,699	9,17%
5	12,900	69,814	9,76%	10,500	73,871	9,36%
6	14,100	78,003	9,91%	11,433	83,595	9,58%
7	15,233	85,899	10,04%	12,233	91,044	9,73%
8	16,367	94,120	10,15%	13,067	100,109	9,88%
9	17,600	103,322	10,28%	13,800	107,945	10,03%
10	18,767	112,484	10,40%	14,667	117,287	10,17%

FIG. 4.2 – Comparaison de génération simple et sur  $P_{L\&P}$  (moyenne sur les 30 instances de Chu etBeasley)

variables actives et inactives) et 3 (où on traite les variables par ordre de leur distance à 0.5). Pour chacune de ces deux variantes, on indique les temps de calculs et l'accélération par rapport à la variante 1 (donnée par le temps d'exécution de la variante 1 divisé par le temps d'exécution de la variante considérée).

Les deux variantes permettent d'accélérer sensiblement le temps de résolution sur tous les problèmes, la variante 2 étant en moyenne plus efficace. Pour la variante 2, l'accélération est due directement à la diminution du nombre de fois où on teste l'appartenance du point à couper. Avec la variante 1 on teste en moyenne 192 fois l'appartenance et dans 43% des cas comme le point appartient au polyèdre disjonctif considéré aucune coupe n'est générée. Avec la variante 2, l'appartenance est testée en moyenne 131 fois, et pour seulement 20% des cas aucune coupe n'est générée (notons que le nombre total de tests permettant de générer une coupe est à peu près équivalent 108 en moyenne pour la variante 1 et 103 pour la variante 2). Dans le cas de la variante 3, par contre le nombre total de tests d'appartenance est supérieur (en moyenne 246 dont 24% inutiles). L'accélération par rapport à la variante 1 vient ici du fait que l'on effectue souvent plusieurs fois de suite la séparation sur le même polyèdre disjonctif, ce qui permet de diminuer les temps de mise à jour et de recommencer l'optimisation à partir d'une meilleure solution.

### 4.3.2 Résultats de séparation sur des problèmes en nombres entiers généraux

Le tableau 4.5 rapporte les expériences réalisées sur quelques problèmes de la MIPLIB. Les mesures données sont identiques à celles du tableau 4.1. Nous avons traité des problèmes de la MIPLIB ayant au plus 400 variables et sans variable entière non 0/1. Nous indiquons dans le tableau 4.4 les principales caractéristiques des problèmes traités.

Sur ces problèmes mixtes généraux, les résultats du tableau 4.5 ne montrent pas un avantage net pour l'une ou l'autre des méthodes. Les renforcements obtenus pour le lift and project itéré sont meilleurs dans 8 problèmes sur 18.

Le tableau 4.6 présente les résultats d'expériences portant sur la comparaison des différentes va-

Instance	Temps variante 2	Accélération variante 2	Temps variante 3	Accélération variante 3
7.00	27,74	1,44	35,14	1,64
7.01	27,74	1,76	35,61	1,37
7.02	20,60	2,65	27,82	1,96
7.03	26,93	2,01	54,01	1,00
7.04	24,06	2,00	37,31	1,29
7.05	37,58	2,25	41,33	2,05
7.06	28,63	3,00	49,51	1,74
7.07	37,90	2,46	47,23	1,97
7.08	28,51	1,82	38,42	1,35
7.09	22,27	1,94	27,86	1,55
7.10	31,50	1,72	37,23	1,45
7.11	33,46	2,30	34,65	2,23
7.12	52,51	1,89	42,91	2,31
7.13	50,45	1,89	48,95	1,95
7.14	36,72	1,47	36,78	1,47
7.15	31,62	1,80	36,57	1,55
7.16	33,12	1,38	29,41	1,55
7.17	34,71	1,87	38,73	1,67
7.18	31,88	2,37	40,14	1,88
7.19	42,12	1,39	45,04	1,30
7.20	37,60	1,24	49,77	0,94
7.21	39,80	1,46	51,20	1,14
7.22	38,95	1,22	50,25	0,95
7.23	34,64	1,73	42,15	1,42
7.24	39,01	2,25	45,79	1,92
7.25	41,28	1,70	54,98	1,28
7.26	43,57	1,74	72,76	1,04
7.27	46,38	1,96	52,57	1,73
7.28	46,11	2,01	50,58	1,83
7.29	37,36	1,35	47,36	1,06
Moyenne	35,49	1,87	43,40	1,55

FIG. 4.3 – Comparaisons des différentes méthodes d'optimisation sur  $P_{L\&P}$

Nom	N contraintes	N var	N var 0/1	Opt
bm23	20	27	pure	34
misc01	54	83	82	563,5
misc02	39	59	58	1690
mod013	62	96	48	280,95
p0033	16	33	pure	3089
p0040	23	40	pure	62027
pipex	25	48	pure	788,26
sample2	45	67	21	375
sentoy	30	60	pure	-7772
lseu	28	89	pure	1120
misc03	96	160	159	3360
rgn	24	180	100	82,2
mod008	6	319	pure	307
modglob	291	422	98	20740508
misc07	212	260	259	2810
p0282	241	282	pure	258411
pp08a	134	240	64	7350
vpm1	234	378	168	20

FIG. 4.4 – Description des problèmes de la MIPLIB traités

Instance	Renforcement L&P	Renforcement P_(L&P)	Renforcement L&P itéré	Temps P_(L&P)	Temps L&P itéré
bm23	6,62%	18,65%	25,81%	4,00	4,28
misc01	0,00%	19,21%	3,65%	24,06	24,11
misc02	0,00%	40,27%	3,43%	5,68	5,72
mod013	4,41%	17,56%	20,51%	0,95	1,70
p0033	0,45%	8,19%	57,23%	0,45	0,54
p0040	6,68%	23,60%	28,94%	0,37	0,53
pipex	0,79%	4,68%	6,28%	0,50	0,61
sample2	3,13%	34,83%	3,91%	1,16	1,18
sentoy	8,60%	22,40%	30,23%	29,64	31,28
lseu	14,84%	16,58%	32,45%	3,32	3,84
misc03	0,00%	40,21%	10,85%	88,05	88,41
rgn	0,00%	11,88%	1,73%	16,61	20,68
mod008	0,86%	9,02%	11,38%	24,24	33,10
modglob	0,30%	57,10%	65,21%	182,37	195,37
misc07	0,00%	11,44%	0,00%	958,83	958,26
p0282	50,13%	93,90%	92,12%	98,34	100,59
pp08a	41,79%	79,29%	85,41%	26,10	35,57
vpm1	11,88%	36,28%	31,42%	11,59	15,21
Moyenne	8,36%	30,28%	28,36%	82,01	84,50

FIG. 4.5 – Comparaisons de génération simple et sur  $P_{L\&P}$ , problèmes de la MIPLIB

riantes. Ici les variantes 2 et 3 ne sont pas uniformément meilleures que la variante 1, et, sur plusieurs problèmes, la variante 3 est nettement plus lente que les deux autres. Cependant, en moyenne, la variante 3 est plus de deux fois plus rapide que la variante 1. Au niveau du nombre de tests d'appartenance effectués, avec la variante 1, on effectue en moyenne 509 tests dont 60% sont inutiles. Avec la variante 2, on effectue en moyenne 300 test dont 53% sont inutiles (le nombre d'inégalités générées est donc nettement inférieur à celui de la variante 1). Avec la variante 3, en moyenne 294 tests sont effectuée dont 40% sont inutiles. Contrairement au cas des sacs à dos multidimensionnels, ici la variante 3 permet donc de diminuer le nombre de tests inutiles.

Instance	Temps variante 2	Accélération variante 2	Temps variante 3	Accélération variante 3
bm23	2,05	1,95	2,42	1,65
misc01	19,52	1,23	6,03	3,99
misc02	4,50	1,26	5,11	1,11
mod013	0,38	2,50	0,18	5,28
p0033	0,54	0,83	0,75	0,60
p0040	0,43	0,86	0,34	1,09
pipex	0,70	0,71	0,91	0,55
sample2	1,65	0,70	1,63	0,71
sentoy	6,90	4,30	6,17	4,80
lseu	4,11	0,81	1,78	1,87
misc03	44,97	1,96	16,24	5,42
rgn	24,64	0,67	18,55	0,90
mod008	49,49	0,49	36,39	0,67
modglob	51,02	3,57	19,48	9,36
misc07	947,41	1,01	1906,58	0,50
p0282	45,31	2,17	67,00	1,47
Moyenne	75,23	1,56	130,60	2,50

FIG. 4.6 – Comparaisons des différentes méthodes d'optimisation sur  $P_{L\&P}$  pour les problèmes de la miplib

### 4.3.3 Résultats sur des problèmes de MAX-2SAT

Le problème MAX-2SAT est une restriction du problème MAX-SAT (présenté au paragraphe 1.2.6), consistant à satisfaire un nombre maximal de clauses comprenant chacune deux littéraux. Un problème de MAX2-SAT est défini par un ensemble de clauses à satisfaire :  $c_1 \wedge \dots \wedge c_m$ . Où chaque clause est donnée par une disjonction sur deux littéraux ou leur complément du type  $x_i \vee x_j$ ,  $\bar{x}_i \vee x_j$  ou  $\bar{x}_i \vee \bar{x}_j$ .

Ce problème peut se modéliser de plusieurs manières sous la forme d'un programme linéaire en variables pseudo-booléennes. Dans la modélisation que nous avons choisie, on a une variable pseudo-booléenne  $x_i$  par littéral et une variable continue par clause  $z_s$ .

$$(MAX2SAT) \left\{ \begin{array}{l} \min \sum_{i=1}^m z_i \\ t.q. : \\ -x_i - x_j - z_s \leq -1 \quad \text{pour les clauses } c_s \text{ du type } x_i \vee x_j \\ x_i - x_j - z_s \leq 0 \quad \text{pour les clauses } c_s \text{ du type } \bar{x}_i \vee x_j \\ x_i + x_j - z_s \leq 1 \quad \text{pour les clauses } c_s \text{ du type } \bar{x}_i \vee \bar{x}_j \\ x \in \{0, 1\}^n, z \in [0, 1]^m \end{array} \right.$$

Les tableaux 4.7, 4.8 et 4.9 rapportent des expériences faites sur des problèmes de MAX-2SAT générés à l'aide du générateur aléatoire de Bart Selman<sup>1</sup>. Les mesures données ici sont la valeur optimale du problème, les renforcements obtenus respectivement en optimisant sur l'intersection des polyèdres et en itérant la procédure de lift and project pour un temps équivalent, et le temps d'optimisation sur l'intersection des polyèdres.

Problem	v(MIP)	Renforcement pour L&P itéré	Renforcement pour inter(X_i)	T inter(x_i)
20.150.09	-19	82,42%	100,00%	9,41
20.150.09	-17	74,45%	100,00%	6,9
20.150.09	-19	64,31%	100,00%	7,25
20.150.09	-14	94,05%	100,00%	4,52
20.150.09	-17	81,05%	100,00%	5,48
20.150.09	-20	76,67%	99,00%	9,79
20.150.09	-19	62,17%	100,00%	7,28
20.150.09	-19	84,43%	100,00%	7,69
20.150.09	-18	89,26%	100,00%	8,02
20.150.09	-18	83,33%	100,00%	6,34
20.150.09	-16	84,38%	100,00%	7,51
20.150.09	-14	86,61%	100,00%	3,87
20.150.09	-14	93,27%	100,00%	3,79
20.150.09	-17	73,53%	100,00%	6,35
20.150.09	-17	78,50%	100,00%	7,22
20.150.09	-17	87,06%	100,00%	5,68
20.150.09	-18	78,47%	100,00%	6,76
20.150.09	-22	74,95%	99,59%	9,63
20.150.09	-16	76,72%	100,00%	5,06
20.150.09	-17	80,80%	100,00%	5,47

FIG. 4.7 – Comparaisons de génération simple et sur  $P_{L\&P}$  pour des problèmes de Max2Sat à 20 littéraux et 150 clauses

<sup>1</sup>disponible à ftp://dimacs.rutgers.edu/pub/challenge/sat/contributed/selman/

Problem	v(MIP)	Renforcement pour L&P itéré	Renforcement pour inter(X_i)	T inter(x_i)
cnf2.50.215.566406	-13	77,66%	100,00%	7,34
cnf2.50.215.566407	-18	82,09%	100,00%	10,11
cnf2.50.215.566408	-14	85,61%	100,00%	5,02
cnf2.50.215.566409	-16	83,13%	100,00%	7,77
cnf2.50.215.566410	-15	95,56%	100,00%	5,96
cnf2.50.215.566411	-15	87,54%	100,00%	5,71
cnf2.50.215.566412	-16	91,64%	100,00%	6,49
cnf2.50.215.566413	-15	82,58%	100,00%	10,11
cnf2.50.215.566414	-17	79,41%	100,00%	7,94
cnf2.50.215.566415	-18	82,72%	100,00%	9,93
cnf2.50.215.566416	-16	88,92%	100,00%	9,49
cnf2.50.215.566417	-10	87,04%	100,00%	3,59
cnf2.50.215.566418	-17	87,66%	100,00%	8,65
cnf2.50.215.566419	-18	82,43%	100,00%	18,28
cnf2.50.215.566420	-18	81,07%	100,00%	13,03
cnf2.50.215.566421	-20	72,62%	97,21%	30,09
cnf2.50.215.566422	-16	85,45%	100,00%	13,3
cnf2.50.215.566423	-18	81,19%	100,00%	15,27
cnf2.50.215.566424	-16	87,12%	100,00%	9,04
cnf2.50.215.566425	-15	73,83%	100,00%	8,34

FIG. 4.8 – Comparaisons de génération simple et sur  $P_{L\&P}$  pour des problèmes de Max2Sat à 50 littéraux et 215 clauses

Problem	v(MIP)	Renforcement pour L&P itéré	Renforcement pour inter(X_i)	T inter(x_i)
cnf2.50.350.546101	-35	77,12%	100,00%	46,25
cnf2.50.350.546102	-40	75,83%	99,65%	296,83
cnf2.50.350.546103	-34	75,40%	100,00%	46,42
cnf2.50.350.546104	-36	64,67%	100,00%	30,41
cnf2.50.350.546105	-39	70,57%	99,81%	150,84
cnf2.50.350.546106	-38	68,27%	100,00%	62,55
cnf2.50.350.546107	-41	65,94%	99,74%	170,31
cnf2.50.350.546108	-41	69,02%	100,00%	78,72
cnf2.50.350.546109	-38	70,40%	100,00%	37,01
cnf2.50.350.546110	-38	74,11%	100,00%	73,32
cnf2.50.350.546111	-43	80,16%	98,45%	147,32
cnf2.50.350.546112	-43	68,39%	99,21%	286,02
cnf2.50.350.546113	-39	71,55%	100,00%	66,78
cnf2.50.350.546114	-36	75,93%	100,00%	50,68
cnf2.50.350.546115	-39	61,55%	100,00%	65,07
cnf2.50.350.546116	-40	74,17%	100,00%	102,67
cnf2.50.350.546117	-33	66,67%	100,00%	22,35
cnf2.50.350.546118	-39	75,15%	98,82%	125,12
cnf2.50.350.546119	-42	70,32%	99,55%	205,81
cnf2.50.350.546120	-43	70,73%	97,87%	213,43

FIG. 4.9 – Comparaisons de génération simple et sur  $P_{L\&P}$  pour des problèmes de Max2Sat à 50 littéraux et 350 clauses

Sur ces problèmes de MAX-2SAT, on peut constater au vu des résultats obtenus que l'intersection des polyèdres semble être une très bonne relaxation permettant d'obtenir l'optimum entier dans 50 problèmes sur 60. Le lift and project même itéré donne un renforcement significativement moins bon.

S'agissant de la comparaison entre les différentes variantes pour le calcul de  $P_{L\&P}$ , les trois variantes sont quasiment équivalentes sur ces problèmes (les variantes 2 et 3 sont toutes deux 1,1 fois plus rapides en moyenne sur tous les problèmes de MAX2SAT). Avec la variante 1, l'appartenance est testée en moyenne 192 fois, et 4% seulement des tests ne permettent pas de générer de coupes. Avec la variante 2 l'appartenance est testée en moyenne 191 fois, et 4% aussi de ces tests ne permettent pas de générer de coupe (dans le détail les variantes 1 et 2 sont presque identiques sur ces problèmes, comme très peu de variables deviennent inactives). Avec la variante 3 l'appartenance est testée en moyenne 277 fois et ne permet pas de générer de coupe dans 5% des cas.

Dans ce chapitre, nous avons voulu explorer de nouvelles stratégie pour la génération de coupes par lift-and-project en nous basant sur l'optimisation sur  $P_{L\&P}$ . Ces stratégies permettent d'obtenir de meilleurs renforcements pour un temps équivalent dans beaucoup de cas. Ceci est dû notamment au fait qu'on met à jour plus souvent le point à séparer, et qu'on exploite au maximum les problèmes de séparation avant de les modifier (par l'ajout de coupes à leur formulation).

Pour essayer d'obtenir des renforcements encore meilleurs, dans le prochain chapitre nous allons nous intéresser à la séparation sur un polyèdre contenu dans  $P_{L\&P}$ , la relaxation de Sherali-Adams. chapitre Technique de reformulation-linéarisation (RLT) : une nouvelle approche par relaxation et décomposition itérative Dans ce chapitre, nous nous intéressons à l'utilisation de techniques dites de "reformulation-linéarisation" introduites par Sherali et Adams [54], [55] pour le renforcement des relaxations continues. Cette méthode s'exprime habituellement par une reformulation dans un espace de dimension supérieure puis linéarisation. Après avoir brièvement rappelé le principe de cette technique (cf. paragraphe 4.4), nous décrirons la relaxation de Sherali-Adams comme une restriction du polyèdre  $P_{L\&P}$  décrit par la programmation disjonctive (cf. paragraphe 4.5). Nous étudierons ensuite l'optimisation sur la relaxation de Sherali-Adams par des méthodes de décomposition et de plans coupants (cf. paragraphe 4.6.1). Nous proposerons notamment une nouvelle approche par relaxation et décomposition itérative permettant d'optimiser en fonction du temps de calcul disponible, sur une famille de polyèdres comprise entre  $P_{L\&P}$  et la relaxation de Sherali-Adams. Nous donnerons enfin au paragraphe 4.7 des résultats d'expérimentations sur des problèmes de la MIPLIB.

## 4.4 Expression de la relaxation de Sherali-Adams par reformulation-linéarisation

Soit  $P = \{x \in \{0, 1\}^n : Ax \leq b\}$ . Nous allons formuler la relaxation de Sherali-Adams d'ordre  $k \in \{1, \dots, n\}$  telle qu'elle est exprimée dans [54].

Soit  $B^k$  l'ensemble des couples de sous-ensembles de  $\{1, \dots, n\}$  disjoints dont l'union est de cardinalité  $k$  :

$$B^k = \{(J_1, J_2) : J_1 \cap J_2 = \emptyset, J_1 \cup J_2 \subseteq \{1, \dots, n\} \text{ et } |J_1 \cup J_2| = k\}$$

Pour tout  $(J_1, J_2) \in B^k$ , on définit le polynôme,  $F(J_1, J_2) = \prod_{j \in J_1} x_j \prod_{j \in J_2} (1 - x_j)$ .

Pour effectuer la reformulation, on commence par multiplier la matrice par chacun des polynômes  $F(J_1, J_2)$  de manière à obtenir le système :

$$\begin{aligned} F(J_1, J_2)(Ax - b) &\leq 0 & \forall (J_1, J_2) \in B^k \\ 0 \leq F(J_1, J_2)x &\leq F(J_1, J_2) & \forall (J_1, J_2) \in B^k \end{aligned} \quad (4.12)$$

Ensuite, on développe les polynômes, puis on linéarise le système en remplaçant  $x_j^2$  par  $x_j$  et en substituant à chaque produit  $\prod_{i \in I} x_i$  une variable  $y_I \geq 0$  (en convenant des indentifications  $y_{\{i\}} = x_i$  et  $y_\emptyset = 1$ ).

Pour écrire simplement la nouvelle formulation du problème ainsi obtenue, il est commode d'utiliser des variables complémentaires  $y_{(J_1, J_2)}$  représentant l'expression linéarisée du polynôme  $F(J_1, J_2) = \prod_{j \in J_1} x_j \prod_{j \in J_2} (1 - x_j) = \sum_{J \subseteq J_2} (-1)^{|J|} \prod_{j \in J_1 \cup J} x^j$  :

$$y_{(J_1, J_2)} = \sum_{J \subseteq J_2} (-1)^{|J|} y_{J_1 \cup J} \quad (4.13)$$

En notant  $A^i$  la  $i^{\text{me}}$  colonne de la matrice  $A$ , on obtient ainsi le système :

$$\left\{ \begin{aligned} \left( \sum_{i \in J_1} A^i - b \right) y_{(J_1, J_2)} + \sum_{i \notin J_1 \cup J_2} A^i y_{(J_1 \cup \{i\}, J_2)} &\leq 0 & \forall (J_1, J_2) \in B^k \\ y_{(J_1, J_2)} &\geq 0 & \forall (J_1, J_2) \in B^k \\ y_{(J_1, J_2)} &= \sum_{J \subseteq J_2} (-1)^{|J|} y_{J_1 \cup J} & \forall (J_1, J_2) \in B^k \end{aligned} \right. \quad (4.14)$$

La projection de ce système d'inégalités notée  $P_{SA}^k$  dans l'espace des variables  $x$  est une relaxation du problème de départ (il est facile de le vérifier en prenant pour une solution 0/1 réalisable  $y_I = \prod_{i \in I} x_i$ ). Les différentes projections prises pour  $k = 1, 2, \dots$  ont comme propriété remarquable que :

**Théorème 4.2** [54]

$$\text{conv}(P) = P_{SA}^n \subseteq P_{SA}^{n-1} \subseteq \dots \subseteq P_{SA}^1 \subseteq P_{SA}^0 = \{x \in [0, 1]^n, Ax \leq b\}.$$

Dans la suite, nous allons nous intéresser plus particulièrement à la reformulation des relaxations d'ordre 1 à l'aide de la programmation disjonctive.

**Remarque 4.3 :** Notons que les relaxations de Sherali-Adams peuvent être renforcées par des inégalités matricielles suivant le schéma proposé dans un cadre plus général par Lovász et Schrijver [39]. On commence par ordonner complètement les sous ensembles de  $\{1, \dots, n\}$  de cardinalité inférieure ou égale à  $k + 1$ . On note  $\sigma = |\{I \in \{1, \dots, n\} : |I| \leq k + 1\}|$  le nombre de sous-ensembles de cardinalité inférieure à  $k + 1$ , de manière à avoir  $\{I \in \{1, \dots, n\} : |I| \leq k + 1\} = \{I_1, \dots, I_\sigma\}$ .

On peut alors définir le vecteur  $y \in R^\sigma$  dont la composante  $i$  est la variable  $y_{I_i}$  dans la formulation de Sherali-Adams.

On définit maintenant la matrice  $Y \in \mathbb{R}^{\sigma \times \sigma}$  où la ligne  $i$  (resp. la colonne  $j$ ) correspond au sous-ensemble  $I_i$  (resp.  $I_j$ ) de  $\{1, \dots, n\}$  et où l'élément de la matrice  $Y_{I_i, I_j} = y_{I_i \cup I_j}$ .

On a  $x \in \{0, 1\}^n$  implique  $Y$  semi-définie positive, car si  $x \in \{0, 1\}^n$  on a  $y_{I_i \cup I_j} = \prod_{i \in I_i \cup I_j} x_i = y_{I_i} y_{I_j}$  et donc  $Y = yy^t$ . On peut donc renforcer la formulation étendue en y ajoutant une contrainte portant sur la semi-définie positivité de  $Y$ . ||

## 4.5 Formulation par la programmation disjonctive de la relaxation de Sherali-Adams d'ordre 1

Dans la suite, on s'intéressera à la relaxation de Sherali-Adams d'ordre 1 que l'on notera  $P_{SA}$  pour plus de simplicité.

**Proposition 4.2** *Le polyèdre correspondant à la relaxation de Sherali-Adams d'ordre 1 est contenu dans  $P_{L\&P}$  et on a exactement :*

$$P_{SA} = P_x \left( \left( \bigcap_{i \in \{1, \dots, n\}} X^i \right) \cap \left\{ (x, y) : y_j^i = y_i^j \right\} \right)$$

(on rappelle que  $X^i$ , défini page 75, est la formulation étendue du polyèdre issu de la disjonction  $i$  et que  $P_x$  est la projection sur l'espace des variables  $x$ ).

**Preuve :**  $P_{SA}^1$  s'écrit comme le système linéaire :

$$(I) \begin{cases} (A^i - b) x_i + \sum_{j \in J: j > i} A^j y_{ij} + \sum_{j \in J: j < i} A^j y_{ji} \leq 0 & \forall i \in J \\ -b(1 - x_i) + \sum_{j \in J: j > i} A^j (x_j - y_{ij}) + \sum_{j \in J: j < i} A^j (x_j - y_{ji}) \leq 0 & \forall i \in J \\ 0 \leq y_{ij} \leq x_i \quad \forall i \in J, \quad \forall j \in J, \quad i < j \\ x_j - y_{ij} \leq 1 - x_i \quad \forall i \in J, \quad \forall j \in J, \quad i < j. \end{cases}$$

La description étendue de l'intersection des polyèdres disjonctifs est donnée par :

$$\begin{cases} Ay^i - f' y_0^i \leq 0 & \forall i \in J \\ y_j^i - y_0^i \leq 0 & \forall i \in J, \quad j \in J \\ Az^i - f' z_0^i \leq 0 & \forall i \in J \\ z_j^i - z_0^i \leq 0 & \forall i \in J, \quad j \in J \\ y_0^i + z_0^i = 1 & \forall i \in J \\ y_i^i - y_0^i = 0 & \forall i \in J \\ z_i^i = 0 & \forall i \in J \\ x - y^i - z^i = 0 & \forall i \in J \\ y^i \geq 0, z^i \geq 0, y_0^i \geq 0, z_0^i \geq 0. \end{cases}$$

En utilisant les contraintes d'égalité de ce système :

- $z_i^i = 0$  peut être supprimé.
- $y_i^i$  and  $y_0^i$  sont tous deux égaux à  $x_i$  et peuvent être remplacés.
- $z_j^i$  peut être remplacé par  $x_j - y_j^i$ .
- $z_0^i$  peut être remplacé par  $1 - x_i$ .

Comme résultat nous obtenons le système :

$$(II) \left\{ \begin{array}{l} (A^i - b) x_i + \sum_{j \in J \setminus \{i\}} A^j y_j^i \leq 0 \\ -b(1 - x_i) + \sum_{j \in J \setminus \{i\}} A^j (x_j - y_j^i) \leq 0 \\ 0 \leq y_j^i \leq x_j \quad \forall i \in J, \quad \forall j \in J \setminus \{i\} \\ 0 \leq y_j^i \leq x_i \quad \forall i \in J, \quad \forall j \in J \setminus \{i\} \\ x_j - y_{ij} \leq 1 - x_i \quad \forall i \in J, \quad \forall j \in J \setminus \{i\}. \end{array} \right.$$

qui est identique à (I) excepté pour les variables  $y_{ij}$  qui sont représentés suivant la contrainte par  $y_j^i$  ou  $y_i^j$  dans (II). Donc si on impose  $y_i^j = y_j^i$  les deux systèmes sont équivalents. ■

La relaxation d'ordre 1 peut donc être reformulée comme l'ensemble des  $x$  vérifiant :

$$\left\{ \begin{array}{l} \tilde{A}x + \tilde{B}\tilde{y} \leq \tilde{b} \\ \tilde{C}y = 0 \end{array} \right. \quad (4.15)$$

où  $\tilde{A}$ ,  $\tilde{B}$ ,  $\tilde{y}$  et  $\tilde{b}$  sont définis comme au paragraphe 4.2.3 et  $\tilde{C}y = 0$  est le système d'équations :  $y_j^i - y_i^j = 0$ , pour tout  $(i, j)$  tel que  $1 \leq i < j \leq n$ .

Cette expression de la relaxation de Sherali-Adams comporte plus de variables ( $n^2$  variables au lieu de  $\frac{n(n-1)}{2}$ ) et plus de contraintes structurelles. Par contre la structure bloc-diagonale de  $\tilde{B}$  et le caractère fortement creux de  $\tilde{C}$  permettent d'envisager des résolutions par décomposition voisines de celles employées pour optimiser sur  $P_{L\&P}$ .

## 4.6 Optimisation par décomposition

Nous allons maintenant nous intéresser à l'optimisation sur la relaxation de Sherali et Adams d'ordre 1 dans sa formulation par la programmation disjonctive par des approches de génération d'inégalités valides ; notre objectif est de nous affranchir par ces méthodes de décomposition des variables couplantes  $x$  et des contraintes couplantes  $\tilde{C}y = 0$ , pour tirer parti de la structure bloc-diagonale de  $\tilde{B}$ .

D'après les résultats du paragraphe 2.2.1, le problème de séparation sur la relaxation de Sherali-Adams d'ordre 1 s'exprime comme :

$$\left\{ \begin{array}{l} \min u(b - \tilde{A}x^*) \\ (u, v) \in \Pi \left( \begin{array}{c} \tilde{B} \\ \tilde{C} \end{array} \right) \end{array} \right. \quad (4.16)$$

où :

$$\Pi \left( \begin{array}{c} \tilde{B} \\ \tilde{C} \end{array} \right) = \left\{ (u, v) : \begin{array}{l} u^T \tilde{B} + v^T \tilde{C} \leq 0 \\ u \geq 0 \end{array} \right\},$$

et  $x^*$  est le point à séparer.

La difficulté pour séparer une inégalité coupante vient du fait que les variables  $v$  couplent tous les blocs de la matrice  $\tilde{B}$ .

Nous allons discuter ici deux manières de s'affranchir de la partie couplante. La première que nous discuterons rapidement consiste à appliquer la décomposition de Benders au problème (4.16). La seconde est une méthode de décomposition basée sur l'ajout itératif des contraintes couplantes.

#### 4.6.1 Décomposition du problème de séparation

L'optimisation avec  $v$  fixé conduit au problème :

$$\begin{cases} \min u(b - \tilde{A}x^*) \\ t.q. : \\ u\tilde{B} \geq v\tilde{C} \\ u \geq 0 \end{cases}$$

dont le dual est :

$$(P_{SA}(x^*)) \begin{cases} \max v\tilde{C}y \\ t.q. : \\ \tilde{B}y \leq u(b - \tilde{A}x^*) \\ y \geq 0 \end{cases}$$

Ce problème est non vide si et seulement si  $\bar{x} \in P_{L\&P} = \{x \in [0, 1]^n : \tilde{A}x + \tilde{B}y \leq b\}$ . Si ce problème est vide la décomposition de Benders appliquée telle quelle ne donne aucun résultat. On ne peut donc appliquer la décomposition de Benders que si  $x^* \in P_{L\&P}$ . La décomposition de Benders de (4.16) est alors donnée par le problème maître :

$$\begin{cases} \min \eta \\ t.q. : \\ v\tilde{C}y \leq \eta \quad \forall y \in \overline{P_{SA}(x^*)} \subseteq \{\text{points extrêmes de } P_{SA}(x^*)\} \end{cases}$$

(où  $\overline{P_{SA}(x^*)}$  est le sous-ensemble des points extrêmes générés à l'itération courante) et son problème esclave :

$$\begin{cases} \max \bar{v}\tilde{C}y \\ t.q. : \\ \tilde{B}y \leq \bar{b} - \tilde{A}\bar{x} \\ y \geq 0 \end{cases}$$

(où  $\bar{v}$  est la valeur de  $v$  à l'itération courante). Dans ce problème, il n'y a plus ni variable ni contrainte couplante, on peut donc le résoudre en résolvant séparément les  $n$  problèmes :

$$\begin{cases} \max (\bar{v}\tilde{C})^i y^i \\ t.q. : \\ \tilde{B}^i y^i \leq u(\bar{b}^i - \tilde{A}^i \bar{x}) \\ y^i \geq 0 \end{cases} \quad (4.17)$$

pour  $i = 1, \dots, n$  (où  $(\bar{v}\tilde{C})^i$  est le vecteur de  $\mathbb{R}^{n-1}$  dont les composantes correspondent à celles de  $y^i$ ).

Dans le paragraphe suivant nous proposons une autre méthode pour effectuer l'optimisation sur la relaxation de Sherali-Adams.

### 4.6.2 Optimisation par ajout itératif des contraintes de $\tilde{C}$

Nous allons maintenant proposer une deuxième méthode pour optimiser sur  $P_{SA}$ . Celle-ci consiste à ajouter progressivement les contraintes de  $\tilde{C}$  exprimant les conditions de symétrie  $y_j^i = y_i^j$ .

Dans une approche par décomposition, on commence par optimiser en utilisant des inégalités valides de  $P_{L\&P}$ . Lorsqu'on atteint l'optimum  $\bar{x}$  (ou que l'on a une solution qu'on estime suffisamment proche), on dispose d'une description du problème par des contraintes sur les variables  $x$  formant le problème maître de Benders :

$$\begin{cases} \max c^T x \\ u(\tilde{b} - \tilde{A}x) \leq 0 \quad \forall (u, v) \in \overline{Q} \end{cases} \quad (4.18)$$

Les variables duales de la dernière résolution des sous-problèmes de séparation associés nous donnent  $\bar{y}$ , tel que  $(\bar{x}, \bar{y})$  est optimal pour  $P_{L\&P}$ . On peut donc déterminer les contraintes de  $\tilde{C}y = 0$  non vérifiées par  $\bar{y}$ . Notons qu'en pratique il est fréquent que seulement un sous-ensemble des contraintes de  $\tilde{C}$  seulement ne soit pas vérifié (en particulier par exemple, si  $\bar{x}_i \in \{0, 1\}$  toutes les contraintes  $y_j^i = y_i^j$  sont vérifiées pour tout  $j \neq i$ ).

Le schéma que nous proposons ici consiste à ajouter progressivement les contraintes couplantes de  $\tilde{C}$  au problème maître (4.18), ce qui contraste avec le schéma de décomposition précédent où les contraintes sont gardées dans les sous-problèmes. Ceci induit que les variables  $y_j^i$  présentes dans ces nouvelles contraintes deviennent des variables couplantes et doivent aussi être ajoutées au problème maître. Cependant, étant donné le caractère fortement creux de la matrice  $\tilde{C}$ , on verra que l'ajout d'une contrainte au problème maître implique l'ajout de 4 variables seulement à l'ensemble des variables couplantes. Nous allons maintenant détailler le schéma proposé.

Notons  $\tilde{C}^1$  la matrice constituée des  $p$  lignes de  $\tilde{C}$  que l'on va ajouter au problème, et  $I^1 = \{(i_1, j_1), \dots, (i_p, j_p)\}$  (où  $i_k < j_k$ ) les indices des variables pour lesquelles on veut ajouter la contrainte  $y_{j_k}^{i_k} = y_{i_k}^{j_k}$ . Pour que les solutions données par le nouveau problème maître ne violent pas de contraintes structurelles simples du problème nous allons lui ajouter en plus de  $y_{j_k}^{i_k}$  et  $y_{i_k}^{j_k}$  les variables  $z_{j_k}^{i_k}$  et  $z_{i_k}^{j_k}$  et les contraintes structurelles :

$$\begin{cases} x_{j_k} = y_{j_k}^{i_k} + z_{j_k}^{i_k} \\ x_{i_k} = y_{i_k}^{j_k} + z_{i_k}^{j_k} \end{cases} \quad \forall (i_k, j_k) \in \tilde{C}^1$$

De manière à obtenir, juste après l'ajout des variables et contraintes mentionnées ci-dessus, comme nouveau problème maître :

$$\begin{cases} \max c^T x \\ u(\tilde{b} - \tilde{A}x) \leq 0 \quad \forall (u, v) \in \overline{Q} \\ y_{j_k}^{i_k} = y_{i_k}^{j_k} \\ x_{j_k} = y_{j_k}^{i_k} + z_{j_k}^{i_k} \\ x_{i_k} = y_{i_k}^{j_k} + z_{i_k}^{j_k} \\ y_{j_k}^{i_k} \geq 0, z_{j_k}^{i_k} \geq 0, z_{i_k}^{j_k} \geq 0 \end{cases} \quad \forall (i_k, j_k) \in \tilde{C}^1 \quad (4.19)$$

Nous allons maintenant examiner les nouveaux sous-problèmes de séparation d'inégalités. Notons  $B'$  la matrice issue de  $\tilde{B}$  par suppression des lignes et colonnes correspondant aux variables et

contraintes qui sont passées dans le problème maître. Cette matrice reste bloc diagonale et la recherche de plans coupants continue donc à se faire indépendamment sur chacune des disjonctions.

Nous allons donc expliciter le nouveau problème de séparation correspondant à une disjonction quelconque  $i$ . Soit  $I_i^1 = \{j : \exists (i_k, j_k) \in I^1 \text{ avec } i = i_k \text{ ou } i = j_k\}$  l'ensemble des indices des variables  $y_j^i$  passées dans le problème maître. Suivant les modifications effectuées, pour tout  $j \in I_i^1$  sont supprimées du sous-problème :

- la variables  $\alpha_j$  (i.e. variables duales de la contrainte  $x_j = y_j^i + z_j^i$ ).
- la variables  $u_k$  et  $v_k$  correspondant aux variables duales des contraintes de bornes sur  $y_j^i$  et  $z_j^i$ .
- les contraintes  $\alpha_j = uA^j$  et  $\alpha_j = vA^j$  (dont les variable duales sont  $y_j^i$  et  $z_j^i$ ).

Le nouveau sous-problème est ainsi donné par :

$$\left\{ \begin{array}{l} \max \sum_{j \notin I^i} \alpha_j \bar{x}_j - \sum_{j \in I^i} (uA_j^i \bar{y}_j^i + vA_j^i \bar{z}_j^i) - \beta \\ \text{t.q. :} \\ \sum_{j \notin I^i} uA^i + u_0 e_i = \alpha \\ \sum_{j \notin I^i} vA^i + v_0 e_i = \alpha \\ ub + u_0 e_i = \beta \\ vb = \beta \\ u \geq 0, v \geq 0 \end{array} \right. \quad (4.20)$$

(où  $\bar{x}$  et  $\bar{y}_j^i$  sont les valeurs courantes des variables dans le problème maître). Subséquemment, dans les nouvelles contraintes ajoutées au problème maître si  $j \in I_i^1$ , la variable  $x_j$  a un coefficient nul qui est remplacé par des coefficients sur les deux variables  $y_j^i$  et  $z_j^i$ .

## 4.7 Expérimentation du calcul de la relaxation

Nous allons présenter ici des résultats comparant le calcul de la relaxation  $P_{SA}$  par la résolution directe sur le système obtenu par reformulation-linéarisation (tel que décrit par (I)) et par l'utilisation de la technique par décomposition et ajout itératif des contraintes couplantes.

Pour valider l'idée d'ajouter itérativement les contraintes couplantes, une question importante est de savoir si toutes les contraintes couplantes sont nécessaires pour obtenir l'optimum (ou une bonne approximation) ou si un sous-ensemble seulement est suffisant. Pour tester cela, nous avons optimisé sur la relaxation de Sherali-Adams en ajoutant une par une les contraintes couplantes pour calculer le renforcement obtenu après l'ajout d'une contrainte, de deux contraintes, etc . . .

Le graphique 4.10 donne pour huit problèmes de la MIPLIB les résultats obtenus, en indiquant la part de l'écart entre l'optimum de  $P_{L\&P}$  et l'optimum de la relaxation de Sherali-Adams comblée en fonction du pourcentage des contraintes ajoutées par rapport au nombre total de contraintes couplantes.

Pour tous les problèmes traités, l'optimum est atteint avec moins de 50% des contraintes couplantes ajoutées ; avec 20% des contraintes ajoutées on a comblé plus de 75% de l'écart et dans 5 problèmes sur 8 on est déjà à l'optimum. Ces résultats semblent indiquer qu'une approche où ne sont pas exprimées toutes les contraintes couplantes est pertinente.

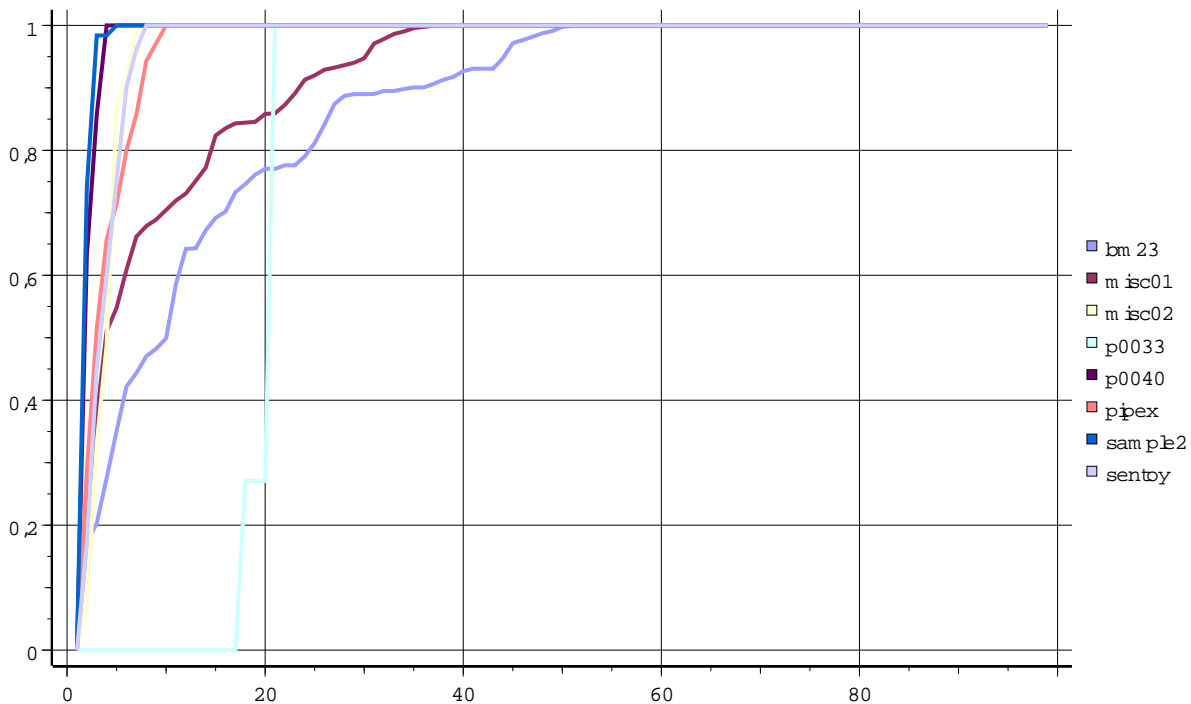


FIG. 4.10 – Renforcement en fonction du nombre de contraintes couplantes ajoutées

Le tableau 4.11 relate des expériences faites sur des problèmes de la MIPLIB dans le but de comparer notre approche développée dans le paragraphe 4.6.2 avec la résolution brute du programme linéaire donné par la formulation complète de la relaxation de Sherali-Adams. Ces expériences sont préliminaires. De nombreux choix sont possibles dans l'implémentation de l'approche par décomposition comme par exemple la fréquence d'ajout des contraintes couplantes au problème maître, le choix des contraintes couplantes ajoutées, le choix des sous-problèmes à optimiser. Nous n'avons pas, à ce jour, effectué suffisamment d'expériences pour valider de bons réglages dans l'implémentation de la méthode ; il reste donc, à notre avis, des améliorations possibles dans les résultats que nous proposons ici.

Dans le tableau 4.11, on donne les renforcements de la relaxation continue obtenus avec la méthode exacte et la méthode par décomposition en ajoutant 10% des contraintes couplantes ainsi que les temps de calcul pris par chacune des deux méthodes. Nous avons limité le temps de calcul à 1 heure. Pour les plus gros problèmes, l'écriture de la matrice de Sherali-Adams demande un espace mémoire trop important (plus de 200Mo). Enfin, certaines expériences n'ont pas encore pu être réalisées.

La comparaison des deux méthodes semble indiquer que l'approche par décomposition est moins bonne que la résolution avec l'approche par programmation linéaire sur les petits problèmes (jusqu'à environ 50 variables dans la formulation originale soit environ 1300 variables dans la formulation de Sherali-Adams). Dans les problèmes plus gros, l'approche par décomposition dans des temps de calculs inférieurs donne des renforcements très proches de ceux obtenus par la relaxation de Sherali-Adams. Les temps de calculs restent cependant élevés.

Un autre aspect intéressant de ce tableau est la comparaison des renforcements obtenus avec ceux obtenus au chapitre précédent avec le lift and project et  $P_{L\&P}$ . Dans une grande majorité des cas la relaxation de Sherali-Adams apporte peu au renforcement obtenu avec  $P_{L\&P}$ . Par contre sur un petit

Instance	Renforcement Sherali Addams par PL	Renforcement par décomposition et ajout itératif	Temps Sherali Addams par PL	Temps décomposition et ajout itératif
bm23	21.33%	19.82%	11.39	20.73
misc01	54.76%	27.56%	>3600	197.85
misc02	100.00%	98.94%	99.66	181.67
p0033	8.19%	8.62%	14.01	5.05
p0040	28.16%	28.16%	6.88	7.00
pipex	4.97%	5.01%	22.51	29.92
sample2	36.66%	34.83%	8.82	2.93
sentoy	28.49%	27.48%	2461.75	210.74
lseu	18.08%	17.04%	899.84	353.27
misc03	11.21%	69.41%	>3600	899.31
rgn	-146.11%	---	>3600	---
mod008	13.64%	---	>3600	---
modglob	Mémoire	---	+++	---
misc07	Mémoire	---	+++	---
p0282	0.00%	---	>3600	---
pp08a	Mémoire	---	+++	---
vpm1	36.28%	---	11.59	---

FIG. 4.11 – Comparaisons calculs de Sherali-Adams partiels par PL et par décomposition

nombre (misc01, misc02, misc03) elle permet d'obtenir un renforcement conséquent (allant même jusqu'à trouver l'optimum exact pour misc02 avec moins de 20% des contraintes couplantes). Il paraîtrait donc intéressant de pouvoir déterminer sous quelles conditions la relaxation de l'ajout des contraintes de symétrie permet de renforcer la relaxation  $P_{L\&P}$ . Au cours de nos expériences une classe de problèmes pour laquelle les deux relaxations nous ont donné des résultats pratiques qui semblent systématiquement identiques est la classe des problèmes MAX-2SAT. Nous avons cherché à montrer de manière théorique cette équivalence. Nos travaux ne nous ont pas encore permis d'aboutir à ce résultat. Nous avons cependant pu déterminer l'équivalence pour le problème d'optimisation d'une fonction pseudo-booléenne quadratique qui permet de modéliser d'une autre manière les problèmes de MAX2-SAT. Nous relatons cette recherche dans le prochain paragraphe.

## 4.8 Équivalence des relaxations $P_{L\&P}$ et $P_{SA}$ pour les problèmes de maximisation de fonction pseudo-booléennes quadratique non contraints

Etant donnée  $C$  une matrice  $n \times n$  diagonale supérieure, on considère le problème d'optimisation quadratique suivant :

$$\max \{x^T C x : x \in \{0, 1\}^n\} \quad (4.21)$$

Comme lorsque  $x \in \{0, 1\}^n$  on a  $x_i^2 = x_i$  ; en notant  $c_i$  l'élément diagonal  $c_{ii}$ , le problème (4.21) s'exprime de manière équivalente comme :

$$\max \left\{ \sum_{i=1}^n c_i x_i + \sum_{\substack{1 \leq i \leq n \\ i < j \leq n}} c_{ij} x_i x_j : x \in \{0, 1\}^n \right\} \quad (4.22)$$

Classiquement ce problème se formule comme un problème linéaire 0/1 au travers de la linéarisation des produits  $x_i x_j$  grâce à l'introduction de variables  $w_{ij}$  continues tels que :

$$x_i + x_j - w_{ij} \leq 1, \quad (4.23a)$$

$$-x_i + w_{ij} \leq 0, \quad (4.23b)$$

$$-x_j + w_{ij} \leq 0, \quad (4.23c)$$

$$-w_{ij} \leq 0. \quad (4.23d)$$

(4.23b), (4.23c) On obtient alors le programme linéaire mixte :

$$\left\{ \begin{array}{l} \max \sum_{i=1}^n c_i x_i + \sum_{\substack{1 \leq i \leq n \\ i < j \leq n}} c_{ij} w_{ij} \\ t.q. : \\ x_i + x_j - w_{ij} \leq 1 \quad \forall 1 \leq i < j \leq n, \\ -x_i + w_{ij} \leq 0 \quad \forall 1 \leq i < j \leq n \\ -x_j + w_{ij} \leq 0 \quad \forall 1 \leq i < j \leq n \\ x \in \{0, 1\}^n, w \in [0, 1]^{\frac{n(n-1)}{2}} \end{array} \right. \quad (4.24)$$

On va s'intéresser ici aux relaxations de ce problème obtenues à l'aide de la fermeture élémentaire des coupes de Lift & Project ( $P_{L\&P}$ ) et de la Reformulation-Linéarisation par Sherali-Adams ( $P_{SA}$ ). Dans un premier temps on établira l'équivalence entre les deux relaxations ; on s'intéressera ensuite aux implications pour les problèmes de Max2SAT et de stable.

**Théorème 4.3** *La relaxation  $P_{L\&P}$  de (4.24) est la même que la formulation  $P_{SA}$ .*

**Preuve :** Pour montrer l'équivalence, nous allons commencer par formuler la relaxation  $P_{L\&P}$  de (4.24), nous montrerons ensuite que les contraintes supplémentaires nécessaires à la formulation de  $P_{SA}$  sont implicitement dans  $P_{L\&P}$ .

Commençons par introduire les variables supplémentaires nécessaires à la formulation explicite de  $P_{L\&P}$ . On va ajouter au problème une variable  $y_l^j$  pour la linéarisation du produit  $x_l x_j$  (dans cet ordre) et une variable  $w_{ij}^l$  pour linéariser le produit  $x_l w_{ij}$ . Ces variables sont soumises aux contraintes structurelles :

$$y_j^l \leq \min(x_i, x_j) \quad l \neq j = 1, \dots, n \quad (4.25)$$

$$y_j^l \geq \max(0, x_i + x_j - 1) \quad l \neq j = 1, \dots, n \quad (4.26)$$

$$w_{ij}^l \leq \min(x_l, w_{ij}) \quad l = 1, \dots, n, i = 1, \dots, n \text{ et } j = i + 1, \dots, n \quad (4.27)$$

$$w_{ij}^l \geq \max(0, x_l + w_{ij} - 1) \quad l = 1, \dots, n, i = 1, \dots, n \text{ et } j = i + 1, \dots, n \quad (4.28)$$

Les contraintes non structurelles du problèmes s'obtiennent en multipliant à gauche les contraintes de (QP) respectivement par  $x^l$  et  $(1 - x_l)$  puis en remplaçant  $x_l^2$  par  $x_l$  et les produits  $x_l x_j$  et  $x_l w_{ij}$  respectivement par les variables  $y_j^l$  et  $w_{ij}^l$ .

Nous allons maintenant établir contrainte par contrainte la reformulation par  $P_{L\&P}$  dans cet espace.

En multipliant la contrainte  $x_i + x_j - w_{ij} \leq 1$  respectivement par  $x_l$  (avec  $l \neq i$  et  $l \neq j$ ) par  $x_i$  et par  $x_j$  on obtient :

$$\begin{aligned} y_i^l + y_j^l - w_{ij}^l - x_l &\leq 0 \\ y_j^i - w_{ij}^i &\leq 0 \\ y_i^j - w_{ij}^j &\leq 0 \end{aligned}$$

En la multipliant respectivement par  $1 - x_l$ ,  $1 - x_i$  et  $1 - x_j$  on obtient :

$$\begin{aligned} x_i + x_j + x_l - y_i^l - y_j^l - w_{ij} + w_{ij}^l &\leq 1 \\ x_i + x_j - y_j^i - w_{ij} + w_{ij}^i &\leq 1 \\ x_i + x_j - y_i^j - w_{ij} + w_{ij}^j &\leq 1 \end{aligned}$$

Remarquons que les deux dernières contraintes sont redondantes, comme elles sont la somme de  $x_i + x_j - y_j^i \leq 1$  (respectivement  $x_i + x_j - y_i^j \leq 1$ ) et de  $w_{ij}^i \leq w_{ij}$  (respectivement  $w_{ij}^j \leq w_{ij}$ ) et peuvent donc être supprimées.

En multipliant à gauche la contrainte  $-x_i + w_{ij} \leq 0$  respectivement par  $x_l$ ,  $x_i$  et  $x_j$ , on obtient :

$$\begin{aligned} -y_i^l + w_{ij}^l &\leq 0 \\ -x_i + w_{ij}^i &\leq 0 \\ -y_i^j + w_{ij}^j &\leq 0 \end{aligned}$$

La deuxième de ces contraintes, faisant parti des contraintes structurelles, peut être supprimée. Par multiplication par  $1 - x_l$ , on obtient :

$$\begin{aligned} -x_i + y_i^l + w_{ij} - w_{ij}^l &\leq 0 \\ w_{ij} - w_{ij}^i &\leq 0 \\ -x_i + y_i^j + w_{ij} - w_{ij}^j &\leq 0 \end{aligned}$$

Pour la contrainte  $-x_j + w_{ij} \leq 0$ , on obtient des résultats symétriques.

$P_{L\&P}$  est donc la projection dans l'espace des  $x$  du système de contraintes suivant :

$$y_i^l + y_j^l - w_{ij}^l - x_l \leq 0 \quad \forall l \neq i, l \neq j \quad (4.29a)$$

$$x_i + x_j + x_l - y_i^l - y_j^l - w_{ij} + w_{ij}^l \leq 1 \quad \forall l \neq i, l \neq j \quad (4.29b)$$

$$-x_i + y_i^l + w_{ij} - w_{ij}^l \leq 0 \quad \forall l \neq i, l \neq j \quad (4.29c)$$

$$-y_j^l + w_{ij}^l \leq 0 \quad \forall l \neq i, l \neq j \quad (4.29d)$$

$$-x_j + y_j^l + w_{ij} - w_{ij}^l \leq 0 \quad \forall l \neq i, l \neq j \quad (4.29e)$$

$$-x_{l'} + y_{l'}^l + w_{ij} - w_{ij}^j \leq 0 \quad \forall \{l, l'\} = \{i, j\} \quad (4.29f)$$

$$y_{l'}^l - w_{ij}^l \leq 0 \quad \forall \{l, l'\} = \{i, j\} \quad (4.29g)$$

$$-y_{l'}^j + w_{ij}^j \leq 0 \quad \forall \{l, l'\} = \{i, j\} \quad (4.29h)$$

$$w_{ij} - w_{ij}^l \leq 0 \quad \forall \{l, l'\} = \{i, j\} \quad (4.29i)$$

Par application de la proposition 4.2, il suffit maintenant de démontrer que les contraintes  $y_j^i = y_i^j$  sont induites par les contraintes définissant  $P_{L\&P}$ . La contrainte (4.29i) combinée avec la contrainte structurelle  $-w_{ij} + w_{ij}^l \leq 0$  donne l'égalité entre  $w_{ij}$ ,  $w_{ij}^j$  et  $w_{ij}^i$ .

Les deux contraintes (4.29g) et (4.29h) combinées établissent l'égalité entre  $y_j^i$  et  $w_{ij}^i$  d'une part,  $y_i^j$  et  $w_{ij}^j$  d'autre part.

En cumulant tout, on a :

$$y_j^i = w_{ij}^i = w_{ij} = w_{ij}^j = y_i^j$$

■

Les problème de MAX2SAT peuvent se formuler comme la maximisation d'une fonction pseudo-quadratique :

$$\max \left\{ \max - \sum_{(i,j) \in C_1} x_i x_j - \sum_{(i,j) \in C_2} \bar{x}_i x_j - \sum_{(i,j) \in C_3} \bar{x}_i \bar{x}_j : x \in \{0, 1\}^n \right\} \quad (4.30)$$

(où  $C_1$  est l'ensemble des couples  $(i, j)$  sur lesquels porte une clause du type  $\bar{x}_i \vee \bar{x}_j$ ,  $C_2$  celui des couples sur lesquels porte une clause  $x_i \vee \bar{x}_j$ ,  $C_3$  celui des couples sur lesquels portent une clause du type  $x_i \vee x_j$ ). On peut développer les termes du polynôme pour obtenir un problème quadratique sous la forme de 4.24 et effectuer la linéarisation pour obtenir un programme linéaire en nombres entiers que l'on note (QP-MAX2SAT). Par application directe du théorème 4.3 on peut alors établir le corollaire suivant :

**Corollaire 4.1** *Les relaxations  $P_{L\&P}$  et  $P_{SA}$  de (QP-MAX2SAT) sont identiques*

On voudrait maintenant établir l'équivalence entre les relaxations  $P_{SA}$  de (QP-MAX2SAT) d'une part, et de (MAX2SAT) d'autre part. On peut voir assez facilement que les relaxations linéaires des deux formulations sont identiques mais que par contre les polyèdres ne sont pas identiques à cause des contraintes  $w_{ij} \leq x_i$  et  $w_{ij} \leq x_j$  présentes dans (QP-MAX2SAT) et qui n'ont pas d'équivalent dans (MAX2SAT) (ces contraintes ne sont jamais actives à l'optimum continu comme  $w_{ij}$  doit être minimisé et  $x_i$  et  $x_j$  ont un coût nul). Néanmoins leur présence implique que les polyèdres des deux relaxations continues sont différents et que donc les formulations de Sherali-Adams sont différentes. Nos expériences semblent indiquer néanmoins que pour cette fonction de coûts les résultats sont équivalents.

Notons aussi que nos expériences indiquent que l'équivalence éventuelle ne se situe pas dans l'espace de grande dimension mais seulement après projection sur l'espace des variables originales du problème.



## Chapitre 5

# Génération d'inégalités coupantes conditionnellement valides

Dans les chapitres précédents nous avons étudié plusieurs méthodes de génération d'inégalités coupantes valides. Une autre approche, moins connue, pour la génération de plans coupants est la génération d'inégalités *conditionnellement valides* pour le polyèdre des solutions entières. Ces inégalités ne sont valides que pour un sous-ensemble des solutions réalisables du problème. Leur utilisation rend donc nécessaire l'énumération (explicite ou implicite) des solutions éventuellement éliminées pour déterminer l'optimum du problème.

Dans les années 70, utilisant la notion de coupes d'intersection de Balas[2], Balas [3], Burdet [13] et Glover [28] ont proposé des méthodes de séparation d'inégalités non-valides appelées coupes énumératives. A notre connaissance, peu d'expériences ont été effectuées à partir de ces idées. En 1972, dans [13], Burdet rapporte seulement quelques expériences sur une dizaine de petits problèmes (moins de 10 variables entières et 10 contraintes). Ces idées ont été reprises récemment par Balas *et al.*[9], mais en vue de concevoir une méthode de recherche heuristique uniquement.

Récemment aussi, Osman Oguz [46] a proposé un algorithme de résolution des programmes 0/1 basé sur l'utilisation d'inégalités non valides. Nediak et Eckstein [43] ainsi que Fischetti et Lodi [25] utilisent aussi une approche basée sur des inégalités conditionnellement valides pour la résolution heuristique.

Nous proposons dans ce chapitre de reconsidérer l'utilisation d'inégalités conditionnellement valides *en vue de la résolution exacte*, particulièrement en étudiant divers schémas algorithmiques nouveaux pour utiliser de telles coupes dans le cours de la résolution par un solveur de programmation entière.

Dans un premier temps, nous allons brièvement rappeler les approches des années 70 pour la dérivation de coupes d'énumération, et les approches plus récentes. Nous proposerons ensuite plusieurs schémas algorithmiques nouveaux d'utilisation des coupes non-valides et nous discuterons les résultats expérimentaux obtenus avec ces algorithmes.

### 5.1 Coupes énumératives

Le premier type d'inégalités non valides que nous allons étudier est celui des coupes énumératives. Ces coupes sont utilisables pour la résolution de programmes en nombres entiers généraux. Nous

allons d'abord rappeler le principe des coupes d'intersection qui sont à la base de la construction des coupes énumératives, puis décrire plusieurs types de coupes énumératives.

### 5.1.1 Coupes d'intersections

Soit  $x^*$  la solution optimale de la relaxation continue d'un problème d'optimisation en nombres entiers :

$$\left\{ \begin{array}{l} \max c^T x \\ t.q. \\ Ax \leq b \\ x \in \mathbb{Z}^n \end{array} \right.$$

Reformulons le problème en fonction des variables de base à l'optimum continu ; tout d'abord en ajoutant les variables d'écart  $(x_{n+1}, \dots, x_{n+m})$  ; soient  $B$  les indices des variables en bases,  $N$  les indices des variables hors base et  $\bar{A}$  la matrice du tableau optimal ; les contraintes du problème sont reformulées de la façon suivante :

$$x_i = x_i^* - \sum_{j \in N} \bar{a}_{ij} x_j \quad \forall i \in B.$$

Les  $n$  rayons définis par les contraintes saturées par  $x^*$  sont :

$$C_j = \{x' : x' = x^* - \bar{a}_j x_j, \quad x_j \geq 0\}.$$

Ces rayons définissent un cône  $C$  contenant l'ensemble des solutions réalisables. Les coupes d'intersection viennent du résultat suivant :

**Théorème 5.1 ([2])** *Soit  $S$  un ensemble convexe contenant strictement  $x^*$  mais ne contenant aucun point entier. L'hyperplan  $\{x' : \alpha^T x' = \beta\}$  défini par les  $n$  points affinement indépendants donnés par l'intersection de chaque  $C_j$  avec la frontière de  $S$  définit une inégalité  $\alpha^T x \geq \beta$  valide pour  $X$ .*

### 5.1.2 Dérivation des coupes énumératives

La différence entre les coupes d'intersection et les coupes énumératives est que pour les coupes énumératives l'ensemble convexe  $S$  peut contenir des points extrêmes entiers. Dans ce dernier cas on a le théorème suivant :

**Théorème 5.2 ([13] [28])** *Soit  $S$  un ensemble convexe contenant strictement  $x^*$ . L'hyperplan  $\{x' : \alpha^T x' = \beta\}$  contenant les  $n$  points affinement indépendants donnés par l'intersection de chaque  $C_j$  avec la frontière de  $S$  définit une inégalité  $\alpha^T x \leq \beta$  valide pour  $X \setminus S$ .*

Pour séparer des inégalités, différentes approches ont été proposées suivant le choix de  $S$ .

Ce choix de  $S$  est guidé par deux critères contradictoires :

- Avoir  $S$  le plus grand possible en terme de volume (afin de rendre l'inégalité plus coupante).
- Faire que  $S$  ait le moins possible de points entiers réalisables et que leur énumération soit simple (afin de réduire l'énumération).

Un dernier critère est enfin la simplicité de calcul des coefficients de l'inégalité.

Le premier choix de  $S$ , pour des inégalités valides, a été de prendre la boule circonscrite au cube unitaire :  $K = \{x \in R^n : \lfloor x_i^* \rfloor \leq x_i \leq \lceil x_i^* \rceil\}$ .

Dans [3], Balas étudie des ensembles convexes circonscrits à  $K$  mais de plus grand volume par le biais de l'utilisation des polaires extérieures.

### 5.1.3 Les coupes énumératives de Burdet

Dans [13], Burdet ajoute plusieurs améliorations aux coupes d'intersections pour dériver les coupes énumératives.

La dérivation des coupes d'intersections s'appuie sur plusieurs phénomènes locaux : le fait que l'ensemble des solutions réalisables soit contenu dans le cône  $C$  (qui implique qu'on puisse séparer sur  $C$ ), l'optimalité de  $x^*$ , l'intégrité des variables en base (qui implique que l'intérieur de  $K$  est vide). Burdet prend de plus en compte l'intégrité des variables hors base, et le fait qu'on peut remplacer  $K$  par un ensemble de plus grande taille à condition d'énumérer les solutions éventuelles contenues dans celui-ci.

#### Prise en compte de l'intégrité des variables hors base

La prise en compte de l'intégrité des variables hors base permet de renforcer les coupes générées. Ceci s'effectue en modifiant l'expression des rayons  $C_j$  de manière à prendre en compte uniquement les parties fractionnaires des coefficients des contraintes définissant les variables en base. Définissons :

$$g_i = x_i^* - \lfloor x_i^* \rfloor$$

la partie fractionnaire de la variable en base  $i$  et

$$f_{ij} = \begin{cases} \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor, & \text{si } \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor \leq g_i \\ \bar{a}_{ij} - \lceil \bar{a}_{ij} \rceil, & \text{si } \bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor \geq g_i \end{cases} \quad i \in B \quad j \in N$$

la partie fractionnaire des coefficients de la matrice.

Remarquons ici que par construction on a :  $\lfloor x_i^* \rfloor \leq x_i^* - f_{ij} \leq \lceil x_i^* \rceil$ . Définissons les rayons :

$$\tilde{C}_j = \{y' : y' = g - \tilde{\lambda}_j f_j, \tilde{\lambda}_j \geq 0\}$$

et  $\tilde{C}$  le cône généré par ces rayons. On a :

**Lemme 5.1** *Les ensembles de solutions entières des cônes  $C_j$  et  $\tilde{C}_j$  sont équivalents dans le sens où si  $\lambda_j = \lambda_j$  est entier pour tout  $j$  alors la solution  $x'$  définie par  $\lambda$  dans  $C$  est entière si et seulement si la solution  $y'$  définie par  $\tilde{\lambda}$  dans  $\tilde{C}$  est entière.*

**Preuve :** Il suffit de calculer  $x' - y'$ . ■

**Proposition 5.1** *Si  $S$  est un ensemble convexe contenant  $g$  dans son intérieur et si  $F = (f_{ij})$  est de pleine dimension alors la coupe déterminée par l'intersection des  $\tilde{C}_j$  avec  $S$  est valide à condition d'énumérer les points entiers de  $S$ .*

Pour la preuve de cette proposition se reporter à [13].

La génération par l'intermédiaire de  $\tilde{C}$  a deux avantages. Premièrement, les coupes sont en générales plus profondes (du fait que  $\tilde{C}_j \leq C_j$ , pour un  $S$  équivalent les valeurs obtenues pour l'intersection sont plus grandes avec  $\tilde{C}$ ). Deuxièmement, les coupes générées requièrent au plus le même effort d'énumération que les coupes générées par l'intermédiaire de  $C$  (car on peut prendre des ensembles  $S$  de même taille). Pour des exemples illustrant ce point, on se reportera à [13].

### Choix d'ensembles $S$ pouvant contenir des points entiers réalisables

La base des ensembles  $S$  choisis par Burdet est constituée par les parallélotopes définis par :

$$U(x^*, \Delta^+, \Delta^-) = \{x : -\Delta_i^- \leq x_i - \tilde{x}_i \leq \Delta_i^+\}$$

où  $\tilde{x}_i = x_i^* - g_i$ ,  $g_i$  étant la partie entière supérieure ou inférieure (au choix) de  $x^*$ .  $\Delta_i^+$  et  $\Delta_i^-$  sont des entiers positifs. Deux de ces parallélotopes utilisés par Burdet sont :  $U_0(x^*) = U(x^*, 0, 1)$ , où  $g_i = x_i^* - \lfloor x_i^* \rfloor$  ( $U_0(x^*)$  est le cube unitaire contenant  $x^*$  et ne contenant aucun point entier dans son intérieur).

Et  $U_1(x^*) = U(x^*, \Delta^-, 1 - \Delta^-)$ , où  $\tilde{x}_i$  est la valeur arrondie de  $x_i^*$  et  $\Delta^-$  est le vecteur 0/1 tel que  $x^* \in U_1(x^*)$ . Il est notable que le volume de  $U_1$  est  $2^n$  mais  $U_1$  ne contient qu'un seul point entier ( $x$ ); alors que le volume de  $U_0$  est 1 et qu' $U_0$  ne contient aucun point entier.

Burdet propose d'étendre encore les parallélotopes en construisant des polyèdres qu'il appelle diamants. Pour un entier  $k$  donné en un parallélotope  $U(x^*, \Delta^+, \Delta^-)$ , le diamant  $D(x^*, k, \Delta^+, \Delta^-)$  est l'enveloppe convexe des points extrêmes de  $U$  et des  $2n$  points :

$$x_j^{i+} = \begin{cases} x_j^* & \text{si } j \neq i \\ x_j^* + k(\Delta_i^+ - g_i) & \text{si } j = i \end{cases}$$

$$x_j^{i-} = \begin{cases} x_j^* & \text{si } j \neq i \\ x_j^* - k(\Delta_i^- + g_i) & \text{si } j = i \end{cases}$$

L'intérêt des diamants est que les coefficients des coupes sont simples à calculer et que les coupes générés dominent les coupes de Gomory dans certains cas.

#### 5.1.4 Les coupes énumératives de Glover

L'approche utilisée par Glover dans [28] est sensiblement différente. Si on choisit des multiplicateurs  $t_j^* > 0$  définis pour tout  $j$  hors base (*i.e.*  $j \in N$ ), la coupe d'intersection définie par ces multiplicateurs est :

$$\sum_{j \in N} \frac{1}{t_j^*} x_j' \geq 1. \quad (5.1)$$

La partie du cône  $C$  retirée par cette coupe est donnée par :

$$C^* = \{x' : x' = x^* - \sum_{j \in N} \bar{a}_j x_j, \quad x_j < t_j^*\} \quad (5.2)$$

La méthode de génération de coupes est basée sur le lemme suivant :

**Lemme 5.2** *Si le point  $x'$  appartient à  $C^*$ , alors tout hyperplan passant par  $x'$  intersecte au moins une des arêtes de  $C^*$  (données par  $\{x' : x' - x^* - \bar{a}_j x_j, \quad x_j < t_j^*\}$ ).*

En particulier, si  $x'$  est un point entier de  $C^*$ , les hyperplans  $x_i = x_i'$  intersectent au moins une des arêtes de  $C^*$ . Et inversement, pour  $t^*$  donné, si pour chaque  $i \in \{1, \dots, n\}$  on détermine parmi les hyperplans  $x_i = k$  (pour tout  $k \in \mathbb{Z}$ ) ceux intersectés par au moins une des arêtes  $C_j^*$ . On peut déterminer un ensemble :

$$S^* = \{x \in \mathbb{R}^{m+n} : L_i \leq x_i \leq U_i \text{ et } x_i \in \mathbb{Z}, \forall i = 1, \dots, n\}$$

tel que  $C^* \subset S^* \cap C$  (remarquons que sauf cas particulier ces deux ensembles ne sont pas égaux car si, pour  $i = 1, \dots, n$  il existe  $k_i$  tel que  $x_i = k_i$  intersecte une des arêtes de  $C^*$ , le point  $(k_1, \dots, k_n)$  n'est pas forcément dans  $C^*$ ).

De ceci, Glover peut dériver une méthode énumérative de génération de coupes. A chaque étape de la méthode, on va augmenter la valeur de l'un des  $t_j^*$  pour augmenter la taille de  $S^*$ . Pour cela, on détermine pour tout  $j$  la valeur  $t_j^+ > t_j^*$  qui est la plus petite telle que si l'on prend  $t_j^* = t_j^+$  l'ensemble  $S^*$  s'en trouve augmenté.

Une itération de l'algorithme consiste à remplacer  $t_j^*$  par  $t_j^+$  pour  $\hat{j} = \arg \max(t_j^+ - t_j^*)$ , puis à remettre à jour  $t_j^+$ . Un critère d'arrêt doit être choisi (comme par exemple la taille de  $S^*$  ou la profondeur de la coupe). Lorsque l'algorithme s'arrête il faut énumérer  $C^*$ , on peut ensuite générer la coupe déterminée par les  $t_j^+$ .

## 5.2 Coupes géométriques

Nous allons décrire une autre classe d'inégalités conditionnellement valides basées sur la proximité des points  $\{0, 1\}$  en distance de Hamming à une face de l'hypercube  $[0, 1]^n$  (pour un exposé plus complet sur ce type d'inégalités on pourra se reporter à Maculan Macambira et De Souza[40]).

### 5.2.1 Coupes de proximité à un point extrême de l'hypercube

Soit  $\bar{x} \in \{0, 1\}^n$ ; l'inégalité :

$$\sum_{\{i:\bar{x}_i=0\}} x_i - \sum_{\{i:\bar{x}_i=1\}} x_i \leq r - |\{i:\bar{x}_i=1\}| \quad (5.3)$$

est satisfaite pour tout  $x \in \{0, 1\}^n$  tel que  $\|x - \bar{x}\|_1 \leq r$  et non satisfaite pour tout  $x \in \{0, 1\}^n$  tel que  $\|x - \bar{x}\|_H > r$ . En effet si  $\|x - \bar{x}\|_1 = r'$  alors notons  $\mathcal{U} = \{i : \bar{x}_i = 1, x_i = 0\}$  et  $\mathcal{L} = \{i : \bar{x}_i = 0, x_i = 1\}$ . On a  $|\mathcal{U}| + |\mathcal{L}| = r'$ ,  $\sum_{\{i:\bar{x}_i=1\}} x_i = |\{i:\bar{x}_i=1\}| - |\mathcal{U}|$  et  $\sum_{\{i:\bar{x}_i=0\}} x_i = |\mathcal{L}|$

donc :

$$\begin{aligned} \sum_{\{i:\bar{x}_i=0\}} x_i - \sum_{\{i:\bar{x}_i=1\}} x_i &= |\mathcal{U}| + |\mathcal{L}| - |\{i:\bar{x}_i=1\}| \\ &= r' - |\{i:\bar{x}_i=1\}| \end{aligned}$$

Clairement, suivant que  $r'$  est plus petit ou plus grand que  $r$ , l'inégalité est satisfaite ou non.

Inversement, on peut exprimer une inégalité donnant l'éloignement à un point de l'hypercube :

$$\sum_{\{i:\bar{x}_i=0\}} x_i - \sum_{\{i:\bar{x}_i=1\}} x_i > r - |\{i:\bar{x}_i=1\}| \quad (5.4)$$

satisfaite si et seulement si  $x \in \{0, 1\}^n$  est tel que  $\|x - \bar{x}\|_1 > r$  (cette inégalité peut être vue comme exprimant la proximité au point diamétralement opposé à  $\bar{x}$ ).

Nous allons maintenant étudier l'utilisation de coupes d'éloignement à un point pour couper un point extrême  $x^*$  de la relaxation continue d'un problème combinatoire. Pour cela il faut trouver  $\bar{x}$  et  $r$  tels que  $\|\bar{x} - x^*\|_1 \leq r$ . Une telle coupe n'est en général pas valide ; on va donc devoir explorer l'ensemble des solutions réalisables éliminées par une telle coupe. Pour la séparation, on doit donc prendre en compte à la fois la "profondeur" de l'inégalité :  $\delta = \|\bar{x} - x^*\|_1 - r$  et la taille de l'ensemble à explorer proportionnelle à  $r$  ( $2^r$ ). Si l'on veut avoir un rapport avantageux il est facile de voir qu'il faut choisir :

$$\bar{x}_i \begin{cases} 0 & \text{si } x_i^* \leq 0,5 \\ 1 & \text{si } x_i^* > 0,5 \end{cases}$$

et,

$$r \geq \lceil \|\bar{x} - x^*\|_1 \rceil.$$

L'inégalité décrivant l'ensemble des  $x$  tels que  $\|\bar{x} - x\|_1 > r$  (ou  $\|\bar{x} - x\|_1 \geq r + 1$ ) est alors clairement coupante. On peut ajouter cette inégalité au problème à condition d'explorer l'ensemble coupé :  $\{x \in \{0, 1\}^n : \|x - \bar{x}\|_1 \leq r\}$ .

## 5.2.2 Coupes de proximité à une face de l'hypercube

Ces coupes étant une généralisation triviale des précédentes, nous resterons très bref dans leur exposé. Soit  $Y$  une face de l'hypercube, cette face est caractérisée par :

$$Y = \{x \in \{0, 1\}^n : x_i = 1 \ i \in U, x_i = 0 \ i \in L\}$$

avec  $U$  et  $L$  tels que  $U \cap L = \emptyset$  et  $U \cup L \subset \{1, \dots, n\}$ .

On a  $\|x - Y\|_1 \leq r$  si et seulement si :

$$\sum_{i \in L} x_i - \sum_{i \in U} x_i \leq r - |U| \quad (5.5)$$

et  $\|x - Y\|_1 \geq r$  si et seulement si :

$$\sum_{i \in L} x_i - \sum_{i \in U} x_i \geq r - |U| \quad (5.6)$$

Si on veut couper un point  $x^*$ , de manière à avoir un rapport le meilleur possible entre la profondeur de la coupe et la taille de l'ensemble à explorer, il faut prendre  $U = \{i : x_i^* \geq 1 - \epsilon_1\}$ , et  $L = \{i : x_i^* \leq \epsilon_2\}$  (les valeurs  $\epsilon_1$  et  $\epsilon_2$  étant à déterminer),  $r \geq \lceil \|x^* - Y(U, L)\|_1 \rceil$  et utiliser l'inégalité coupante exprimant  $\|x - Y(U, L)\|_1 \geq r$ .

Pour déterminer une inégalité coupante, on utilisera l'algorithme 4 ci-dessous qui construit  $Y(U, L)$  tel que  $\|x - Y(U, L)\|_1 \geq r + 1$  soit une inégalité coupante (pour  $r$  entier positif donné).

## 5.3 Utilisation des coupes de proximité dans la littérature

### 5.3.1 Algorithme heuristique([43])

Ici, les coupes sont utilisées dans le cadre d'un algorithme de recherche d'une bonne solution entière par "arrondi" d'une solution fractionnaire. Les coupes sont utilisées pour restreindre l'espace sur

---

**Algorithme 4** Construire  $Y(U, L)$  tel que  $\|x^* - Y(U, L)\| \leq r$

---

**Nécessite:**  $r$  est un entier positif ou nul

$U \leftarrow \emptyset$

$L \leftarrow \emptyset$

**boucle**

Soit  $\hat{i}$  tel que  $x_i^* = \min_i(\min(x_i^*, 1 - x_i^*))$

**si**  $x_i^* < 0.5$  **alors**

**si**  $\sum_{i \in L} x_i^* + x_i^* - \sum_{i \in U} x_i^* > r - |U|$  **alors**

$L \leftarrow \hat{i}$

**sinon**

*fin*

**fin si**

**sinon**

**si**  $\sum_{i \in U} x_i^* - \sum_{i \in L} x_i^* - x_i^* > r - |U| - 1$  **alors**

$U \leftarrow \hat{i}$

**sinon**

*fin*

**fin si**

**fin si**

**fin boucle**

---

lequel s'applique la méthode d'arrondis lorsque celle-ci ne permet pas d'obtenir une solution entière sur l'espace de départ.

Nous donnons ici (algorithme 5) en pseudo-code l'algorithme récursif utilisé par Nediak et Eckstein [43]. Nous avons ici volontairement simplifié la procédure pour qu'y apparaisse uniquement la partie utilisant les coupes de proximité.

Cet algorithme fait appel à deux procédures. La procédure *arrondi* est une procédure qui essaie de trouver une solution entière à partir d'une solution fractionnaire. La procédure *Construire  $Y(U, L)$  tel que  $\|x^* - Y(U, L)\| = 0$*  est la procédure de l'algorithme 4 appelée avec  $r = 0$ .

**Remarque 5.1 :** Il est à noter que quelle que soit la fonction arrondi cette procédure permet toujours de trouver une solution entière réalisable ou d'établir qu'il n'en existe pas en un nombre fini d'itérations. ||

### 5.3.2 Algorithme "explorer puis couper" ([46])

Ici les coupes de proximité sont utilisées pour un algorithme exact d'optimisation. A chaque étape de cette méthode on a une borne inférieure donnée par la meilleure solution réalisable connue, et une borne supérieure donnée par la relaxation continue du problème. Le principe de la méthode est le suivant :

- Étant donné la solution fractionnaire de la relaxation, on établit une inégalité de proximité à un point coupante suivant le principe expliqué dans le paragraphe 5.2.1.

**Algorithme 5** Procédure récursive de Nediak et Eckstein [43]

---

Résoudre en continu le programme linéaire courant  
**si** le programme linéaire est vide **alors**  
 On remonte d'un niveau dans la récursion  
**sinon**  
**si**  $\text{arrondi}(x^*)$  trouve une solution entière réalisable **alors**  
 Une solution réalisable a été trouvée : fin de la procédure.  
**sinon**  
 Soit  $x^*$  l'optimum continu  
 Construire  $Y(U, L)$  tel que  $\|x^* - Y(U, L)\| = 0$   
 Ajouter au problème l'inégalité  $\|x - Y(U, L)\| = 0$ .  
 Appeler récursivement la procédure sur le nouveau problème.  
 Retirer l'inégalité  $\|x - Y(U, L)\| = 0$  et ajouter  $\|x - Y(U, L)\| \geq 1$ .  
 Appeler récursivement la procédure sur le nouveau problème.  
 Retirer l'inégalité  $\|x - Y(U, L)\| \geq 1$  et remonter d'un niveau dans la récursion.  
**fin si**  
**fin si**

---

- On explore l'ensemble des points coupés  $\|x - Y(U, L)\| \leq r - 1$ , en prenant soin de mettre à jour la borne inférieure si une meilleure solution est trouvée au cours de l'exploration..
- On ajoute la coupe à la formulation du problème et on recommence, jusqu'à ce que la relaxation continue courante soit moins bonne que la meilleure solution réalisable obtenue.

La procédure en pseudo-code est explicitée dans l'algorithme 6.

**Algorithme 6** Explorer puis couper

---

Soit  $l \leftarrow -\infty$  la borne inférieure.  
 Soit  $u$  la borne supérieure, initialisée avec la valeur de la relaxation continue.  
**tant que**  $l < u$  **faire**  
 Soit  $x^*$  l'optimum de la relaxation continue.  
 Appeler l'algorithme (4) pour trouver  $Y(U, L)$  tel que  $\|x^* - Y(U, L)\| \leq r - 1$ .  
 Explorer l'ensemble  $\{x \in \{0, 1\}^n : \|x - Y(U, L)\| \leq r - 1\}$ , et mettre à jour la borne inférieure  
 $l \leftarrow \max(l, \max\{c^T x : x \in X \cap \|x - Y(U, L)\| \leq r - 1\})$ .  
 Ajouter l'inégalité exprimant  $\|x - Y(U, L)\| \geq r$ .  
 Mettre à jour  $u$  avec la nouvelle relaxation continue.  
**fin tant que**

---

Dans [46], Oguz propose d'utiliser cette méthode avec comme paramètres pour la fonction (4)  $r = 0$  et donne des résultats d'expériences. La phase d'exploration est faite par énumération explicite dans [46], mais il paraît plus avantageux de le faire par une méthode d'énumération implicite. Nous avons nous-même réalisé des expériences avec plusieurs stratégies dans le choix d'inégalité que nous présenterons au paragraphe 5.5.1.

**Remarque 5.2 :** Ce schéma algorithmique peut être vu comme effectuant des branchements à deux niveaux. Le niveau supérieur est constitué de branchements successifs sur les inégalités géométriques. Le niveau inférieur est celui de l'exploration où on fait appel à une méthode d'énumération classique.

||

## 5.4 Nouveaux schémas algorithmiques proposés

Nous allons maintenant présenter des schémas algorithmiques originaux pour l'utilisation de coupes conditionnellement valides dans la résolution de programmes linéaires en nombres entiers 0/1.

### 5.4.1 Algorithme “couper puis explorer”

Dans l'algorithme 6, on a effectué des explorations sur des sous-espaces du problème déterminés par les relaxations continues successives du problème (notons ces sous-espaces  $Y_1, \dots, Y_m$ ). Les ensembles  $Y_i$  sont explorés au fur et à mesure qu'ils sont trouvés sans suivre aucune stratégie.

Dans la méthode “couper puis explorer” que nous proposons ici, l'exploration des  $Y_i$  n'est pas effectuée avant d'ajouter l'inégalité mais est reportée dans une deuxième phase. Compte-tenu d'une borne inférieure raisonnablement bonne, on ajoute les inégalités sans faire aucune exploration et ce jusqu'à ce que la borne supérieure soit moins bonne que la borne inférieure. Le problème se réduit alors à optimiser sur l'union des ensembles éliminés :

$$\left\{ \begin{array}{l} \max c^T x \\ \cup_i (X \cap Y(U^i, \emptyset)) \end{array} \right.$$

On résout ce nouveau problème en explorant successivement les  $Y_i$  mais en suivant une stratégie donnée. On explicite le pseudo-code dans l'algorithme 7

---

#### Algorithme 7 Couper puis explorer

---

Soit  $l$  une borne inférieure.

Soit  $u$  la borne supérieure, initialisée avec la valeur de la relaxation continue.

Soit  $i \leftarrow 1$ .

**tant que**  $l < u$  **faire**

    Soit  $x^*$  l'optimum de la relaxation continue.

    Appeler l'algorithme (4) pour trouver  $Y_i(U, L)$  tel que  $\|x^* - Y_i(U, L)\| \leq r$ .

    Ajouter l'inégalité exprimant  $\|x - Y_i(U, L)\| \geq r + 1$ .

    Mettre à jour  $u$  avec la nouvelle relaxation continue.

$i \leftarrow i + 1$ .

**fin tant que**

    Énumérer successivement les  $Y_i$  avec une stratégie donnée

---

Nous proposons quatre stratégies pour l'ordre d'exploration sur lesquelles nous donnerons des résultats d'expérience : explorer les ensembles les plus grands d'abord, les plus petits d'abord, ceux de plus grande relaxation d'abord, ceux de plus petite relaxation d'abord. Deux stratégies qui semblent donner de bons résultats sont de prendre les ensembles par taille décroissante, ou par valeur de la relaxation continue décroissante (remarquons que les stratégies contraires qui peuvent paraître intuitivement meilleures ont donné des résultats assez nettement moins bons sur nos jeux de données).

**Remarque 2 (suite) :** Si l'on poursuit l'analogie avec le fait d'effectuer des branchements à deux niveaux, cette méthode peut être vue comme le fait de réordonner les branchements au niveau supérieur

une fois connue une liste suffisante de branchements (au niveau supérieur) à effectuer pour résoudre le problème à l'optimum.

### 5.4.2 Algorithme d'unification des explorations

Dans cette méthode on garde le schéma général de la méthode précédente de coupe puis recherche. Lors de la phase finale d'exploration, on va maintenant modifier l'ensemble à énumérer. Plus précisément, on va créer une nouvelle famille d'ensembles à explorer équivalente ou contenant la précédente de manière à avoir moins d'ensembles à explorer quitte à ce que ceux-ci soient de plus grande taille. L'idée de ceci est que souvent certains ensembles  $Y_i$  à explorer sont très proches. Quand cela est le cas on préfère énumérer un grand ensemble contenant plusieurs  $Y_i$  plutôt que d'énumérer chacun successivement.

On va ainsi créer une nouvelle famille de  $Y^{l_j}$ , où chaque élément contient plusieurs  $Y^i$ . Une telle famille est trouvée au travers d'une petite procédure gloutonne explicitée dans l'algorithme 8. Cet algorithme prend un paramètre *couv*, qui est le pourcentage maximal des  $Y^i$  couvert par chaque  $Y^{l_j}$  (le pourcentage des  $Y^i$  couverts est défini comme le nombre de  $Y^i$  contenus dans  $Y^{l_j}$  divisé par le nombre total d'ensemble  $Y^i$ ). L'algorithme consiste à ajouter à  $Y^{l_j}$  la variable la plus souvent fixée dans les restrictions contenues dans  $Y^{l_j}$  jusqu'à ce que le pourcentage couvert par  $Y^{l_j}$  soit inférieur à  $x$ .

---

#### Algorithme 8 Unification des $Y^i$

---

$Y \leftarrow \{Y^1, Y^2, \dots, Y^k\}$

$i \leftarrow 1$

**tant que**  $Y \neq \emptyset$  **faire**

$K \leftarrow Y$

$Y^{n_i} \leftarrow [0, 1]^n$

**tant que**  $Y^{n_i}$  couvre plus de *couv*% de  $Y$  **faire**

    Trouver  $\hat{i}$  non fixée dans  $Y^{n_i}$  et fixée dans le maximum de  $Y^i \in K$ .

$Y^{n_i} \leftarrow Y^{n_i} \cap x_{\hat{i}}$

$K \leftarrow K \setminus \{Y^i : Y^i \cap \{x_{\hat{i}} = 0\} \neq \emptyset\}$

**fin tant que**

$i++$

$Y \leftarrow Y \setminus k$

**fin tant que**

---

Cette procédure s'insère dans l'algorithme 7 juste avant la phase d'exploration ; celle-ci est alors effectuée en choisissant une des stratégie explicitées au paragraphe précédent.

**Remarque 2 (suite) :** On peut voir cette méthode comme le fait de modifier la séquence de branchements du niveau supérieur avant de lancer l'exploration qui permet d'établir l'optimalité.

Dans des cas extrêmes où les nouveaux ensembles  $Y^{l_j}$  ne contiennent qu'une variable (si par exemple on fixe le paramètre *couv* à 100% dans la procédure d'unification, cette procédure peut être aussi vue comme le fait de fixer des variables dont le branchement est prioritaire (basé sur le fait que dans les ensembles coupés ces variables sont les plus souvent fixées à une valeur donnée).

## 5.5 Expérimentations

Nous allons présenter une série d'expériences avec les trois algorithmes de résolution exacte utilisant des coupes conditionnellement valides que nous avons présentés. Les expériences que nous relatons ici ont été réalisées sur deux séries de sacs à dos multidimensionnels. La première est une série de petits exemples que nous avons générés aléatoirement. Ces problèmes vont de 20 variables et 20 contraintes pour le plus petit à 50 variables et 50 contraintes pour le plus gros. Ces problèmes sont générés de manière à ce que les coefficients directeurs des inégalités et ceux de la fonction de coûts soient corrélés pour qu'il soient tout de même relativement difficiles. La deuxième série de problèmes est une partie des benchmarks de Chu et Beasley [17] présentés au chapitre 3 page 53, les expériences ont été effectués sur les 13 premiers problèmes à 100 variables et 30 contraintes.

Dans toutes les expériences les phases d'explorations sont effectués par CPLEX en mode MIP. Dans chacune des expériences, nous comparerons les résultats obtenus avec ceux donnés par CPLEX 7.0 en mode MIP. En fonction de notre expérience de la résolution de ces problèmes, nous avons effectué quelques réglages dans la résolution par CPLEX pour la rendre plus rapide. Nous avons désactivé les générations d'inégalités valides à l'exception des inégalités de Gomory et des inégalités de cover, et nous avons désactivé l'appel aux résolutions heuristique (ces réglages s'appliquent aussi bien aux résolutions complètes qu'aux résolutions pour résoudre les sous-problèmes des phase d'explorations des différents algorithmes). La durée totale de résolution a été limitée à 3 heures.

### 5.5.1 Résultats avec l'algorithme "explorer puis couper"

Nous présentons ici des résultats sur l'algorithme "explorer puis couper" présenté au paragraphe 5.3.2 Ici, outre l'efficacité des algorithmes d'exploration et coupe nous avons voulu comparer plusieurs stratégies pour la génération des inégalités. La première stratégie est celle proposée par Oguz pour la résolution des problèmes 0/1 ; elle consiste à construire les inégalités en faisant appel à la fonction 4 avec comme paramètre  $r = 0$ . La seconde est une stratégie adaptée au problème de sac à dos multidimensionnel. Dans celle-ci les coupes générés sont toujours avec  $L = \emptyset$ , ceci pour plusieurs raisons. Premièrement parce que les inégalités générées sont ainsi elles-mêmes des inégalités de sac à dos et elles ne changent donc pas la structure du problème. Deuxièmement, on a considéré que fixer une variable à 1 réduit plus la complexité du problème que fixer une variable à 0 car lorsqu'une variable est fixée à 1 on réduit le nombre d'objets pouvant être mis dans le sac (c'est à dire la profondeur maximale de la recherche arborescente).

Dans le tableau 5.1, nous donnons les résultats d'expériences pour ces deux stratégies. Dans les deux premières colonnes du tableau, nous indiquons la provenance et le numéro du benchmark. Dans les troisièmes et quatrièmes colonnes, nous indiquons le nombre de variables et le nombre de contraintes de chaque problème. Dans les colonnes suivantes, nous indiquons successivement pour CPLEX brut puis pour les deux stratégies le temps total de résolution et le nombre total de nœuds explorés dans les recherches arborescentes (pour CPLEX ceci est juste le nombre de nœuds explorés, pour les autres stratégies ceci est la somme des nombres de nœuds explorés au cours de toutes les recherches arborescentes). Pour chacun des problèmes on a mis en gras les temps de la méthode la plus rapide.

Les résultats montrent la supériorité de la seconde stratégie pour les sac à dos multidimensionnels. Par contre les deux méthodes sont en général moins bonnes que CPLEX.

				Cplex		Stratégie 1		Stratégie 2	
		n var	n cont	Temps	Nnoeuds	Temps	Nnoeuds	Temps	Nnoeuds
Problèmes générés aléatoirement †	1	25	20	<b>0.54</b>	789	6,1	2277	1.13	797
	2	25	20	<b>0.81</b>	1108	11,1	4198	1.98	1877
	3	25	20	<b>0.89</b>	1564	11,11	4307	2.08	2284
	4	25	20	<b>1.41</b>	2924	23,51	8621	2.83	2837
	5	25	20	<b>0.83</b>	1138	9,69	3451	1.61	929
	6	50	20	<b>229.74</b>	237599	4207,53	1375748	437.85	426099
	7	50	20	<b>85.29</b>	92339	1380,85	436117	132.46	119518
	8	50	20	<b>409.88</b>	391442	5107,25	1630983	505.04	467832
	9	50	20	<b>218.71</b>	238625	2368,6	937411	271.86	246422
	10	50	20	48.18	48689	1038,72	251416	<b>67.78</b>	53549
	11	50	50	3495.53	1545472	+++	+++	<b>2817.16</b>	1376814
	12	50	50	3238.98	1604764	+++	+++	<b>3133.85</b>	1546495
Problèmes de Chu et Beasley	1	100	30	<b>625,64</b>	447231	+++	+++	782,25	492279
	2	100	30	3404,06	2290686	+++	+++	4065,75	2545706
	3	100	30	<b>1426,06</b>	1020853	+++	+++	2324,76	1474232
	4	100	30	<b>1994,37</b>	1364870	+++	+++	3133,73	1942003
	5	100	30	<b>7931,13</b>	4889036	+++	+++	9675,3	5925664
	6	100	30	<b>8918,28</b>	6011635	+++	+++	+++	+++
	7	100	30	9389,95	6093520	+++	+++	+++	+++
	8	100	30	6807,92	4242182	+++	+++	9644,27	4902726
	9	100	30	+++	+++	+++	+++	+++	+++
	10	100	30	<b>1198,92</b>	764644	+++	+++	1540,45	854290
	11	100	30	<b>8388,46</b>	5340652	+++	+++	+++	+++
	12	100	30	+++	+++	+++	+++	+++	+++
	13	100	30	+++	+++	+++	+++	+++	+++

FIG. 5.1 – Comparaison des méthodes explorer puis couper

			Cplex		Stratégie 1		Stratégie 2	
	n var	n cont	Temps	Nnoeuds	Temps	Nnoeuds	Temps	Nnoeuds
7.01	100	30	<b>3404,06</b>	2290686	4366,33	2584576	4060,72	2429698
7.02	100	30	<b>1426,06</b>	1020853	2348,11	1512209	2327,86	1460225
7.03	100	30	<b>1994,37</b>	1364870	3437,34	2085310	3291,15	1928943

FIG. 5.2 – Résultat de “couper puis chercher” avec les stratégies 1 et 2

### 5.5.2 Résultats avec l’algorithme “couper puis explorer” simple

Nous présentons ici des résultats pour l’algorithme présenté au paragraphe 5.4.1. Pour la séparation des coupes, on a choisi la deuxième stratégie du paragraphe précédent. Comme on l’a vu, l’algorithme nécessite une borne inférieure. Pour obtenir celle-ci, on effectue ici l’exploration sur les 3 premiers  $Y^i$  trouvés (ceux-ci ne seront donc pas explorés par la suite), ce qui permet toujours pour les sac à dos multidimensionnels de trouver une très bonne borne. Nous avons expérimenté les quatre stratégies proposées au paragraphe 5.4.1 :

- La stratégie 1 est d’explorer les ensembles du plus petit au plus grand.
- la stratégie 2 est d’explorer par ordre croissant des relaxations continues.
- La stratégie 3 est d’explorer les ensembles du plus grand au plus petit
- la stratégie 4 est d’explorer par ordre décroissant des relaxations continues.

Le tableaux 5.2 donnent les résultats d’expériences en utilisant les stratégies 1 et 2 respectivement (les résultats portent sur peu de problèmes étant donné la mauvaise qualité de ceux-ci pour ces deux stratégies). Les problèmes traités sont des problèmes tests de Chu et Beasley. Pour chacun des problèmes, on indique le temps de résolution totale et le nombre totale de nœuds des recherches arborescentes.

Dans le tableau 5.3, on donne la provenance et les numéros des problèmes traités ; puis on donne les résultats de CPLEX brut puis ceux des stratégies d’énumération 3 et 4. Dans les colonnes de résultats on donne le temps et le nombre de nœuds.

Les stratégies 1 et 2 auraient pu paraître intuitivement plus intéressantes car les ensembles les plus petits y sont explorés en premier, mais les résultats semblent indiquer clairement que ces stratégies sont moins efficaces que les deux dernières sur les problèmes de sac à dos multidimensionnels.

Les stratégies 3 et 4 semblent elles donner des résultats assez proches l’une de l’autre. Dans les deux cas, l’optimum est trouvé dans les toutes premières explorations effectuées. La stratégie 1 se comporte dans quatre cas mieux que CPLEX et en général les résultats sont assez proches entre les trois méthodes.

### 5.5.3 Résultats avec la méthode de coupe puis réunion puis exploration

Le tableau 5.4 récapitule les expériences réalisées avec la méthode du paragraphe 5.4.2. Sur les problèmes étudiés cette méthode est en moyenne 5% plus rapide que CPLEX brut. La méthode donne de meilleurs résultats dans quinze cas (dont quatre assez nettement), CPLEX est meilleur dans 7 cas.

				Cplex		Stratégie 3		Stratégie 4	
		n var	n cont	Temps	Nnoeuds	Temps	Nnoeuds	Temps	Nnoeuds
Problèmes générés aléatoire ment	1	25	20	<b>0.54</b>	789	1.13	818	1.17	791
	2	25	20	<b>0.81</b>	1108	1.26	1752	1.45	1810
	3	25	20	<b>0.89</b>	1564	2.26	2617	2.26	2300
	4	25	20	<b>1.41</b>	2924	2.18	2537	2.34	2631
	5	25	20	<b>0.83</b>	1138	1.23	1102	1.22	1035
	6	50	20	<b>229.74</b>	237599	317.17	319921	365.5	370559
	7	50	20	<b>85.29</b>	92339	107.7	106623	107.73	108054
	8	50	20	<b>409.88</b>	391442	395.68	411503	431.6	426439
	9	50	20	<b>218.71</b>	238625	217.71	224049	236.45	227088
	10	50	20	48.18	48689	48.18	46440	<b>42.99</b>	41232
	11	50	50	3495.53	1545472	2833.46	1372272	<b>2831.43</b>	1373888
	12	50	50	3238.98	1604764	<b>3193.17</b>	1542384	<b>3213.11</b>	1582074
Problèmes de Chu et	7.00	100	30	<b>625,64</b>	447231	819,84	481969	849,25	453211
	7.01	100	30	3404,06	2290686	<b>3187,69</b>	2182649	3736,69	2445226
	7.02	100	30	<b>1426,06</b>	1020853	3670,22	2615322	1842,77	1262530
	7.03	100	30	<b>1994,37</b>	1364870	2353,49	1590578	2474,7	1625587
	7.04	100	30	<b>7931,13</b>	4889036	12343,4	7768050	8265,03	5293677
	7.05	100	30	<b>8918,28</b>	6011635	9569,22	6591361	11173,8	6255904
	7.06	100	30	9389,95	6093520	<b>7379,53</b>	4949781	10021,1	5339253
	7.07	100	30	6807,92	4242182	<b>6514,74</b>	4309439	8288,9	4440462
	7.08	100	30	+++	+++	+++	+++	+++	+++
	7.09	100	30	<b>1198,92</b>	764644	1337,7	744625	1446,89	809452
	7.10	100	30	<b>8388,46</b>	5340652	8754,7	4623528	9490,79	4853044
	7.11	100	30	+++	+++	+++	+++	+++	+++
7.12	100	30	+++	+++	+++	+++	+++	+++	

FIG. 5.3 – Résultat de “couper puis chercher” avec les stratégies 3 et 4

				Cplex		Stratégie 4	
		n var	n cont	Temps	Nnoeuds	Temps	Nnoeuds
Problèmes généralisés aléatoires	1	25	20	0.54	789	<b>0,52</b>	779
	2	25	20	<b>0.81</b>	1108	0,86	1629
	3	25	20	0.89	1564	<b>0,86</b>	1721
	4	25	20	<b>1.41</b>	2924	<b>1,38</b>	2584
	5	25	20	<b>0.83</b>	1138	<b>0,8</b>	1341
	6	50	20	<b>229.74</b>	237599	345,59	428041
	7	50	20	<b>85.29</b>	92339	95,76	114212
	8	50	20	409.88	391442	<b>323,62</b>	397800
	9	50	20	218.71	238625	<b>204,44</b>	256681
	10	50	20	48.18	48689	<b>45,22</b>	55672
	11	50	50	3495.53	1545472	<b>2480,14</b>	1447940
	12	50	50	3238.98	1604764	<b>2782.53</b>	1641876
Problèmes de Chu et	7.00	100	30	<b>625,64</b>	447231	645.55	452523
	7.01	100	30	3404,06	2290686	3142,84	2043244
	7.02	100	30	<b>1426,06</b>	1020853	1931,86	1332646
	7.03	100	30	<b>1994,37</b>	1364870	2422,42	1580635
	7.04	100	30	<b>7931,13</b>	4889036	8992,49	5748958
	7.05	100	30	8918,28	6011635	<b>8847,97</b>	6165260
	7.06	100	30	9389,95	6093520	<b>7427,81</b>	5128032
	7.07	100	30	6807,92	4242182	<b>6531,36</b>	4309623
	7.08	100	30	+++	+++	+++	+++
	7.09	100	30	1198,92	764644	<b>1188,29</b>	816490
	7.10	100	30	8388,46	5340652	<b>7040,63</b>	4422513
	7.11	100	30	+++	+++	<b>11115,6</b>	7058737
7.12	100	30	+++	+++	+++	+++	

FIG. 5.4 – Résultat de coupe puis réunion

## 5.6 Conclusions

Il y a une grande liberté de choix pour l'intégration de coupes conditionnellement valides dans un algorithme de résolution exacte. Comme pour les inégalités valides classiques, on doit effectuer des choix portant sur la fréquence de génération, le nombre total d'inégalités générées, etc...

Mais en plus on doit choisir quand et comment effectuer les énumérations nécessaires pour valider les coupes ajoutées. Les schémas algorithmiques que nous avons proposés et expérimentés sur des problèmes de sac à dos multidimensionnels diffèrent surtout dans ces choix. Ces expériences montrent que, suivant le schéma utilisé, on obtient des résultats très différents en terme de temps de calcul pour un même type de coupes. Particulièrement, sur cette étude, il semble que le choix le plus évident, qui consiste à effectuer les explorations au fur et à mesure que des coupes sont ajoutées, donne de moins bons résultats que des schémas plus élaborés où les explorations sont reportées et effectuées dans un ordre différent de celui de l'apparition des coupes. D'après nos expériences, il nous semble qu'il faut favoriser l'exploration des ensembles de plus grandes tailles, ou sur lesquels la valeur de la relaxation est la plus forte (contenant potentiellement les meilleures solutions réalisables).

Une caractéristique qui semble commune aux différents travaux connus et aux nôtres est la possibilité de récupérer très vite de très bonnes solutions réalisables grâce aux explorations effectuées, ce qui explique sûrement pourquoi la plupart de ces travaux portent sur des résolutions heuristiques. Un aspect nouveau que semblent indiquer nos expériences est qu'il est aussi possible (au moins pour des problèmes de sac à dos multidimensionnels) d'accélérer la résolution complète des problèmes en utilisant des coupes conditionnellement valides.

Par rapport aux travaux précédents, lorsque les inégalités sont utilisées dans une résolution complète, une question préoccupante est de savoir dans quelle mesure les inégalités conditionnellement valides se rattachent à des méthodes de plans coupants ou à des stratégies de branchement. C'est pourquoi il serait intéressant de tester ces schémas algorithmiques avec d'autres types de coupes comme celles développées par Burdet et Glover qui demandent moins d'efforts d'énumération.

Il serait aussi intéressant d'expérimenter sur des problèmes plus généraux, et d'étudier l'application récursive de coupes conditionnellement valides (c'est à dire de réutiliser des coupes dans les phases d'exploration).

# Conclusions

Dans cette étude, nous avons expérimenté plusieurs méthodes de génération d'inégalités valides pour la programmation mixte. On peut classer ces méthodes en trois familles : les méthodes de séparation par oracle présentées au chapitre 3, les méthodes de séparation par reformulation dans un espace de dimension supérieure et projection présentées au chapitre 4 et 5, et les méthodes utilisant des inégalités conditionnellement valides présentées au chapitre 6.

Nous nous proposons de revenir ici, pour chacune de ces méthodes, sur ce qui nous semble nouveau dans ce que nous avons proposé, sur les principaux résultats obtenus, sur les conclusions que nous en tirons et les pistes pour de futurs travaux qui semblent s'en dégager.

## Séparation par oracle

Au chapitre 3, nous avons utilisé une méthode par oracle pour séparer l'inégalité la "plus" coupante, évaluer sa qualité et tester diverses méthodes approchées pour essayer d'obtenir une coupe de qualité comparable dans un temps de calcul raisonnable. Nous avons proposé et expérimenté des méthodes approchées basées sur l'utilisation pour la résolution de l'oracle d'une heuristique et d'une relaxation mixte du problème. Nous avons aussi proposé d'effectuer la séparation dans une restriction puis d'étendre la coupe obtenue par une méthode de lifting relaxé qui permet d'obtenir une facette d'un polyèdre compris entre l'enveloppe convexe des solutions entières et la relaxation où toutes les variables entières dans la solution continue sont relâchées.

Concernant la meilleure coupe, les expériences faites indiquent que, si son temps de séparation est tout à fait prohibitif, sa qualité du point de vue du critère du ratio et son impact en terme de renforcement sont très forts. La méthode par lifting relaxé, elle, permet d'obtenir en un temps nettement plus court (en moyenne 80 fois plus vite) une coupe dont la qualité (au vu des critères choisis) est très proche (en moyenne 86% du ratio et 82% du renforcement obtenu avec la meilleure coupe).

Même s'il est à noter qu'avec la méthode de lifting relaxé on a un temps total de résolution (séparation comprise) inférieur en moyenne quand on ajoute une coupe, l'expérimentation de la résolution complète en ajoutant les coupes séparées à la formulation ne donne pas des résultats satisfaisants. En l'absence de structure spécifique, il paraît difficile d'accélérer significativement les temps de séparation de la méthode par lifting relaxé en restreignant encore plus le problème. Effectivement, comme il est mentionné au chapitre 2, se placer dans une restriction ne permet de simplifier le problème de séparation que si la restriction est sur une face du polyèdre duquel on veut séparer. Pour améliorer les résultats obtenus, une approche possible serait d'utiliser une méthode de génération multiple d'inégalités utilisant efficacement les calculs effectués pour séparer la première inégalité pour déduire

plusieurs inégalités valides.

## Séparation par reformulation et projection

Au chapitre 4, nous avons étudié des techniques dérivées de la programmation disjonctive (Balas [4]). Nous avons proposé une méthode pour séparer des coupes sur l'intersection des polyèdres disjonctifs simples (ou fermeture élémentaire des coupes de lift-and-project selon Cornuéjols et Li [19]). Contrairement au chapitre 3, ici on ne cherche pas une inégalité la plus coupante mais un ensemble de coupes permettant d'obtenir le meilleur renforcement.

Les résultats expérimentaux montrent qu'avec cette méthode, en utilisant uniquement des coupes de lift-and-project de rang 1, on peut obtenir des renforcements significativement meilleurs que ceux obtenus en séparant la meilleure inégalité sur chacune des disjonctions simples (au prix d'un temps de calcul supérieur). En particulier sur des problèmes de MAX2SAT la méthode permet d'obtenir directement l'optimum entier dans 49 problèmes sur 60.

Au chapitre 5, nous avons proposé une méthode permettant de renforcer les résultats obtenus au chapitre précédent par l'intermédiaire de la relaxation de Sherali-Adams. La relaxation de Sherali-Adams pouvant être vue comme un renforcement de l'intersection des polyèdres disjonctifs par l'introduction de contraintes supplémentaires qui couplent les polyèdres disjonctifs entre eux, nous avons proposé une méthode qui introduit itérativement ces contraintes en conservant le plus possible la structure décomposable du problème. Cette méthode permet de renforcer le problème en fonction du temps de calcul disponible, en ajoutant uniquement une partie des contraintes couplantes.

Nos expériences de calcul indiquent que, pour la plupart des problèmes traités, l'introduction d'une faible partie des contraintes couplantes seulement permet d'atteindre l'optimum de la relaxation de Sherali-Adams. Sur les problèmes testés, les expériences montrent que pour une petite partie d'entre eux seulement la relaxation de Sherali-Adams est significativement plus forte que la fermeture élémentaire des coupes de lift and project. En particulier, pour les problèmes de MAX2SAT, les résultats obtenus sont toujours équivalents sur les problèmes testés. Nous avons montré cette équivalence de manière théorique pour les problèmes de maximisation de fonctions pseudo-booléennes quadratiques qui peuvent être utilisés pour modéliser entre autres MAX2SAT.

Les résultats obtenus dans ces deux chapitres ouvrent plusieurs perspectives de travail. Tout d'abord, on voudrait confirmer ou infirmer les résultats pratiques du chapitre 4 sur la force des coupes de rang 1 en effectuant des expérimentations étendues à des problèmes de plus grande taille que ceux que nous avons traité (qui ont tous moins de 250 variables). Il serait aussi intéressant d'étudier l'utilisation d'algorithmes pour optimiser sur la fermeture des coupes de lift and project qui prendrait mieux en compte le critère de renforcement dans le problème de séparation. Enfin, il serait intéressant de chercher et d'expérimenter des méthodes similaires pour optimiser sur les fermetures élémentaires d'autres familles d'inégalités valides plus fortes.

En ce qui concerne le chapitre 5, tout d'abord, au vu des résultats, il paraîtrait utile d'étudier du point de vue théorique les conditions sous lesquelles la relaxation de Sherali-Adams est différente de la fermeture élémentaire des coupes de lift and project. A notre connaissance l'équivalence entre les deux relaxations a seulement été étudiée pour deux problèmes : le stable maximum par Lovász et Shrijver dans [39] et l'optimisation de fonctions pseudo-booléennes quadratiques qui est un des résultats originaux de notre chapitre 5. Une autre piste est d'étudier du point de vue pratique l'utili-

sation de relaxations plus fortes que la relaxation de Sherali-Adams. Une première façon de renforcer cette relaxation est d'utiliser des contraintes matricielles de semi-définie positivité comme cela a été proposée par Lovász et Schrijver dans [39].

### **Utilisation d'inégalités conditionnellement valides**

Au chapitre 6, nous avons proposé plusieurs schémas algorithmiques nouveaux pour l'utilisation de coupes conditionnellement valides dans la résolution exacte d'un programme mixte. Contrairement aux chapitres précédents, nous ne nous sommes pas concentrés ici sur le problème de séparation, qui est simple pour la classe de coupes que nous avons utilisées, mais sur la manière d'utiliser des coupes qui nécessitent un effort d'énumération dans un algorithme de résolution exacte. Les expérimentations effectuées montrent que le choix du schéma algorithmique a une grande importance dans l'utilisation de telles coupes.

Il serait intéressant de tester les schémas algorithmiques proposés avec d'autres familles de coupes conditionnellement valides (comme celles développées à partir des coupes d'intersections de Balas [3]), et d'effectuer des tests étendus sur des problèmes plus généraux.



# Bibliographie

- [1] D. Applegate, R. Bixby, V. Chvátal, and W. Cook. Tsp cuts which do not conform to the template paradigm. In D. Naddef M. Jünger, editor, *Computational Combinatorial Optimization*, Lecture Notes in Computer Science. Springer, 2001.
- [2] E. Balas. Intersection cuts - a new type of cutting planes for integer programming. *Operations Research*, 19 :19–39, 1971.
- [3] E. Balas. Integer programming and convex analysis : Intersection cuts from outer polars. *Mathematical Programming*, 2 :330–382., 1972.
- [4] E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5 :3–51, 1979.
- [5] E. Balas. Projection with a minimal system of inequalities. *Computational Optimization and Applications*, 10 :189–193, 1998.
- [6] E. Balas, S. Ceria, G. Cornuéjols, and N. Natraj. Gomory cuts revisited. *Operations Research Letters*, 19 :1–9, 1996.
- [7] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58 :295–324, 1993.
- [8] E. Balas, S. Ceria, and G. Cornuéjols. Mixed 0-1 programming by lift-and-project in a branch-and-cut framework. *Management Sci.*, 42 :1229–1246, 1996.
- [9] E. Balas, S. Ceria, M. Dawande, F. Margot, and G. Pataki. Octane : A new heuristic for pure 0-1 programs. *Operations Research*, 49(2) :207–225, 2001.
- [10] E. Balas and M. Perregaard. Lift and project for mixed 0-1 programming : Recent progress. Technical Report MSSR No. 627, Graduate School of Industrial Administration, Carnegie Mellon University, September 1999.
- [11] E. Balas and E. Zemel. Facets of the knapsack polytope from minimal covers. *SIAM Journal on Applied Mathematics*, 34 :119 – 148, 1978.
- [12] R. Borndörfer and R. Weismantel. Discrete relaxations of combinatorial programs. *Discrete Applied Mathematics*, 112 :11–26, 2001.
- [13] C.-A. Burdet. Enumerative inequalities in integer programming. *Mathematical Programming*, 2 :32–64, 1972.
- [14] A. Cagnard. Algorithmes gloutons et métaheuristiques en dimensionnement de réseaux de télécommunication. Master’s thesis, DEA-IRO Paris 6, 2002. (sous la supervision de V. Gabrel, A. Knippel et M. Minoux).
- [15] A. Caprara and A. N. Letchford. On the separation of split cuts and related inequalities. *Mathematical Programming*, 94(2-3) :279–294, 2003.

- [16] S. Ceria and G. Pataki. Solving integer and disjunctive programs by lift and project. In *IPCO*, pages 271–283, 1998.
- [17] P.C. Chu and J.E. Beasley. A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, 4 :63–86, 1998.
- [18] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial optimization. *Discrete Mathematics*, 4 :305–337, 1973.
- [19] G. Cornuéjols and Y. Li. Elementary closures for integer programs. *Operations Research Letters*, 28 :1–8, 2001.
- [20] G. Cornuéjols, Y. Li, and D. Vandenbussche. K-cuts : a variation of gomory mixed integer cuts from the lp tableau. *To appear in INFORMS Journal on Computing*.
- [21] H. P. Crowder, E. L. Johnson, and M. Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31 :803–834, 1983.
- [22] S. de Vries D. Cheng. Antiweb-wheel inequalities and their separation problems over the stable set polytopes. *Math. Program.*, 92 :153–175, 2002.
- [23] G. Dantzig, D. Fulkerson, and S. Johnson. Solution of a large scale traveling salesman problem. *Oper. Res.*, 2 :393–410, 1954.
- [24] F. Eisenbrand. On the membership problem for the elementary closure of a polyhedron. *Combinatorica*, 19(2), 1999.
- [25] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming Online First Publications*, (DOI 10.1007/s10107-003-0395-5), 2003.
- [26] Fabrice Fourcade, Ellis Johnson, Mourad Bara, and Philippe Cortey-Dumont. Optimizing nuclear power plant refueling with mixed-integer programming. *European Journal of Operational Research*, pages 269–280, 1997.
- [27] V. Gabrel and M. Minoux. A scheme for exact separation of extended cover inequalities and application to multidimensional knapsack problems. *Operations Research letters*, 30 :252–264, 2002.
- [28] F. Glover. Cut search methods in integer programming. *Mathematical Programming*, 3 :86–100, 1973.
- [29] R. Gomory. An algorithm for the mixed integer problem. Technical Report RM-2597, The RAND corporation, 1960.
- [30] R.E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. Amer. Soc.*, 64 :275–278, 1958.
- [31] R.E. Gomory. Solving linear programming problems in integers. In R. Bellman and M.Hall, editors, *Combinatorial Analysis, Proceedings of Symposia in Applied Mathematics 10*, pages 211–216, Providence, RI, 1960.
- [32] R.E. Gomory. An algorithm for integer solution solutions to linear programming. In R.L. Graves and P. Wolfe, editors, *Recent Advances in Mathematical Programming*, pages 269–302. McGraw-Hill, 1963.
- [33] M. Grotschel. The symmetric travelling salesman problem : solution of a 120-city problem. *Mathematical Programming Study*, 12 :61–77, 1980.
- [34] Z. Gu, G. L. Nemhauser, and M. W. P. Savelsbergh. Lifted cover inequalities for 0–1 integer programs : computation. *INFORMS Journal on Computing*, 10 :427–438, 1998.

- [35] A. Knippel, V. Gabrel, and M. Minoux. Exact solution of multicommodity network optimization problems with general step cost functions. *Operations Research Letters*, 25 :15–23, 1999.
- [36] Arnaud Knippel. *Modèles et algorithmes de multiflôts à coût discontinu pour l'optimisation de réseaux de télécommunications*. PhD thesis, Université Paris 6, 2001.
- [37] A. Land. The solution of some 100-city travelling salesman problems. non publié, 1979.
- [38] A. N. Letchford and A. Lodi. Strengthening chvatal-gomory cuts and gomory fractional cuts. *Operations Research Letters*, 30(2) :74–82, 2002.
- [39] L. Lovász and A. Schrijver. Cone of matrices and set functions, and 0, 1 optimization. *SIAM J. Optimization*, 1(2) :166–190, May 1991.
- [40] N. Maculan, E. M. Macambira, and C. C. de Souza. Geometrical cuts for 0-1 integer programming. Technical Report IC-02-006, Instituto de Computação, Universidade Estadual de Campinas, 2002.
- [41] H. Minkowski. *Geometrie der zahlen (erste lieferung)* taubner,. Leipzig, 1896.
- [42] M. Minoux. Discrete cost multicommodity network optimization problems and exact solution methods. *Annals of Operations Research*, 106 :19–46, 2001.
- [43] M. Nediak and J. Eckstein. Pivot, cut and dive : A heuristic for 0-1 mixed integer programming. Technical Report RRR 53-2001, RUTCOR, October 2001.
- [44] G.L. Nemhauser and L.A. Wolsey. A recursive procedure to generate all cuts for 0 – 1 mixed integer programs. *Math. Prog.*, 46 :379–390.
- [45] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
- [46] O. Oguz. Search and cut : New class of cutting planes for 0-1 programming. [http://www.optimization-online.org/DB\\_HTML/2002/05/484.html](http://www.optimization-online.org/DB_HTML/2002/05/484.html), 2002.
- [47] H. Ouzia. Séparation d'inégalités valides par la programmation disjonctive. Master's thesis, DEA-IRO Paris 6, 2002.
- [48] M. Padberg and S. Hong. On the symmetric travelling salesman problem : a computational study. *Math. Programm. Study*, 12 :78–107, 1980.
- [49] M. W. Padberg. (1,k)-configurations and facets for packing problems. *Mathematical Programming*, 18 :94–99, 1980.
- [50] M.W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5 :199–215, 1973.
- [51] M. Perregaard and E. Balas. Generating cuts from multiple-term disjunctions. In *IPCO*, pages 348–360, 2001.
- [52] T. Roy and L. Wolsey. Solving mixed integer programming problems using automatic reformulation. *Operations Research*, 35 :45–57, 1987.
- [53] A. Schrijver. On cutting planes. *Ann. Discrete Math*, 9 :291–296, 1980.
- [54] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between continuous and convex hull representations for zero one programming problems. *SIAM J. Discrete Math.*, 1990.
- [55] H.D. Sherali and W.P. Adams. *A reformulation-linearization technique for solving discrete and continuous nonconvex problems*. Kluwer Academic Publishers, 1999.
- [56] L. A. Wolsey. Valide inequalities for 0 – 1 knapsacks and mips with generalized upper bound constraint. *Discrete Applied Mathematics*, 29 :251–261, 1990.

- [57] L.A. Wolsey. Faces for a linear inequality in 0/1 variables. *Mathematical Programming*, 8 :165–178, 1975.
- [58] L.A. Wolsey. Facets and strong valid inequalities for integer programs. *Operations Research*, 24 :367–372, 1976.
- [59] F. Zeghal and M. Minoux. Modélisation et résolution d'un problème d'affectation d'équipages en transport aérien. In *Actes de la conférence MOSIM01*, pages 25–27, Troyes, France, Avril 2001.
- [60] E. Zemel. Lifting the facets of zero-one polytopes. *Mathematical Programming*, 15 :268–277, 1978.