



Titre: Stabilisation de l'algorithme de génération de colonnes
Title:

Auteur: Hatem Ben Amor
Author:

Date: 2002

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Ben Amor, H. (2002). Stabilisation de l'algorithme de génération de colonnes
Citation: [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie.
<https://publications.polymtl.ca/7056/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/7056/>
PolyPublie URL:

**Directeurs de
recherche:** François Soumis, & Jacques Desrosiers
Advisors:

Programme: Non spécifié
Program:

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

**ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600**

UMI[®]

UNIVERSITÉ DE MONTRÉAL

STABILISATION DE L'ALGORITHME
DE GÉNÉRATION DE COLONNES

HATEM BEN AMOR
DÉPARTEMENT DE MATHÉMATIQUES
ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIAE DOCTOR (Ph.D.)
(MATHÉMATIQUES DE L'INGÉNIEUR)

JUIN 2002



**National Library
of Canada**

**Acquisitions and
Bibliographic Services**

**395 Wellington Street
Ottawa ON K1A 0N4
Canada**

**Bibliothèque nationale
du Canada**

**Acquisitions et
services bibliographiques**

**395, rue Wellington
Ottawa ON K1A 0N4
Canada**

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-73430-7

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

STABILISATION DE L'ALGORITHME
DE GÉNÉRATION DE COLONNES

présentée par : BEN AMOR Hatem

en vue de l'obtention du diplôme de : Philosophiae Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. SAVARD, Gilles, Ph.D., président

M. SOUMIS François, Ph.D., membre et directeur de recherche

M. DESROSIERS Jacques, Ph.D., membre et codirecteur de recherche

M. AUDET Charles, Ph.D., membre

M. FRANGIONI Antonio, Ph.D., membre externe

À la mémoire de mon oncle Zouhaier

À mes parents

À Hanene, Habib, Hazem et Hassen

À toute la famille

À mes amis

Remerciements

Il m'est agréable de pouvoir remercier mes directeurs de recherche, M. Jacques Desrosiers et M. François Soumis, qui par leurs précieux conseils et leurs permanents encouragements, m'ont aidé à mener à bien ce travail. Je désire leur témoigner ici de ma profonde gratitude et de rendre hommage à leur compétence et à leur inaltérable gentillesse.

Je tiens aussi à exprimer ma reconnaissance à M. Charles Audet, M. Antonio Frangioni et M. Gilles Savard pour l'intérêt qu'ils ont bien voulu porter à ce travail en acceptant de faire partie du jury.

Je remercie M. Antonio Frangioni pour l'intérêt particulier qu'il a porté à mon travail. Ses remarques, ses suggestions et les discussions que nous avons eues ont contribué à améliorer la qualité de la présentation de la thèse et particulièrement le chapitre 2.

Mes plus vifs remerciements s'adressent aux membres du GERAD et du département de mathématiques de l'École Polytechnique dans lesquels j'étais si bien entouré.

Résumé

Le thème central de cette thèse est la stabilisation des algorithmes de génération de colonnes et de Kelley. Des approches unifiées pour les méthodes de stabilisation qui suivent le schéma des méthodes de faisceaux ont été proposées récemment. Nous introduisons une approche unifiée pour les méthodes de stabilisation de type proximal. Quoique les travaux aient été développés indépendamment, l'idée que nous proposons rentre dans le cadre des travaux de Frangioni [20] et présente certaines ressemblances avec l'approche de Neame [67]. Nous proposons également deux algorithmes de stabilisation de l'algorithme de génération de colonnes basés sur des cas particuliers de l'approche unifiée proposée que nous appliquons à trois problèmes connus.

Au chapitre 1, nous présentons les deux approches connues pour la résolution des problèmes de grande taille : décomposition de Dantzig-Wolfe/génération de colonnes et relaxation lagrangienne/méthodes d'optimisation non-différentiable. La première approche résout ce que nous appellerons le problème primal et la deuxième résout son dual. Nous faisons une synthèse des développements les plus récents concernant ces deux méthodes. Les problèmes de dégénérescence primale et duale causent de l'instabilité dans le comportement de ces algorithmes dont l'effet est plus grand pour les problèmes de grande taille. Les premières approches de stabilisation ont été regroupées en deux classes : méthodes du type faisceaux et méthodes du type proximal. Un survol des approches unifiées récentes a été inclu à la fin du chapitre.

Nous introduisons une approche unifiée pour les méthodes de stabilisation de l'algorithme de Kelley de type proximal au chapitre 2. Nous considérons le problème de maximisation d'une fonction concave. Nous suivons l'idée de Kim *et al.*(1994)

qui est elle-même une généralisation du principe de l'algorithme du point proximal dû à Rockafellar (1976). Cet algorithme consiste à résoudre une suite de problèmes pénalisés autour du point dual courant. Nous proposons des fonctions de pénalité qui s'annulent sur des régions de confiance contenant le point dual courant que nous appelons centre de stabilité. Les problèmes pénalisés, appelés problèmes stabilisés, sont résolus par un algorithme de plans coupants. La convergence est étudiée en utilisant des régions de confiance de pleine dimension. L'utilisation de ce type de régions de confiance permet de prouver la convergence finie de l'algorithme même pour des fonctions non-polyédrales en supposant la résolution exacte des problèmes stabilisés. Nous motivons également le choix de fonctions de pénalité linéaires par morceaux pour la stabilisation de l'algorithme de génération de colonnes.

La première approche de stabilisation de l'algorithme de génération de colonnes est présentée au chapitre 3. Nous proposons la pénalisation du dual par une fonction linéaire à cinq morceaux. Le problème résultant est appelé dual stabilisé et son dual est appelé primal stabilisé. L'algorithme utilise deux types d'itérations. Une itération majeure correspond à la résolution d'une paire de problèmes stabilisés primal et dual par un algorithme de plans coupants. Les itérations de cet algorithme sont appelées itérations mineures. L'utilisation des pénalités permet de limiter les déplacements dans l'espace dual et d'assurer une croissance de l'objectif dual à chaque itération majeure. Du point de vue primal, la stabilisation correspond à la pénalisation du lagrangien du primal pour les multiplicateurs qui sont à l'intérieur de la région de confiance.

Nous montrons d'abord qu'une solution optimale duale peut être utilisée pour calculer une solution primale optimale de base. L'efficacité de la technique résultante est justifiée par des arguments mathématiques et est vérifiée sur un problème de grande taille. La convergence finie vers une paire de solutions primale et duale est

assurée par l'utilisation de régions de confiance de pleine dimension. Nous montrons également qu'il est possible d'enrichir les directions fournies sans compromettre la convergence de l'algorithme.

La stabilisation par des inégalités valides sur le polyèdre des solutions optimales duales est décrite au chapitre 4. Deux types d'inégalités sont définis : inégalités valides sur le polyèdre optimal dual appelées *Inégalités Valides Duales-Optimales* et inégalités valides sur un sous-ensemble de ce polyèdre appelées *Inégalités Partiellement Valides Duales-Optimales*. Ces deux types traduisent en fait la qualité de l'information disponible sur l'optimalité duale. Les deux types d'inégalités valides préservent l'optimalité duale, mais la réalisabilité primale peut être compromise vu que des variables sont ajoutées à la formulation du primal. Nous proposons un moyen général pour retrouver la réalisabilité (et l'optimalité) primale. Nous obtenons ainsi une méthode en deux phases. Dans le cas des inégalités valides duales-optimales, nous profitons du fait que ces contraintes sont vérifiées par toutes les solutions optimales duales pour obtenir une paire de solutions primale et duale en une seule étape.

Le chapitre 5 est consacré à l'application de la stabilisation par une fonction de pénalité linéaire par morceaux au problème de tournées de véhicules multi-dépôt (*MDVSP*) formulé comme un problème de partitionnement. Nous montrons d'abord comment cette formulation est obtenue à partir de la formulation multiflots en appliquant le principe de décomposition des problèmes en nombres entiers vu au chapitre 2. Nous proposons une procédure d'initialisation qui fournit des estimations pour les variables duales, une borne inférieure sur la valeur optimale de la relaxation linéaire, une solution primale entière ainsi qu'une borne supérieure sur la valeur optimale du problème en nombres entiers. Nous présentons des résultats numériques sur des problèmes de moyenne et grande tailles. Nous testons plusieurs types de stratégie

de mise à jour de la fonction de pénalité, quelques astuces d'implantations et des directions de déplacement supplémentaire des centres de stabilité. Les résultats sont très satisfaisants.

Dans le chapitre 6, nous montrons que la technique d'accélération de la résolution du problème de découpe binaire (*BCSP*) qui consiste à agréger les contraintes de couverture des items de même longueur est en fait un cas particulier de la stabilisation par des inégalités partiellement valides duales-optimales. Nous testons deux façons d'implanter la stabilisation afin de montrer l'apport intrinsèque de cette dernière.

La stabilisation du problème de découpe unidimensionnelle (*CSP*) est traitée au chapitre 7. Nous montrons que des coupes duales déjà utilisées dans la littérature sont en fait des inégalités valides duales-optimales pour *CSP*. Nous justifions également l'efficacité de l'utilisation de ces inégalités dans le cadre de la génération de colonnes. Les résultats montrent une amélioration impressionnante du comportement de la génération de colonnes. Nous testons également une approche de stabilisation autour d'estimations initiales des valeurs duales calculées à partir des poids des items. Les résultats confirment l'apport considérable de la stabilisation de l'évolution des variables duales.

Abstract

This thesis mainly studies stabilization techniques of the column generation and Kelley algorithms. Unified frameworks have been recently proposed for methods that follow the bundle algorithm scheme. We propose a unified framework for stabilization approaches that follow the proximal point algorithm scheme. Even though this work has been independently done, our idea fits in the works of Frangioni [20] and has similarities with the work of Neame [67]. We describe two algorithms for column generation stabilization. These algorithms are then tested on well known optimization problems.

We describe the two widely used large scale problem solution approaches, D-W decomposition/column generation and lagrangean relaxation/non-differentiable optimization methods, in chapter 1. The former solves what we'll call the primal problem and the latter solves its dual. We give a comprehensive synthesis of the recent results related to these approaches. Primal and dual degeneracy affect the behaviour of these methods and this causes instability to the (primal and dual) convergence process. Many techniques have been developed in order to make these methods more efficient especially for large problems. Such techniques are called stabilization methods. We distinguish two classes : bundle type and proximal type stabilization approaches. Methods of the former class follow the bundle algorithm scheme while using more general penalty functions (see Neame 1999 and Frangioni 2002). Methods of the latter class follow the proximal point algorithm (Rockafellar 76) while using more general penalty functions. A survey of recently developed unified approaches is presented at the end of the chapter.

A unified approach for proximal type stabilization methods is introduced in chap-

ter 2. We generalize the idea of Kim & al. (1994) which is itself an extension of the proximal point algorithm of Rockafellar (1976). The method is presented in the case of the maximization of a concave function. We use functions that take the value 0 over a convex closed set containing the stability center (trust region) and are negative elsewhere. Convergence is studied in the case of full-dimension trust regions containing stability centers in their interior. This condition allows proving finite convergence even for nonpolyhedral functions. However, this requires the stabilized problems to be solved to optimality at every major iteration. We also motivate our choice of a piecewise linear norm penalty function for column generation algorithm stabilization.

In chapter 3, we study a first special case of the method introduced in chapter 2. We propose a 5-pieces linear penalty function to stabilize linear programs solution by column generation. The penalty is added to the dual objective function to give the stabilized dual problem. Its dual is called the stabilized primal. The algorithm uses two types of iterations. Each major iteration corresponds to the entire solution process of a stabilized problem by a cutting plane algorithm. The iterations of this algorithm are called minor iterations. The use of a penalty function aims to “restrict” the set of possible multipliers and to ensure an increase in the dual lower bound at the end of every major iteration. From a primal point of view, this corresponds to penalizing the lagrangean for multipliers in the trust region.

We first show that the knowledge of an optimal multiplier can be used to obtain a primal (basic) optimal solution by solving an easy problem by column generation. We give mathematical arguments in favour of the efficiency of this technique. Results obtained on a large MDVSP instance confirm this efficiency. Finite convergence to a pair of primal-dual solutions is ensured by using trust regions that are boxes of width greater than a fixed positive value. We also prove that it's possible to enhance

the computed ascent directions while preserving convergence.

In chapter 4, we study another special case of the unified framework. We use penalty functions that are defined by two types of polyhedral inequalities. Dual-optimal partial valid inequalities are satisfied by a subset of dual optimal solutions (or at least, this is all what we would be able to prove). Dual-optimal valid inequalities are satisfied by every dual optimal solution. The two types of inequalities preserve dual optimality, but primal feasibility can be lost since variables are added to the primal. We propose a general way to recover primal feasibility. This leads to a 2-stage method. In the case of dual-optimal valid inequalities, we take advantage from the fact that every dual optimal solution satisfies the added inequalities to obtain a pair of primal and dual optimal solutions by slightly modifying the stabilized problems.

In chapter 5, we apply the piecewise linear penalty function stabilization algorithm developed in chapter 3 to the linear relaxation of the Multidepot Vehicle Scheduling Problem (*MDVSP*) formulated as a set partitioning problem. We first show how to obtain this formulation from a multiflot formulation using integer-decomposition principle (see chapter 2). We also suggest an initializing procedure that gives a lower bound on the linear relaxation optimal value, initial dual variables estimations, a primal integer solution and an upper bound on the integer problem optimal value. Numerical tests are performed using 6 problems (2×400 tasks-problems, 2×800 tasks-problems and 2×1000 tasks-problems). Results are compared to those obtained by standard column generation. Moreover, several acceleration techniques are tested. Results were very satisfactory.

Stabilization of the Binary Cutting Stock Problem (*BCSP*) using dual-optimal partial valid inequalities occurs in chapter 6. We first prove that the famous aggregation technique of items of same weight is equivalent to add, to the dual formulation,

constraints saying that dual variables corresponding to items of same weight have to be equal. We also show that there always exists a dual optimal solution satisfying this property. The aggregation technique being an implicit way to implement stabilization constraints, we consider an explicit way to impose the stabilization constraints. This lets us point out the real contribution of the stabilization to the efficiency of the solution process.

In chapter 7, we apply our stabilization techniques to the Cutting Stock Problem (*CSP*). We prove that dual cuts already used in the literature are dual-optimal valid inequalities for *CSP*. We discuss efficiency issues for primal problem solving by column generation and present computational results that confirm the power of these valid inequalities. We also used a stabilization technique with a piecewise linear penalty function defined around dual values computed directly from item weights. We obtain very good results.

Table des matières

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	x
TABLE DES MATIÈRES	xiv
LISTE DES TABLEAUX	xxiii
LISTE DES FIGURES	xxiv
INTRODUCTION	1
 I Résolution des problèmes de grande taille	 7
 CHAPITRE 1 : RÉOLUTION DES PROBLÈMES DE GRANDE TAIL-	
LE	8
1.1 Génération de colonnes et décomposition de Dantzig-Wolfe	10
1.1.1 Génération de colonnes	10

1.1.2	Principe de décomposition de Dantzig-Wolfe	13
1.1.3	Décomposition des problèmes à variables entières	18
1.1.4	Remarques sur l'implantation	23
1.1.5	Notes bibliographiques	24
1.2	Relaxation lagrangienne et méthodes d'optimisation non-différentiable	24
1.2.1	Relaxation lagrangienne	24
1.2.2	Méthodes du sous-gradient	29
1.2.3	Méthode des plans coupants de Kelley	32
1.2.4	Notes bibliographiques	34
1.3	Difficultés et instabilité	34
1.4	Méthodes de résolution stabilisées	37
1.4.1	Méthodes de centres	37
1.4.2	Méthodes de faisceaux	38
1.4.3	Approches de stabilisation récentes	42
1.4.4	Approches unifiées	46

1.5 Conclusion	48
II Stabilisation des méthodes des plans coupants	49
CHAPITRE 2 : UNE MÉTHODE PROXIMALE POUR L'OPTIMI- SATION CONVEXE NON-DIFFÉRENTIABLE	50
2.1 Principe	51
2.2 Pénalisation du problème dual	53
2.3 Étude de la convergence	54
2.3.1 Algorithme	58
2.3.2 Remarques	61
2.4 Maximisation d'une fonction concave affine par morceaux	63
2.5 Conclusion	65
CHAPITRE 3 : STABILISATION PAR UNE FONCTION DE PÉ- NALITÉ LINÉAIRE PAR MORCEAUX	67
3.1 Pénalité linéaire à cinq morceaux	70
3.1.1 Fonction de pénalité	71

3.1.2	Formulation des problèmes stabilisés	73
3.2	Propriétés des problèmes stabilisés	75
3.2.1	Analyse duale	75
3.2.2	Analyse primale	77
3.2.3	Remarques importantes	80
3.3	Cas où une solution duale optimale est connue	83
3.4	Analyse des solutions des problèmes stabilisés dans le cas général . .	87
3.5	Algorithme	91
3.5.1	Étude de la convergence	93
3.5.2	Stratégies de mise à jour de la fonction de pénalité	95
3.5.3	Quelques aspects de l'implantation	97
3.6	Conclusion	99
 CHAPITRE 4 : INÉGALITÉS VALIDES SUR LE POLYÈDRE OPTIMAL DUAL		101
4.1	Inégalités partiellement valides duales-optimales	103
4.1.1	Effets sur les problèmes originaux	104

4.1.2	Étude de la convergence	107
4.2	Inégalités valides duales-optimales	110
4.2.1	Effets sur les problèmes originaux	111
4.2.2	Étude de la convergence	113
4.3	Remarques générales	117
4.4	Conclusion	118
III	Applications	120
 CHAPITRE 5 : STABILISATION DU PROBLÈME DE TOURNÉES		
	DE VÉHICULES MULTI-DÉPÔTS	121
5.1	Formulation	122
5.2	Résolution par génération de colonnes	125
5.2.1	Décomposition en nombres entiers	126
5.2.2	Expérimentation	128
5.3	Stabilisation	130
5.3.1	Initialisation	132

5.3.2	Calcul d'une borne inférieure après une itération mineure . .	135
5.3.3	Choix de la fonction de pénalité	136
5.3.4	Quelques astuces d'implantation	138
5.4	Résultats numériques	139
5.4.1	Stratégies de mise à jour	140
5.4.2	Effet de l'utilisation des astuces	142
5.4.3	Enrichissement des directions de déplacement	144
5.4.4	Utilisation simultanée de plusieurs techniques d'accélération .	146
5.5	Conclusion	148

CHAPITRE 6 : STABILISATION DU PROBLÈME DE DÉCOUPE

BINAIRE	151
6.1	Résolution par génération de colonnes	152
6.2	Stabilisation	155
6.2.1	Implantation explicite de la stabilisation	156
6.2.2	Imposition implicite de la stabilisation	159
6.3	Résultats numériques	160

6.3.1	Imposition implicite de la stabilisation	160
6.3.2	Imposition explicite de la stabilisation	163
6.4	Conclusion	165
 CHAPITRE 7 : STABILISATION DU PROBLÈME DE DÉCOUPE UNIDIMENSIONNELLE		167
7.1	Résolution par génération de colonnes	170
7.2	Stabilisation	172
7.2.1	Effet sur la formulation du dual	175
7.2.2	Effet sur la formulation du primal	178
7.2.3	Remarques sur l'implantation	179
7.3	Résultats numériques	182
7.4	Conclusion	190
 CHAPITRE 8 : CONCLUSION		192
8.1	Contributions	193
8.2	Perspectives et travaux futurs	197
8.2.1	Approche unifiée	197

8.2.2	Utilisation d'un multiplicateur optimal dual pour le calcul d'une solution optimale primale de base	197
8.2.3	Stabilisation par une fonction de pénalité linéaire par mor- ceaux : utilisation pour la résolution des problèmes en nombres entiers	200
8.2.4	Inégalités valides duales-optimales	201
BIBLIOGRAPHIE		202

Liste des tableaux

3.1	Stabilisation : résultats obtenus quand π^* est connu.	86
5.1	<i>MDVSP</i> : caractéristiques des instances résolues.	129
5.2	<i>MDVSP</i> : résolution par génération de colonnes.	130
5.3	<i>MDVSP</i> : stabilisation par une fonction de pénalité linéaire par morceaux avec 3 stratégies de mise à jour différentes.	141
5.4	<i>MDVSP</i> : utilisation des astuces dans l'implantation de la stabilisation par une fonction de pénalité linéaire par morceaux.	143
5.5	<i>MDVSP</i> : déplacement supplémentaire des centres de stabilité.	145
5.6	<i>MDVSP</i> : utilisation simultanée de plusieurs techniques d'accélération dans l'implantation de la stabilisation.	147
6.1	<i>BCSP</i> : Caractéristiques des problèmes résolus.	154
6.2	<i>BCSP</i> : résolution par génération de colonnes	155
6.3	<i>BCSP</i> : caractéristiques des problèmes avec l'utilisation implicite des inégalités partiellement valides duales-optimales.	161

6.4	<i>BCSP</i> : utilisation implicite des inégalités partiellement valides duales-optimales.	162
6.5	<i>BCSP</i> : utilisation explicite des inégalités partiellement valides duales-optimales.	164
7.1	<i>CSP</i> : caractéristiques des problèmes.	171
7.2	<i>CSP</i> : Résolution par génération de colonnes.	172
7.3	<i>CSP</i> : stabilisation par des inégalités valides duales-optimales.	183
7.4	<i>CSP</i> : stabilisation par une fonction de pénalité linéaire par morceaux.	187
7.5	<i>CSP</i> : stabilisation simultanée par des inégalités valides duales-optimales et une pénalité linéaire par morceaux.	188
7.6	<i>CSP</i> : résumé des résultats de l'utilisation de la stabilisation.	189
8.1	Résultats préliminaires du <i>Crossover</i>	199

Liste des figures

1.1	Comportement typique de l'algorithme de génération de colonnes. . .	36
2.1	Stabilisation : Illustration du type de fonction de pénalité utilisée. . .	65
3.1	Évolution de l'objectif primal et des bornes inférieures.	71
3.2	Fonction de pénalité linéaire à cinq morceaux.	73
3.3	Pénalité ajoutée au lagrangien de (P) autour de $\pi \in]\delta_-, \delta_+[$	81
3.4	Quantité ajoutée à $b^T \pi^*$ en fonction de la réalisabilité avec (> 0) ou sans stabilisation $(= 0)$ pour les x tels que $\pi^{*T}(b - Ax) = 0$ et $c^T x = b^T \pi^*$	82
3.5	Effet de la non-réalisabilité sur l'objectif du primal $c^T x$ dans le cas où $\gamma_- > 0, \delta_- > 0, \delta_+ > 0$ et $\gamma_+ > 0$	83
4.1	Restriction du domaine réalisable dual par des contraintes d'égalité valides sur un sous-ensemble du polyèdre optimal dual.	105
4.2	Restriction du polyèdre optimal dual par des contraintes d'égalité valides sur un sous-ensemble des solutions optimales duales.	106
4.3	Restriction du domaine réalisable à l'aide une inégalité valides sur le polyèdre optimal dual.	113

Introduction

La résolution des problèmes d'optimisation de grande taille qui proviennent des applications industrielles a souvent été à l'origine du développement de plusieurs méthodes d'optimisation. Ces méthodes sont souvent basées sur des techniques simples, astucieuses mais efficaces. Parmi les méthodes qui furent utilisées avec succès dans de nombreuses applications, l'algorithme de génération de colonnes [9, 28, 53] et l'algorithme de Kelley [42] sont deux formes duales l'une de l'autre d'une même idée. Quoique ces algorithmes soient basés sur des concepts linéaires, leur apport a été intéressant même pour des problèmes non-linéaires. Leur simplicité joue un rôle très important dans leur efficacité pratique.

Avec l'accroissement de la taille des entreprises et par conséquent de la taille des problèmes considérés, cette méthode a rencontré certaines difficultés. Quelques-unes parmi ces difficultés ont pu être surmontées grâce à la puissance de plus en plus grande des calculateurs. Cependant, les difficultés liées aux aspects algorithmiques et mathématiques n'ont pu être résolues par cette puissance. Afin de pousser encore plus loin ces limites, une approche mathématique basée sur des techniques d'optimisation est nécessaire.

Les problèmes de convergence de l'algorithme de génération de colonnes sont dues essentiellement à la dégénérescence primale, à la qualité des colonnes générées, à l'oscillation des valeurs duales durant le processus de génération de colonnes et surtout à la dégénérescence duale. Le comportement pathologique causé par ces phénomènes, qui se voit par exemple sur l'évolution des valeurs des variables duales d'une itération à l'autre, est appelé instabilité. C'est la raison qui a fait que la plupart des approches qui visent à remédier à ce comportement sont dites de stabilisation.

Ces approches consistent généralement à modifier le problème dual afin de guider l'évolution des multiplicateurs, et par conséquent de forcer la génération de certaines colonnes particulières qui seraient de meilleure qualité. Cette modification tend aussi à restreindre le dual de façon à éviter des déplacements inutiles dus au manque d'information sur le problème dual autour des multiplicateurs courants. C'est ainsi que plusieurs techniques de stabilisation de l'algorithme de génération de colonnes basées sur des idées variées furent proposées. Le travail décrit dans cette thèse va dans cette même direction. Nous proposons une approche unifiée de stabilisation pour l'algorithme de génération de colonnes basée sur les concepts de région de confiance (*Trust Region*) et de pénalisation. Nous nous concentrons sur deux idées principales : la première basée sur une pénalisation linéaire par morceaux de l'objectif dual et la deuxième sur la restriction du domaine réalisable dual par deux types d'inégalités valides sur le polyèdre des solutions duales optimales. Les deux approches développées seront appliquées à trois problèmes connus.

La thèse est organisée en trois parties. La première partie, composée par le chapitre 1, présente les méthodes classiques de résolution des problèmes de grande taille. La deuxième partie est consacrée aux développements théoriques des méthodes proposées dans les chapitres 2, 3 et 4. La dernière partie est quant à elle consacrée aux applications des méthodes développées. Chacun des chapitres 5, 6 et 7 présente l'application d'une approche de stabilisation à un problème connu.

Nous consacrons le premier chapitre à la présentation des méthodes de résolution des problèmes de grande taille. Nous y présentons une synthèse des principales méthodes de résolution des problèmes (convexes) de grande taille. Dans la première partie, nous présentons les méthodes qu'on peut qualifier de primales vu qu'elles consistent à combiner le principe de décomposition de Dantzig-Wolfe avec l'algorithme de génération de colonnes. Nous faisons également une revue des dévelop-

pements dans la généralisation du principe de décomposition de Dantzig-Wolfe aux problèmes à variables entières et nous présentons les principaux résultats qui y sont liés. Dans la deuxième partie, nous présentons les méthodes duales basées sur la relaxation lagrangienne et les méthodes classiques d'optimisation non-différentiable. Puis, nous montrons les difficultés de ces approches classiques et les différents aspects d'instabilité dans leur comportement. Ensuite, nous faisons une revue des principales approches de stabilisation récentes. Nous distinguons deux classes : les méthodes du type faisceaux qui suivent le schéma de l'algorithme de faisceaux [45, 37] (*Bundle Algorithm*) et les méthodes du type proximal qui suivent le schéma de l'algorithme du point proximal [76] (*Proximal Point Algorithm*). Nous faisons également un survol des approches unifiées développées récemment.

Dans le deuxième chapitre, nous introduisons une méthode proximale pour la résolution des problèmes d'optimisation convexe non-différentiable. Dans le cas où les sous-problèmes sont résolus par l'algorithme de Kelley, on obtient une approche unifiée pour les méthodes de stabilisation de type proximal. L'approche est basée sur le principe de région de confiance autour d'un centre de stabilité à l'extérieur de laquelle une pénalité est appliquée. Les résultats de convergence de l'algorithme général sont présentés pour une région de confiance de pleine dimension contenant le centre de stabilité dans son intérieur. Cette hypothèse permet d'assurer la convergence finie même dans le cas des fonctions non-polyédrales. Cependant, les sous-problèmes doivent être résolus exactement à chaque itération majeure. Nous présenterons des cas particuliers de la fonction de pénalité et nous justifierons notre choix d'une pénalité linéaire par morceaux pour la stabilisation de l'algorithme de génération de colonnes.

Le troisième chapitre est consacré à l'étude de la stabilisation des algorithmes de génération de colonnes et de Kelley par une pénalité linéaire par morceaux. La fonc-

tion de pénalité comporte cinq morceaux et présente la particularité que la région de confiance est un pavé non-vide de pleine dimension contenant le centre de stabilité dans son intérieur. Cette propriété permet de montrer un résultat fort intéressant sous des hypothèses faibles. Ce résultat affirme que si une solution optimale duale est connue, la résolution des problèmes stabilisés autour de cette solution assure l'obtention d'une solution optimale primale de base. Nous présentons ensuite des arguments mathématiques qui plaident en faveur de l'efficacité de cette résolution. Celle-ci est vérifiée sur une instance de grande taille du problème de tournées de véhicules multi-dépôts. Ensuite, nous établissons les liens entre les problèmes originaux et les problèmes stabilisés avant de présenter l'algorithme. La convergence de l'algorithme vers une paire de solutions optimales primale et duale est établie. En particulier, l'obtention d'une solution primale optimale de base est garantie par la propriété évoquée ci-dessus. Nous donnons également l'interprétation de la pénalité du point de vue primal en faisant le lien entre notre approche et la relaxation lagrangienne et discutons de quelques aspects liés à l'implantation de l'algorithme.

Un autre cas particulier de la fonction de pénalité présentée au chapitre 2 est étudié dans le quatrième chapitre. Nous y introduisons une notion de validité pour des coupes duales (*Dual Cuts*) basée sur la validité de ces contraintes pour le polyèdre optimal dual. Deux types de ces inégalités sont définis : inégalités valides sur le polyèdre optimal dual appelées *Inégalités Valides Duales-Optimales* (*Dual-Optimal Valid Inequalities*) et inégalités valides sur un sous-ensemble de ce polyèdre appelées *Inégalités Partiellement Valides Duales-Optimales* (*Dual-Optimal Partial Valid Inequalities*). Ces deux types traduisent en fait la qualité de l'information disponible sur l'optimalité duale. Nous analysons l'effet de l'utilisation des contraintes résultantes sur les problèmes primal et dual et nous montrons comment obtenir une paire de solutions optimales primale et duale en deux phases. Quand les inégalités sont valides sur tout le polyèdre optimal dual, nous montrons comment modifier les problèmes

pour que l'obtention d'une solution optimale primale de base se fasse en une seule phase.

Dans le chapitre 5, nous appliquons la stabilisation par une fonction de pénalité linéaire par morceaux au problème de tournées de véhicules multi-dépôts (*MDVSP*). Nous donnons d'abord les formulations connues en problème de multiflots et en problème de partitionnement. Nous montrons l'équivalence entre les deux formulations en nombres entiers et entre leur relaxations linéaires en appliquant le principe de décomposition généralisé introduit au chapitre 1. Nous présentons ensuite les résultats obtenus en résolvant la relaxation linéaire du problème de partitionnement par génération de colonnes. Puis, nous montrons comment appliquer l'algorithme de stabilisation à ce problème. La procédure d'initialisation proposée permet de calculer une estimation réalisable pour les variables duales, une borne inférieure sur la valeur optimale de la relaxation linéaire, une solution primale réalisable entière et une borne supérieure sur la valeur optimale du problème en nombres entiers. Nous testons ensuite plusieurs stratégies de mise à jour de la fonction de pénalité, des astuces d'implantation et des directions de déplacement supplémentaire du centre de stabilité. Les résultats montrent l'apport considérable de l'utilisation de la stabilisation pour l'efficacité de la résolution.

Dans le chapitre 6, nous présentons une application des *inégalités partiellement valides duales-optimales*. Nous considérons le problème de découpe binaire (*BCSP*) et nous montrons qu'il existe une solution optimale duale pour laquelle deux items de longueur identique ont leurs variables duales correspondantes égales. Ces contraintes ont déjà été utilisées dans la littérature par plusieurs auteurs sous une forme implicite. Les contraintes correspondant à des items de longueur identique sont agrégées en une seule contrainte dont le second membre n'est plus égal à 1. Ceci réduit considérablement la taille des problèmes résolus et les rend très faciles à résoudre.

Dans le but de montrer l'apport intrinsèque de la stabilisation, nous avons imposé les contraintes explicitement dans les problèmes à résoudre. Les résultats viennent confirmer la pertinence de l'utilisation de l'information duale pour l'accélération et la stabilisation de la résolution par génération de colonnes.

Le chapitre 7 est consacré à l'application des inégalités valides du deuxième type au problème de découpe unidimensionnelle (*CSP*). Nous montrons que des coupes duales qui ont déjà été utilisées pour *CSP* sont des inégalités valides sur le domaine optimal dual (*inégalités valides duales-optimales*). Nous montrons aussi à l'aide d'un exemple que ces inégalités coupent effectivement le domaine dual réalisable. Nous expliquons également comment les colonnes correspondant à ces contraintes dans le primal permettent d'accélérer sa résolution. Les tests sont réalisés sans l'utilisation d'heuristiques primales connues pour fournir des solutions qui à défaut d'être optimales, sont très proches de l'optimalité en particulier pour les problèmes réputés difficiles. Les résultats viennent confirmer que l'utilisation de ces inégalités valides est d'un apport considérable pour l'efficacité de l'algorithme de génération de colonnes. Nous testons également la stabilisation par une fonction de pénalité linéaire par morceaux définies autour de valeurs initiales calculées à partir des poids des items. Sachant que ces estimations sont généralement proches d'une solution optimale duale surtout pour les problèmes les plus difficiles (ayant une perte très faible), les résultats viennent confirmer que l'information duale peut être très utile pour l'efficacité de l'algorithme de génération de colonnes.

Finalement, nous résumons les contributions de notre travail ainsi que les travaux futurs qui lui sont reliés dans le chapitre 8. Nous y donnons également les conclusions que l'élaboration de ce travail nous a permis de tirer sur l'optimisation des problèmes de grande taille et les méthodes de stabilisation des algorithmes de Kelley et de génération de colonnes.

Première partie

Résolution des problèmes de grande taille

Chapitre 1

Résolution des problèmes de grande taille

La résolution des problèmes pratiques provenant de l'industrie a été l'une des principales motivations derrière le développement des techniques avancées de l'optimisation. Les problèmes de fabrication d'itinéraires de véhicules ou d'horaires du personnel reviennent souvent dans l'industrie du transport aérien, ferroviaire ou urbain [12, 14, 22, 38, 83, 97]. Les problèmes d'ordonnancement trouvent leurs applications dans la gestion des chaînes de production dans les ateliers [85, 80]. Les problèmes de logistique servent dans les applications militaires et l'optimisation des coûts dans les grandes entreprises [54]. Les domaines de l'économie, des finances et de l'environnement fournissent aussi des applications intéressantes pour l'optimisation. Le point commun entre ces applications est que les modèles correspondants fournissent des problèmes de grande taille qui se ramènent souvent à la résolution de programmes linéaires.

Ces problèmes sont souvent résolus par la technique de génération de colonnes [9, 28, 53] ou les techniques apparentées tel que la méthode de Kelley [42], l'algorithme du sous-gradient [29, 35, 77] ou d'autres méthodes d'optimisation non-différentiable [44, 36, 37, 72]. Mais, ces problèmes sont généralement obtenus à la suite de l'application des techniques de décomposition [9, 10, 11] ou de relaxation lagrangienne [21, 27, 59] aux modèles de départ. Ainsi, ces problèmes prennent des formes adéquates pour la résolution par génération de colonnes ou par des métho-

des d'optimisation non-différentiable ; d'où le lien étroit entre la décomposition et la génération de colonnes d'un côté et entre la relaxation lagrangienne et les méthodes d'optimisation non-différentiable de l'autre.

La génération de colonnes et les méthodes classiques d'optimisation non-différentiable ont affiché certaines limites suite à l'accroissement des tailles des problèmes résolus. Alors, plusieurs méthodes de résolution dites stabilisées ont vu le jour dans le but de remédier aux difficultés et défauts des méthodes classiques. D'une part, plusieurs modifications ont été proposées pour l'algorithme du sous-gradient et l'algorithme de Kelley pour donner les méthodes de faisceaux [45, 37] et les méthodes de centres [16, 60, 86, 32]. D'autre part, des algorithmes de génération de colonnes stabilisée ont été proposés. Nous reviendrons sur ces méthodes à la fin du présent chapitre.

Le contenu du chapitre constitue une synthèse des principaux développements dans les méthodes classiques de résolution des problèmes (convexes) de grande taille. La première partie est consacrée à l'approche primale qui associe la décomposition de Dantzig-Wolfe et la génération de colonnes. Nous faisons une revue des travaux généralisant ce principe aux problèmes à variables entières et nous décrivons les deux approches utilisées : convexification et discrétisation. La deuxième partie décrit l'approche duale de la première. Nous y présentons le principe de la relaxation lagrangienne et les méthodes d'optimisation non-différentiable classiques. Nous discutons ensuite du comportement de ces approches et des aspects d'instabilité qu'elles affichent. Par la suite, nous présentons les méthodes dites stabilisées. Nous décrivons le principe général des méthodes de centres et des méthodes de faisceaux avant de présenter les principales approches de stabilisation connues. Parmi ces dernières, nous distinguons deux classes : les méthodes suivant le schéma des méthodes de faisceaux et les méthodes suivant le schéma de l'algorithme du point proximal.

1.1 Génération de colonnes et décomposition de Dantzig-Wolfe

Le principe de décomposition a été introduit par Dantzig et Wolfe en 1960 [9] dans un des plus importants articles sur la programmation linéaire. Les auteurs y proposent la résolution du problème maître par génération de colonnes. Indépendamment, Gilmore et Gomory proposèrent en 1961 [28] la résolution par génération de colonnes du problème de découpe unidimensionnelle (*CSP*). Nous introduisons l'algorithme de génération de colonnes indépendamment de la décomposition de Dantzig-Wolfe. Les liens entre les deux concepts seront clairs à la fin de la présente section.

1.1.1 Génération de colonnes

On considère le problème linéaire suivant, appelé problème maître (*master problem*),

$$\begin{aligned} \text{Min } z &= \sum_{j \in P} c_j \lambda_j \\ \text{sc } \left\{ \begin{array}{l} \sum_{j \in P} M_j \lambda_j = b \\ \lambda_j \geq 0, \end{array} \right. \end{aligned} \quad (1.1)$$

où le nombre de colonnes M_j ($j \in P$) est soit très grand soit pratiquement impossible à énumérer préalablement. Lors de la résolution de ce problème par l'algorithme du simplexe, l'évaluation des coûts réduits des variables λ_j est très coûteuse en temps de calcul à cause du nombre très élevé de colonnes. Si π est le vecteur des variables duales actuelles, trouver $\min\{\bar{c}_j = c_j - \pi^T M_j, j \in P\}$ requiert $O(|P|)$ évaluations. Si l'ensemble des colonnes peut être décrit à l'aide de l'ensemble Ω et la fonction de coût $c(M)$ ($M \in \Omega$), il suffit alors de résoudre le problème d'optimisation suivant

appelé sous-problème ou oracle :

$$\begin{aligned} & \text{Min } c(M) - \pi^T M \\ & \text{sc } \{ M \in \Omega. \end{aligned} \quad (1.2)$$

L'efficacité de la résolution du sous-problème dépend de la structure de Ω et de la fonction $c(M)$. Si la fonction $c(M)$ est linéaire et Ω est un polyèdre, (1.2) est un programme linéaire et sa résolution est beaucoup moins coûteuse que l'énumération de toutes les colonnes. Si le nombre de colonnes est fini, Ω est un ensemble discret. Dans certains cas, la structure de Ω permet de résoudre efficacement (1.2) en nombres entiers. Par exemple, pour le problème de découpe unidimensionnelle, l'ensemble des colonnes est le domaine du problème de sac-à-dos (Gilmore et Gomory 61 [28]) ou du problème de plus court chemin avec contraintes de ressources (Ben Amor 97 [4]). Dans certains cas, on peut décrire les colonnes de (1.1) comme les points extrêmes d'un certain ensemble convexe S . Le sous-problème (1.2) est résolu en remplaçant Ω par S . Mais, afin d'assurer que la solution obtenue est un point extrême de S , $c(M)$ doit être concave. Ceci est bien sûr vérifié dans le cas où $c(M)$ est linéaire.

Afin d'éviter de manipuler toutes les colonnes lors de la résolution du simplexe (ou dans le cas où il n'est pas possible de connaître toutes les colonnes au préalable), on résout itérativement une suite de programmes linéaires qui ne tiennent compte que d'un sous-ensemble des colonnes. À chaque itération K , un problème MP_K , appelé problème maître restreint (*restricted master problem*) est résolu par l'algorithme du simplexe. La recherche d'une colonne à coût réduit négatif (*pricing*) se fait en résolvant le problème (1.2). Si le minimum est non-négatif (en fait égal à zéro à cause de la présence des variables de base), les coûts réduits des variables hors-base sont tous non-négatifs et la solution de MP_K est optimale pour (1.1). Si le minimum est négatif, la solution du sous-problème est une colonne à coût réduit négatif. Elle est donc ajoutée au problème maître restreint pour obtenir le problème MP_{K+1} qui sera résolu à son tour, et ainsi de suite. L'opération consistant à optimiser le problème maître restreint et à résoudre le sous-problème est appelée itération de génération

de colonnes.

Convergence : Si $|\Omega|$ est fini ou S est choisi de façon qu'il ait un nombre fini de points extrêmes et de rayons extrêmes (polyèdre), le nombre de colonnes susceptibles d'être générées par le sous-problème est fini. Comme toute colonne ajoutée au problème maître aura son coût réduit supérieur ou égal à 0 à l'optimalité du problème maître restreint, chaque colonne sera générée au plus une fois. Donc, le processus converge après un nombre fini d'itérations de génération de colonnes vers une solution optimale de (1.1).

Dans le cas où le nombre de colonnes que peut générer le sous-problème est infini, la procédure converge aussi, mais le nombre d'itérations requises peut être infini. Des schémas particuliers doivent être développés pour obtenir des solutions quasi-optimales.

Encadrement de la valeur optimale de (1.1) : Le problème maître résolu à l'itération K étant une restriction de (1.1), la valeur optimale \bar{z}_K est une borne supérieure sur la valeur optimale z^* de (1.1). De plus, si L est une borne supérieure sur la valeur de $\sum_{j \in P} \lambda_j$ à l'optimalité du problème maître (1.1) et sachant que $\bar{z}_K = \pi^T b$, on peut écrire

$$\bar{z}_K + f_{\min} L \leq z^* \leq \bar{z}_K \quad (1.3)$$

où f_{\min} est la valeur optimale du sous-problème (1.2).

Il est intéressant de noter que la borne supérieure \bar{z}_K est non-croissante alors que la borne inférieure $\pi^T b + f_{\min} L$ n'est pas nécessairement non-décroissante.

Cette borne inférieure est aussi appelée borne lagrangienne. Dans certains cas,

la borne L peut évoluer avec les itérations K . Dans le cas de CSP , l'objectif est $\sum_{j \in P} \lambda_j$; L peut ainsi être remplacée par z^* . La substitution permet d'obtenir une borne intéressante connue sous le nom de borne de Farley [4, 19].

1.1.2 Principe de décomposition de Dantzig-Wolfe

On considère le programme linéaire suivant :

$$\begin{aligned} & \text{Min } z = c^T x \\ & sc \left\{ \begin{array}{l} Ax = b \\ Cx = d \\ x \geq 0. \end{array} \right. \end{aligned} \quad (1.4)$$

Soit $S = \{x : Cx = d, x \geq 0\}$. S étant un polyèdre convexe, tout élément de S peut s'écrire comme une combinaison convexe (finie) des points extrêmes de S plus une combinaison linéaire non-négative de ses rayons extrêmes. Soit p^i ($i \in I$) et r^j ($j \in J$) respectivement, les points extrêmes et les rayons extrêmes de S (sachant que $|I|$ et $|J|$ sont finis). Tout point $x \in S$ s'écrit :

$$\begin{aligned} x &= \sum_{i \in I} \lambda_i p^i + \sum_{j \in J} \mu_j r^j \\ &\sum_{i \in I} \lambda_i = 1 \\ &\lambda \geq 0, \mu \geq 0. \end{aligned} \quad (1.5)$$

En substituant dans (1.4), et en utilisant les notations suivantes : $c_i = c^T p^i$, $M_i = A p^i$ ($i \in I$), $c_j = c^T r^j$, $M_j = A r^j$ ($j \in J$), on obtient le problème maître suivant [9, 53] :

$$\begin{aligned} & \text{Min } \sum_{i \in I} c_i \lambda_i + \sum_{j \in J} c_j \mu_j \\ & sc \left\{ \begin{array}{l} \sum_{i \in I} \lambda_i M_i + \sum_{j \in J} \mu_j M_j = b \\ \sum_{i \in I} \lambda_i = 1 \\ \lambda_i \geq 0, \mu_j \geq 0. \end{array} \right. \end{aligned} \quad (1.6)$$

Le nombre de contraintes dans le problème maître est $m + 1$, mais le nombre de colonnes est souvent très élevé (nombre de points extrêmes et de rayons extrêmes d'un polyèdre). De plus, il est pratiquement impossible de connaître tous ces points et rayons extrêmes a priori. Le problème est donc résolu par génération de colonnes.

Supposons qu'à l'itération courante de la procédure de génération de colonnes, on a résolu le problème maître restreint avec les sous-ensembles des points extrêmes \bar{I} et des rayons extrêmes \bar{J} . Soit π le vecteur des variables duales optimales correspondant au premier ensemble de contraintes dans (1.6) et α la variable duale optimale associée à la contrainte de convexité. Le sous-problème s'écrit alors

$$\begin{aligned} & \text{Min } (c^T - \pi^T A)x - \alpha \\ & \text{s.c. } \begin{cases} Cx = d \\ x \geq 0. \end{cases} \end{aligned} \quad (1.7)$$

Si le sous-problème n'est pas réalisable, il en est de même pour le problème maître et le problème original. Cette situation est détectée lors de l'initialisation de la génération de colonnes.

Si le sous-problème admet une solution optimale de valeur f_{min} , l'optimalité du problème de départ (1.4) est atteinte si $f_{min} \geq 0$. Si $f_{min} < 0$, la solution courante du problème maître restreint n'est pas optimale pour le problème maître (1.6). La solution du sous-problème, un point extrême p^* , est alors ajoutée à l'ensemble \bar{I} . La colonne ajoutée au problème maître est

$$M^* = [c^T p^* \mid (Ap^*)^T, 1]^T.$$

Si le sous-problème est non-borné, alors on a identifié un rayon extrême r^* qui est ajouté à l'ensemble \bar{J} . La colonne ajoutée au problème maître est

$$M^* = [c^T r^* \mid (Ar^*)^T, 0]^T.$$

Convergence : À l'optimalité du problème maître restreint, toutes les colonnes déjà générées ont des coûts réduits non-négatifs. Donc, une colonne ne peut être générée plus qu'une fois. L'ensemble S étant un polyèdre convexe, le nombre de points extrêmes et de rayons extrêmes est fini. Donc, le nombre d'itérations de génération de colonnes ne peut dépasser $|I| + |J|$.¹

Encadrement de la valeur optimale de (1.4) : Supposons que le sous-problème possède une solution optimale finie et que sa valeur optimale est f_{min} . On peut écrire $f_{min} = \bar{c}_{min} + \alpha$ (le minimum est atteint en un point extrême). Soit (λ^*, μ^*) une solution optimale de (1.6). En utilisant la contrainte de convexité, on obtient :

$$\bar{z}_K + \bar{c}_{min}(1 + \sum_{j \in J} \mu_j^*) + \alpha \leq z^* \leq \bar{z}_K.$$

Si L est une borne supérieure sur le nombre de colonnes à l'optimalité du problème maître, on obtient une borne inférieure semblable à celle obtenue pour la génération de colonnes (cf. 1.1.1).

Le terme $\sum_{i \in J} \mu_j^*$ est dû à la présence de colonnes qui sont des rayons extrêmes du domaine du sous-problème S . Cependant, si on considère les points extrêmes et les rayons extrêmes séparément, on obtient une meilleure borne. Soit $f_{min}^p = \bar{c}_{min}^p + \alpha$ et $f_{min}^r = \bar{c}_{min}^r$ les coûts réduits minimum des points extrêmes et des rayons extrêmes, respectivement. On peut alors écrire

$$\bar{z}_K + \bar{c}_{min}^p + f_{min}^r \sum_{j \in J} \mu_j^* + \alpha \leq z^* \leq \bar{z}_K.$$

Mais, dans le cas où le sous-problème est fini, $f_{min}^r \geq 0$ et $f_{min} = f_{min}^p$. On obtient

¹Il est possible de ne garder toutes les colonnes générées dans le problème maître restreint qu'à partir d'un certain nombre d'itérations de génération de colonnes tout en préservant la convergence de l'algorithme.

alors la relation : ²

$$\bar{z}_K + f_{\min} \leq z^* \leq \bar{z}_K. \quad (1.8)$$

Cette relation peut s'écrire sous la forme

$$0 \leq \bar{z}_K - z^* \leq -f_{\min}.$$

Ainsi, la valeur optimale du sous-problème, f_{\min} , est une mesure de la qualité de la solution actuelle. Notons enfin, que cette relation peut être étendue au cas où le sous-problème n'est pas fini, i.e. $f_{\min} = -\infty$.

Problèmes à structure bloc-angulaire : On considère un problème ayant une structure bloc-angulaire (1.9) où la matrice B est diagonale par blocs et la matrice $A = [A_1|A_2|\dots|A_Q]$.

$$\begin{aligned} & \text{Min} \sum_{q=1}^Q c_q^T x_q \\ & \text{sc} \left\{ \begin{aligned} & \sum_{q=1}^Q A_q x_q = b, \\ & B_q x_q = b_q, \quad q = 1, \dots, Q \\ & x_q \geq 0, \quad q = 1, \dots, Q. \end{aligned} \right. \end{aligned} \quad (1.9)$$

L'application du schéma de décomposition sur le problème (1.9), en définissant un seul sous-problème à l'aide de la matrice B , conduit au sous-problème séparable

$$\begin{aligned} & \text{Min} \sum_{q=1}^Q (c_q^T - \pi^T A_q) x_q - \alpha \\ & \text{sc} \left\{ \begin{aligned} & B_q x_q = b_q, \quad q = 1, \dots, Q \\ & x_q \geq 0, \quad q = 1, \dots, Q. \end{aligned} \right. \end{aligned} \quad (1.10)$$

Donc, Q problèmes sont résolus séparément, chacun conduisant à la valeur optimale f_{\min}^q atteinte aux points $p^{*q}(q = 1, \dots, Q)$:

$$f_{\min}^q = \min\{(c_q^T - \pi^T A_q)x_q - \frac{\alpha}{Q} : B_q x_q = b_q, x_q \geq 0\}. \quad (1.11)$$

²Cette relation est connue pour le cas où le domaine du sous-problème est un polytope (Lasdon 72 [53]).

Si $f_{min} = \sum_{q=1}^Q f_{min}^q < 0$, l'optimalité du problème de départ n'est pas atteinte et la colonne correspondante est ajoutée au problème maître ; celle-ci est donnée par :

$$[\sum_{q=1}^Q c_q^T p^{*q} \mid (\sum_{q=1}^Q A_q p^{*q})^T, 1]^T.$$

Si le sous-problème génère un rayon extrême, la variable μ_j correspondante n'apparaîtra pas dans la contrainte de convexité.

Si le sous-problème (1.10) est fini, l'encadrement de la valeur optimale s'exprime par :

$$\bar{z}_K + \sum_{q=1}^Q f_{min}^q \leq z^* \leq \bar{z}_K. \quad (1.12)$$

En pratique, pour la plupart des applications du principe de décomposition aux problèmes à structure bloc-angulaire, chacun des sous-problèmes est considéré séparément. On note p_q^i ($i \in I_q$) et r_q^j ($j \in J_q$) les points extrêmes et les rayons extrêmes de $S_q = \{x_q \geq 0 : B_q x_q = b_q\}$. Le problème maître contient Q contraintes de convexité, une pour chacun des sous-problèmes. Sa formulation est donnée par [53] :

$$\begin{aligned} \text{Min } z &= \sum_{q=1}^Q (\sum_{i \in I_q} c_i^q \lambda_i^q + \sum_{j \in J_q} c_j^q \mu_j^q) \\ \text{sc } \left\{ \begin{aligned} \sum_{q=1}^Q (\sum_{i \in I_q} M_i^q \lambda_i^q + \sum_{j \in J_q} M_j^q \mu_j^q) &= b, \\ \sum_{i \in I_q} \lambda_i^q &= 1, \quad q = 1, \dots, Q \\ \lambda^q \geq 0, \mu^q \geq 0, \quad q &= 1, \dots, Q \end{aligned} \right. \end{aligned} \quad (1.13)$$

L'encadrement de la solution optimale est aussi :

$$\bar{z}_K + \sum_{q=1}^Q f_{min}^q \leq z^* \leq \bar{z}_K. \quad (1.14)$$

Remarque : Les deux schémas de décomposition sont équivalents et fournissent la même borne inférieure. Mais, les procédures de résolution résultantes peuvent avoir des comportements différents en pratique. En fait, l'ajout de Q colonnes au problème maître a tendance à diminuer davantage l'objectif. De plus, le nombre total de points extrêmes et de rayons extrêmes dans la formulation utilisant un seul sous-problème est beaucoup plus élevé que dans la formulation (1.13). En effet, le nombre total de colonnes dans la première formulation est $\sum_{q=1}^Q (|I_q| + |J_q|)$ alors que dans la formulation à un seul sous-problème, il est de l'ordre $(\Pi_{q=1}^Q |I_q| + \Pi_{q=1}^Q |J_q|)$. Ceci peut avoir des conséquences d'ordre numérique importantes. La première est qu'il faudrait, normalement, moins de colonnes générées pour la formulation (1.13) pour atteindre l'optimalité. De plus, la présence de colonnes correspondant chacune à un sous-problème permet au problème maître restreint de les combiner de plusieurs manières différentes. Cette flexibilité permet de prendre en compte implicitement plusieurs points (ou rayons) extrêmes de S sans qu'ils soient explicitement générés.

1.1.3 Décomposition des problèmes à variables entières

L'application de la relaxation lagrangienne à la programmation linéaire en nombres entiers a permis d'obtenir des bornes meilleures que celles fournies par l'application de la décomposition de Dantzig-Wolfe à la relaxation linéaire du problème. Cependant, le principe de décomposition est souvent utilisé en enrichissant le sous-problème par les contraintes d'intégrité ce qui permet d'obtenir les mêmes bornes que la relaxation lagrangienne. Des présentations de schémas de décomposition pour les problèmes à variables entières sont faites dans Vanderbeck 2000 [92] et Ville-neuve 99 [93]. Les deux principes sous-jacents ont déjà été utilisés pour plusieurs types de problèmes. Desrosiers *et al.* 95 [14] ont utilisé une convexification du domaine de sous-problème (plus court chemin avec contraintes de ressources) dans la

décomposition du problème de tournées de véhicules avec fenêtres de temps. Le principe de discrétisation a été utilisé par Ben Amor 97 [4] pour la décomposition du problème de découpe unidimensionnelle formulé comme un problème de tournées de véhicules avec contraintes de capacité. Ben Amor 97 [4] a également utilisé la convexification du domaine du sous-problème (sac-à-dos) afin de montrer l'équivalence entre la première formulation du problème de découpe unidimensionnelle (*CSP*) (Kantorovich 39 [40]) et la formulation de Gilmore et Gomory (Gilmore et Gomory 61 [28]).

Considérons le problème à variables entières suivant :

$$\begin{array}{ll} \text{Min} & c^T x \\ \text{sc} & \left\{ \begin{array}{l} Ax = b \\ x \in S \end{array} \right. \end{array} \quad (1.15)$$

où $S = E \cap \mathbb{Z}_+^n$ avec $E \subset \mathbb{R}^n$. Les résultats connus concernent les cas où $|S|$ est fini ou E est un polyèdre. Nous présentons les deux approches possibles : la convexification de S et la discrétisation de S .

Approche de convexification :

Supposons que S peut être écrit $S = E \cap \mathbb{Z}_+^n$ où E est un polyèdre. Cette condition englobe des cas où $S = F \cap \mathbb{Z}_+^n$ avec F convexe non-polyédral, mais n'englobe pas tous les cas où $|S|$ est fini. Par exemple, si S est formé par l'union de deux pavés disjoints de \mathbb{Z}^n . Les points extrêmes de $\text{conv}(S)$ sont des points de S et les rayons extrêmes peuvent être choisis entiers. Le problème (1.15) peut être reformulé comme suit :

$$\begin{array}{ll} \text{Min} & c^T x \\ \text{sc} & \left\{ \begin{array}{l} Ax = b \\ x \in \text{conv}(S) \\ x \in \mathbb{Z}_+^n. \end{array} \right. \end{array} \quad (1.16)$$

La présence de la dernière contrainte assure que les deux problèmes ont le même ensemble de solutions. L'approche de décomposition par convexification consiste à appliquer le principe de décomposition de Dantzig-Wolfe à la relaxation linéaire de la formulation (1.16). Si π est le multiplicateur généré par le problème maître restreint, le sous-problème est alors

$$\begin{aligned} & \text{Min } (c - A^T \pi)^T x \\ & \text{sc } \{x \in \text{conv}(S)\}. \end{aligned} \quad (1.17)$$

Vu que l'objectif est linéaire, une solution finie de (1.17) est atteinte en un point extrême de $\text{conv}(S)$ qui est un point de S . Soient $\{p^i : i \in I\}$ l'ensemble des points extrêmes de $\text{conv}(S)$ et $\{r^j : j \in J\}$ l'ensemble de ses rayons extrêmes (I et J sont finis). Alors, tout $x \in \text{conv}(S)$ s'écrit

$$\begin{aligned} x &= \sum_{i \in I} \lambda_i p^i + \sum_{j \in J} \mu_j r^j \\ \sum_{i \in I} \lambda_i &= 1 \\ \lambda &\geq 0, \mu \geq 0. \end{aligned}$$

En substituant dans (1.16), on obtient le problème maître

$$\begin{aligned} & \text{Min } \sum_{i \in I} d_i \lambda_i + \sum_{j \in J} d_j \lambda_j \\ & \text{sc } \left\{ \begin{array}{l} \sum_{i \in I} \lambda_i M_i + \sum_{j \in J} \mu_j M_j = b, \\ \sum_{i \in I} \lambda_i = 1, \\ \lambda \geq 0, \mu \geq 0, \\ x = \sum_{i \in I} \lambda_i p^i + \sum_{j \in J} \mu_j r^j, \\ x \in \mathbb{Z}_+^n \end{array} \right. \end{aligned} \quad (1.18)$$

Une solution entière de (1.15) ou de (1.18) est obtenue via un processus de branchement. À chaque nœud de l'arbre de branchement, la relaxation linéaire du problème maître est résolue par génération de colonnes. Au nœud 0 de l'arbre de branchement, la relaxation linéaire de (1.18) est obtenue en éliminant les contraintes d'intégrité sur les variables x et les contraintes d'identification associées. Ce problème fournit la

même borne lagrangienne connue en optimisation combinatoire et obtenue en dualisant les contraintes $Ax = b$ dans (1.15). Ceci résulte du fait que les contraintes d'intégrité sont prises en compte dans la formulation du sous-problème. Il est aussi important de noter que les contraintes d'intégrité restent toujours sur les variables originales. Il est parfois possible d'établir l'équivalence entre l'intégrité de x et l'intégrité de λ et μ . Ceci est vrai en particulier quand $S \subset \{0, 1\}^n$. Les décisions de branchement peuvent être exprimées en fonction de la variable x ou des variables λ et μ . Dans le premier cas, il est généralement facile d'imposer ces décisions dans le sous-problème et dans le problème maître. Par contre dans le deuxième cas, les décisions seront plus difficiles à imposer dans le sous-problème.

Approche de discrétisation :

Nous considérons deux cas : S fini et E est un polyèdre. Le deuxième cas prend en considération le cas où $S = F \cap \mathbb{Z}_+^n$ avec F convexe quelconque, mais peut être défini avec un polyèdre E . C'est le cas si F est un ensemble convexe borné. Le cas où E est borné est inclus dans le premier cas qui prend en compte tout type d'ensemble fini de \mathbb{Z}_+^n .

Nous considérons d'abord le cas où S est un ensemble fini. Si $|S|$ est fini, c'est-à-dire $S = \{p^i : i \in I\}$, alors $x \in S$ si et seulement si

$$\begin{aligned} x &= \sum_{i \in I} \lambda_i p^i \\ \sum_{i \in I} \lambda_i &= 1 \\ \lambda_i &\in \{0, 1\}, i \in I. \end{aligned} \tag{1.19}$$

Le cas où $S = E \cap \mathbb{Z}_+^n$ utilise un résultat de Nemhauser et Wolsey 86 [68]. Si E est un polyèdre alors il existe deux ensembles finis $Q = \{p^i : i \in I\}$ et $R = \{r^j : j \in$

$J\} \subset \mathbb{Z}_+^n$ tels que $x \in S$ si et seulement si

$$\begin{aligned} x &= \sum_{i \in I} \lambda_i p^i + \sum_{j \in J} \mu_j r^j \\ \sum_{i \in I} \lambda_i &= 1 \\ \lambda_i &\in \{0, 1\}, \mu_j \in \mathbb{Z}_+, i \in I, j \in J. \end{aligned} \quad (1.20)$$

R est l'ensemble des rayons extrêmes (entiers) de E et Q est défini à partir de l'ensemble des points extrêmes de E , $\{p^k : k \in K\}$, et de l'ensemble R par :

$$Q = \{x \in \mathbb{Z}_+^n : x = \sum_{k \in K} \lambda_k p^k + \sum_{j \in J} \mu_j r^j, \sum_{k \in K} \lambda_k = 1, \lambda_k \geq 0, 0 \leq \mu_j < 1\}.$$

En utilisant les relations (1.19) ou (1.20) dans (1.15), on obtient une formulation du type

$$\begin{aligned} &Min \sum_{i \in I} d_i \lambda_i + \sum_{j \in J} d_j \mu_j \\ &sc \left\{ \begin{aligned} &\sum_{i \in I} \lambda_i M_i + \sum_{j \in J} \mu_j M_j = b, \\ &\sum_{i \in I} \lambda_i = 1, \\ &\lambda_i \in \{0, 1\}, \mu_j \in \mathbb{Z}_+. \end{aligned} \right. \end{aligned} \quad (1.21)$$

Les formulations (1.18) et (1.21) présentent beaucoup de ressemblance, mais les ensembles I et J ne sont pas les mêmes dans les deux cas. De plus dans (1.21), les contraintes d'intégrité sont imposées aux variables λ et μ . Une solution entière est obtenue via un processus de branchement où à chaque nœud les contraintes d'intégrité dans (1.21) sont relaxées et la relaxation linéaire résultante est résolue par génération de colonnes. Au nœud 0, la formulation du sous-problème est

$$\begin{aligned} &Min (c - A^T \pi)^T x \\ &sc \{x \in S \end{aligned} \quad (1.22)$$

où π est le multiplicateur généré par la résolution du problème maître restreint. Les relaxations continues de (1.18) et (1.21) sont équivalentes puisque toute solution

optimale produite par l'un est aussi optimale pour l'autre. Les mêmes remarques faites pour l'approche de décomposition par convexification concernant les décisions de branchement s'appliquent à l'approche par discrétisation.

1.1.4 Remarques sur l'implantation

Certains sous-problèmes permettent d'obtenir une multitude de colonnes en résolvant le sous-problème. Il est donc profitable de générer plusieurs colonnes en même temps relativement au même sous-problème. Ceci permet d'enrichir le problème maître dans le but d'accélérer la convergence. Cependant, si le nombre de colonnes devient très grand, le problème maître restreint devient difficile à manipuler. Ainsi, la taille du problème maître restreint est généralement limitée. Ceci ne pose pas de problèmes de convergence en pratique. En fait, une fois que l'objectif du problème a assez diminué, le nombre de colonnes générées est de plus en plus petit et ce sont les "bonnes" colonnes qui sont générées.

Généralement, la résolution du problème maître est une sous-routine d'un processus récursif de séparation et d'évaluation. À chaque nœud de l'arbre de branchement, le problème maître est résolu par génération de colonnes. Dans certains cas, il peut être possible de n'utiliser que des bornes inférieures et supérieures fournies par cette résolution. De plus, chaque problème maître résolu doit tenir compte des contraintes de branchement. Il y a donc une part importante de modélisation dans l'obtention d'une formulation adéquate.

1.1.5 Notes bibliographiques

Les résultats de base sur la génération de colonnes et la décomposition de Dantzig-Wolfe ainsi qu'une liste exhaustive des références sur les travaux reliés à ces deux thèmes peuvent être trouvés dans Lasdon 72 [53] et Minoux 84 [65]. Une synthèse des développements récents concernant la décomposition est présentée dans Soumis 97 [82].

1.2 Relaxation lagrangienne et méthodes d'optimisation non-différentiable

La relaxation lagrangienne est pour les méthodes d'optimisation non-différentiable ce qu'est la décomposition pour la génération de colonnes. Elle consiste à reformuler un problème en une formulation adéquate pour la résolution par ces méthodes. Nous commençons la présente section par la présentation du principe de la relaxation lagrangienne [18, 2] avant de donner un aperçu des méthodes classiques d'optimisation non-différentiable.

1.2.1 Relaxation lagrangienne

Considérons le problème de programmation mathématique (LP) suivant

$$\begin{array}{ll} \text{Min} & f(x) \\ \text{sc} & \left\{ \begin{array}{l} g_i(x) \leq 0, \quad i = 1, \dots, m \\ h_j(x) = 0, \quad j = 1, \dots, p \\ x \in S, \end{array} \right. \end{array} \quad (1.23)$$

où $f, g_i (i = 1, \dots, m)$ et $h_j (j = 1, \dots, p)$ sont des fonctions définies de $\mathbb{R}^n \longrightarrow \mathbb{R}$.

Notons par $G(x)$ la fonction de $\mathbb{R}^n \longrightarrow \mathbb{R}^m$ dont les composantes sont $g_i(x)$ ($i = 1, \dots, m$) et par $H(x)$ la fonction $H(x)$ de $\mathbb{R}^n \longrightarrow \mathbb{R}^p$ dont les composantes sont $h_j(x)$ ($j = 1, \dots, p$).

À chaque contrainte explicite h_j ($j = 1, \dots, p$) et g_i $i = 1, \dots, m$, on associe deux réels v_j et $u_i \geq 0$ appelés multiplicateurs. On définit le lagrangien du problème (LP) par [23]

$$L(x; v, u) = f(x) + \sum_{j=1}^p v_j h_j(x) + \sum_{i=1}^m u_i g_i(x) : x \in S, v \in \mathbb{R}^p, u \in \mathbb{R}_+^m,$$

qu'on peut écrire sous la forme

$$L(x; v, u) = f(x) + v^T H(x) + u^T G(x) : x \in S, v \in \mathbb{R}^p, u \in \mathbb{R}_+^m.$$

On définit la fonction duale par

$$\Theta(v, u) = \min_{x \in S} L(x; v, u).$$

Le problème dual (LD) consiste alors à maximiser la fonction Θ , i.e.

$$\begin{aligned} & \max \Theta(v, u) \\ & u \geq 0, v \end{aligned}$$

Ce problème peut s'écrire

$$\max_{u \geq 0, v} \min_{x \in S} L(x; v, u). \quad (1.24)$$

Remarquons aussi que le problème primal (LP) s'écrit sous une forme semblable

$$\min_{x \in S} \max_{u \geq 0, v} L(x; v, u). \quad (1.25)$$

À partir de la définition du lagrangien, on peut affirmer que

$$\max_{u \geq 0, v} \min_{x \in S} L(x; v, u) \leq \min_{x \in S} \max_{u \geq 0, v} L(x; v, u). \quad (1.26)$$

On peut voir aussi que la valeur de la fonction $\Theta(v, u)$ est une borne inférieure sur la valeur optimale de (LP) ou encore $\forall x \in S, G(x) \leq 0, H(x) = 0, \forall u \geq 0, v \in \mathbb{R}^p$,

$$\Theta(v, u) \leq f(x).$$

Cette dernière relation établit la dualité faible entre (LP) et (LD) . La différence entre les deux valeurs optimales est appelée saut de dualité (*Duality Gap*). Il existe des conditions sous lesquelles les deux problèmes ont la même valeur optimale. La résolution de (LD) fournit alors la valeur optimale de (LP) . C'est le cas en programmation convexe différentiable sous certaines conditions de régularité [2]. Cependant, il reste une autre difficulté à surmonter. La résolution de (LD) fournit des multiplicateurs optimaux v^* et u^* et un vecteur $x(v^*, u^*)$ associé. Ce vecteur n'est généralement pas réalisable pour (LP) .

L'existence d'un point col du lagrangien signifie que les deux problèmes (LP) et (LD) ont la même valeur optimale, c'est-à-dire que le saut de dualité est nul. La résolution de (LD) fournit alors la valeur optimale de (LP) .

Notons d'abord que le lagrangien est une fonction concave (linéaire par morceaux) en v et u . La fonction $\Theta(v, u)$ est convexe. Donc, le problème dual (LD) est un problème d'optimisation convexe. Si f, g_i ($i = 1, \dots, m$) et S sont convexes et h_j ($j = 1, \dots, p$) sont affines ((LP) est un programme convexe), le lagrangien est alors une fonction convexe en x . L'évaluation de la fonction Θ revient à l'optimisation d'un problème convexe.

L'existence d'un point col de L n'est pas garantie. Dans le cas d'un problème convexe, où f, g_i ($i = 1, \dots, m$) sont différentiables, et sous certaines conditions de qualification des contraintes (e.g. condition de Slater [81]), un point de *Karush-Kuhn-Tucker* (K-K-T) de (LP) [2] est un point col du lagrangien. En optimisation non-différentiable, d'autres conditions sont requises afin d'assurer l'absence de saut

de dualité. En plus de la convexité du problème (LP), d'autres conditions telles que S borné ou l'existence d'un multiplicateur optimal vérifiant la propriété de remplissage (*Filling Property*) (Lemaréchal 2000 [59]). Heureusement, dans le cas où f, h_j, g_i sont affines ou quadratiques (ou des normes l_p), le saut de dualité est nul [59]. Dans ce qui suit, nous illustrons le principe de la relaxation lagrangienne sur un programme linéaire que nous utiliserons dans la suite de la thèse et nous établissons le lien entre la relaxation lagrangienne et le principe de décomposition.

Cas d'un programme linéaire : Soit le programme linéaire

$$\begin{aligned} & \text{Min } c^T x \\ & \text{s.c. } \begin{cases} Ax = b \\ x \geq 0. \end{cases} \end{aligned}$$

où A est une matrice $m \times n$ et $x \in \mathbb{R}^n$.

Le dual lagrangien de ce problème est le même que celui fourni par la dualité en programmation linéaire. Nous allons écrire le lagrangien sous une forme particulière qui sera utilisée plus loin dans la thèse.

Si on définit $S = \mathbb{R}_+^n$ et on associe le multiplicateur $\pi \in \mathbb{R}^m$ aux contraintes linéaires $Ax = b$, le lagrangien s'écrit pour $x \geq 0$ et $\pi \in \mathbb{R}^m$,

$$L(x; \pi) = c^T x + \pi^T (b - Ax).$$

On peut également l'exprimer de la manière suivante

$$\begin{aligned} \mathcal{L}(x, y^+, y^-; \pi) &= c^T x + \pi^T (y^+ - y^-) \\ y^+ - y^- &= b - Ax \\ x \geq 0, y^+ &\geq 0, y^- \geq 0. \end{aligned}$$

Cette nouvelle forme introduit des variables non-négatives qui mesurent la non-réalisabilité de x pour le programme linéaire défini ci-dessus.

Si x^* est une solution optimale du programme linéaire considéré et π^* est un multiplicateur optimal, alors (x^*, π^*) est un point col de L . Ce qui implique que

$$L(x^*; \pi^*) \leq L(x; \pi^*), \forall x \geq 0.$$

Sous la nouvelle forme du lagrangien, $(x^*, 0, 0; \pi^*)$ est un point col et la dernière relation s'écrit

$$\begin{aligned} \mathcal{L}(x^*, 0, 0; \pi^*) &\leq L(x, y^+, y^-; \pi^*) \\ y^+ - y^- &= b - Ax \\ x \geq 0, y^+ \geq 0, y^- &\geq 0. \end{aligned}$$

Nous reviendrons sur cette forme plus loin dans la thèse pour illustrer la stabilisation par une pénalité linéaire par morceaux.

Cas d'un problème à objectif et contraintes linéaires. Considérons le cas où f, g_i sont affines. La formulation du problème (LP) est la suivante :

$$\begin{aligned} &Min \ c^T x \\ sc \ &\begin{cases} Ax = b \\ x \in S. \end{cases} \end{aligned} \tag{1.27}$$

Le lagrangien s'écrit

$$L(x; \pi) = c^T x + \pi^T (b - Ax), x \in S.$$

Le calcul de la fonction Θ se fait à l'aide de la résolution du problème

$$\Theta(\pi) = Min \ \{(c - A^T \pi)^T x + b^T \pi : x \in S\}.$$

Ce problème est en fait le sous-problème de la génération de colonnes ou de la décomposition de Dantzig-wolfe. Ainsi, la relaxation lagrangienne et la décomposition sont deux applications d'un même principe mais de deux points de vue différents. Le principe de la relaxation lagrangienne qui ne requiert pas une structure polyédrale de S est plus général. Le principe de décomposition peut être généralisé à des cas où S n'est pas un polyèdre. Cependant, si l'objectif n'est pas linéaire et S est convexe,

la résolution du sous-problème n'assure pas la génération d'un point extrême ou d'un rayon extrême à chaque itération de génération de colonnes³.

Une des différences principales est que la relaxation lagrangienne requiert juste que l'oracle génère un sous-gradient de la fonction Θ au point dual fourni et non forcément un point extrême ou un point minimum. De plus la décomposition donne la formulation du problème maître à résoudre pour le calcul des multiplicateurs, alors que la relaxation lagrangienne requiert juste qu'un multiplicateur soit calculé sans spécifier une méthode pour le faire.

La relaxation lagrangienne a donné naissance à plusieurs méthodes d'optimisation non-différentiable. Nous décrivons les deux principales méthodes classiques : les méthodes du sous-gradients et la méthode de Kelley.

1.2.2 Méthodes du sous-gradient

Dans le but d'alléger la présentation, nous considérons la formulation suivante du problème primal

$$\begin{array}{ll} \text{Min} & f(x) \\ \text{sc} & \left\{ \begin{array}{l} g_i(x) \leq 0, \quad i = 1, \dots, m \\ x \in S. \end{array} \right. \end{array} \quad (1.28)$$

Le lagrangien correspondant est alors

$$L(x; u) = f(x) + \sum_{i=1}^m u_i g_i(x), \quad x \in S, u \geq 0.$$

On peut écrire aussi

$$L(x; u) = f(x) + u^T G(x), \quad x \in S, u \geq 0.$$

³On peut dire que du point de vue de la reformulation utilisée, la relaxation lagrangienne est une approche convexe alors que la décomposition de Dantzig-Wolfe est une approche concave. Ces deux approches, duales l'une de l'autre, coïncident bien sûr quand le problème traité est à la fois convexe et concave, c'est-à-dire linéaire.

La fonction duale est définie pour $u \geq 0$ par

$$\Theta(u) = \text{Min}_{x \in S} L(x; u).$$

Le problème dual consiste alors à maximiser la fonction Θ , i.e.

$$\begin{aligned} & \text{Max } \Theta(u) \\ & u \geq 0. \end{aligned}$$

L'algorithme du sous-gradient [94, 27, 21] est une généralisation de l'algorithme du gradient à l'optimisation non-différentiable⁴. L'algorithme est initialisé par le calcul d'un point de départ u_0 . À chaque itération k , un sur-gradient g_k de la fonction duale Θ au point u_k est généré par un oracle qui prend en entrée le point u_k . Le point suivant u_{k+1} est obtenu en se déplaçant dans la direction de g_k avec un pas t_k . C'est-à-dire que

$$u_{k+1} = u_k + t_k \frac{g_k}{\|g_k\|}.$$

Le calcul d'un sur-gradient est une fonctionnalité de l'oracle utilisé. Aucune hypothèse sur le choix ou le calcul n'est faite. Une façon classique de calculer un sur-gradient est l'évaluation de $\Theta(u_k) = \text{Min}_{x \in S} L(x; u_k) = L(x_k; u_k)$. Le sur-gradient fourni est $g_k = G(x_k)$.

Généralement, un sur-gradient g_k n'est pas une direction de montée pour Θ en u_k . Cependant, g_k est une direction de descente de la fonction distance entre u_k et un multiplicateur optimal u^* , par exemple $\frac{1}{2}\|u - u^*\|$ (Lemaréchal 2000 [59]).

L'algorithme est arrêté si la meilleure solution courante est jugée acceptable, puis renvoie cette solution. Plusieurs critères d'arrêt sont utilisés. Certains tiennent compte de la variation du vecteur u ou de l'amélioration de la valeur de $\Theta(u)$ entre deux itérations successives. Pour une précision souhaitée $\varepsilon > 0$, on teste si $\|u_{k+1} -$

⁴Nous utilisons le nom sous-gradient pour l'algorithme même si dans le cas du problème que nous considérons (maximisation d'une fonction concave), l'algorithme calcule des surgradients de la fonction Θ . Il suffit juste de se rappeler que si s est un sous-gradient de la fonction convexe f en x , alors $-s$ est un surgradient pour la fonction concave $-f$ au même point x .

$u_k|| < \varepsilon$ ou si $|\Theta(u_{k+1}) - \Theta(u_k)| < \varepsilon$. Sachant qu'un point u est optimal pour (LD) si et seulement si 0 est un sur-gradient de Θ en u , si $||g_{k+1} - g_k|| < \varepsilon$ ou $||g_k|| < \varepsilon$, la solution est jugée acceptable. D'autres critères plus généraux testent l'évolution de l'optimisation sur un certain nombre d'itérations. Par exemple, si lors des N dernières itérations la valeur de la meilleure solution n'a pas sensiblement diminuée, le processus est arrêté. Une autre approche consiste à fixer a priori le nombre total d'itérations à effectuer, et retourne à la fin la meilleure solution obtenue. Cette dernière idée profite de l'efficacité affichée par l'algorithme sur des problèmes pratiques de petite et moyenne taille.

L'efficacité de l'algorithme est très sensible au choix de la suite t_k . Plusieurs choix ont été proposés.

Pas constant : $t_k = t_0, \forall k$. C'est le choix le plus simple possible. La convergence est souvent lente et peut afficher de l'instabilité.

Méthode de la série divergente [73, 17] : $\lim_{k \rightarrow \infty} t_k = 0$ et $\sum_{k=0}^{\infty} t_k = \infty$. La convergence de t_k vers 0 assure que le pas devient de plus en plus petit pour que u_{k+1} soit plus proche de l'optimum u^* . La divergence de la série assure la convergence de l'algorithme quelque soit le point initial choisi. Cependant, ce choix risque de ralentir la convergence en pratique. En effet, il a été montré que le taux de convergence de la méthode avec un tel choix de pas n'est pas géométrique [29].

Métrique variable : Ce concept consiste à dilater l'espace dans une certaine direction ξ_k . Une direction $B_k g_k$ avec un choix spécifique du coefficient de dilatation conduit à l'algorithme de l'ellipsoïde [78, 70, 41]. Si $\xi_k = g_k - g_{k-1}$, on obtient le r-algorithme [78].

Méthode de la série convergente [77, 29] : $t_k = M\rho^k, M > 0, 0 < \rho < 1$. Ce choix assure une convergence géométrique. Mais, dans plusieurs cas la valeur de M doit être très grande et celle de ρ proche de 1 afin d'assurer la convergence

globale. Ceci ralentit la convergence sensiblement.

Méthode de relaxation de Polyak [74, 35] : Si $\bar{\Theta}$ est une estimation de la valeur optimale de $\Theta(u)$, le pas est calculé par

$$t_k = \rho \frac{\Theta(u_k) - \bar{\Theta}}{\|g_k\|}.$$

Ce choix de t_k est une version approchée d'un choix qui utilise $\Theta(u^*)$ au lieu de $\bar{\Theta}$. Comme la valeur optimale n'est pas connue en général, c'est la suite t_k décrite ci-dessus qui est utilisée quoiqu'il n'y ait pas de résultat théorique concernant la convergence de l'algorithme qui en découle. Cependant, les tests effectués ont montré que la méthode est efficace et qu'il n'est pas difficile de trouver une bonne valeur de $\bar{\Theta}$.

Plusieurs autres approches ont été développées dans le but d'accélérer la méthode du sous-gradient [78, 30, 70]. Il a également été montré (Nemirovskii 94 [69]) que la méthode possède un taux de convergence optimal pour un certain choix des pas t_k . Cependant, le comportement de la méthode en pratique s'est avéré moyen.

1.2.3 Méthode des plans coupants de Kelley

Chaque fois que l'oracle est appelé avec le multiplicateur courant u_K , il fournit la valeur de $\Theta(u_K)$ et un sous-gradient g_K de Θ en ce point. Ce qui permet d'écrire l'inégalité

$$\Theta(u) \leq \Theta(u_k) - g_k^T(u - u_k), \quad \forall u \geq 0.$$

Le membre de droite est une approximation affine de Θ qui coïncide avec Θ au point u_k . Connaissant la paire $(\Theta(u_k), g_k)$ en K points u_k ($k = 1, \dots, K$), on définit le modèle d'approximation affine par morceaux $\hat{\Theta}^K$ qui sur-estime Θ : pour $u \in \mathbb{R}^m$,

$$\Theta(u) \leq \hat{\Theta}^K(u) := \text{Max}\{\theta(u_k) + g_k^T(u - u_k) : k = 1, \dots, K\}.$$

Le prochain point u_{K+1} est calculé en maximisant $\hat{\Theta}^K$. Ceci revient à la résolution du programme linéaire suivant, appelé problème de Kelley [42],

$$\begin{aligned} \eta_K = \text{Min } \eta \\ \text{sc } \begin{cases} \Theta(u_k) + g_k^T(u - u_k) \geq \eta, & k = 1, \dots, K \\ u \geq 0. \end{cases} \end{aligned} \quad (1.29)$$

L'oracle est appelé avec le point u_{K+1} pour fournir $(\Theta(u_{K+1}), g_{K+1})$ où g_{K+1} est un sur-gradient de Θ en u_{K+1} . Le modèle est alors enrichi en ajoutant la coupe

$$\Theta(u_{K+1}) + g_{K+1}^T(u - u_{K+1}) \geq \eta.$$

Si l'oracle génère 0 comme sur-gradient, l'optimalité est atteinte.

Lors de la résolution du programme linéaire (1.29), le nombre de contraintes peut devenir très élevé, ce qui rendrait la résolution difficile. Généralement, une solution de ce problème est obtenue en résolvant son dual par génération de colonnes. La formulation de ce problème est donnée par

$$\begin{aligned} \eta_K = \text{Min } \sum_{k=1}^K \lambda_k (-g_k^T u_k + \Theta(u_k)) \\ \text{sc } \begin{cases} \sum_{k=1}^K \lambda_k = 1 \\ \sum_{k=1}^K \lambda_k g_k \leq 0 \\ \lambda_k \geq 0, \quad k = 1, \dots, K \end{cases} \end{aligned}$$

On voit ainsi le lien évident entre l'algorithme de Kelley et l'algorithme de génération de colonnes. Les sur-gradients générés à chaque itération correspondent aux colonnes générées par le sous-problème de génération de colonnes.

Enfin, si la variable u est contrainte à être dans un certain domaine D , la formulation du problème de Kelley nécessite une approximation de ce domaine. Du point de vue primal, ceci se traduira par l'ajout de colonnes qui ne contribueront pas aux contraintes de convexité.

Convergence. La convergence globale de la méthode de Kelley est assurée si le problème à résoudre est convexe. Cependant, elle possède un taux de convergence au pire cas désastreux. Nimerovski 94 [69] a montré que dans le pire cas, pour obtenir une solution avec une précision relative ε , et si m est le nombre de contraintes explicites dans le primal, le nombre d'itérations requis est de l'ordre

$$O(1/\varepsilon^{\frac{m-1}{2}}).$$

Cependant, en pratique cette méthode présente un taux de convergence moyen nettement meilleur que ne laisse prévoir le résultat ci-dessus, surtout dans le cas où la fonction objectif est affine par morceaux. Les problèmes d'instabilité et de dégénérescence demeurent l'handicap essentiel de la méthode pour les problèmes de grande taille.

1.2.4 Notes bibliographiques

Une étude exhaustive de la minimisation des fonctions convexes y compris les méthodes de descente, les méthodes du sous-gradient, l'algorithme de Kelley et les méthodes de faisceaux sous leurs formes primale et duale est présentée dans Hiriart-Urruty et Lemaréchal 91 [36, 37]. Lemaréchal 2000 [59] présente une synthèse sur la relaxation lagrangienne, les méthodes des sous-gradients, les conditions d'optimalité primales et duales et le lien avec la génération de colonnes. Une synthèse des développements reliés à la méthode du centre analytique est donnée dans Goffin et Vial 99 [32].

1.3 Difficultés et instabilité

Nous avons déjà vu que l'algorithme du sous-gradient présente l'inconvénient que ce n'est pas un algorithme de descente vu qu'un surgradient n'est pas forcément une

direction de montée pour la fonction Θ au point courant. Cependant, c'est une direction de descente pour la fonction qui mesure la distance entre un multiplicateur u et un multiplicateur optimal u^* , $\frac{1}{2}\|u - u^*\|^2$ (Lemaréchal 2000 [59]). Ceci assure que pour un pas de déplacement assez petit, u_{K+1} est plus proche de u^* que u_K . Pour profiter de cette propriété, les pas de déplacement doivent être suffisamment petits ce qui peut compromettre sérieusement l'efficacité de l'algorithme. La question de trouver une solution optimale primale une fois l'algorithme arrêté revient généralement à résoudre le problème primal en l'enrichissant par les colonnes déjà générées.

L'une des caractéristiques principales du comportement de l'algorithme de génération de colonnes est la lenteur de la convergence à la fin de l'algorithme. La figure 1.1 montre un comportement typique de l'algorithme. Elle représente la valeur de l'objectif en fonction du nombre d'itérations. On y voit clairement l'effet de queue pour les dernières itérations. Le nombre d'itérations nécessaires pour une amélioration minime de l'objectif est très élevé. Ceci est dû essentiellement à la dégénérescence du problème primal.

Un autre aspect du comportement de l'algorithme de génération de colonnes est les oscillations fréquentes et grandes observées dans l'évolution des valeurs des variables duales. Ceci traduit des déplacements de grande amplitude dans l'espace dual. Ces déplacements sont généralement inutiles vu que le modèle utilisé autour de u_K n'est aucunement précis loin de ce point. Donc, un déplacement vers u_{K+1} qui est très loin du point courant serait inutile et même nuisible pour la convergence. L'idée qui consiste à restreindre la résolution à un domaine près du point courant est derrière le développement des méthodes de faisceaux [37] et d'autres approches de stabilisation. L'utilisation d'une pénalité permet entre autre de limiter les déplacements possibles du prochain multiplicateur par rapport à u_K et de corriger la valeur du modèle $\hat{\Theta}^K(u)$ loin de u_K .

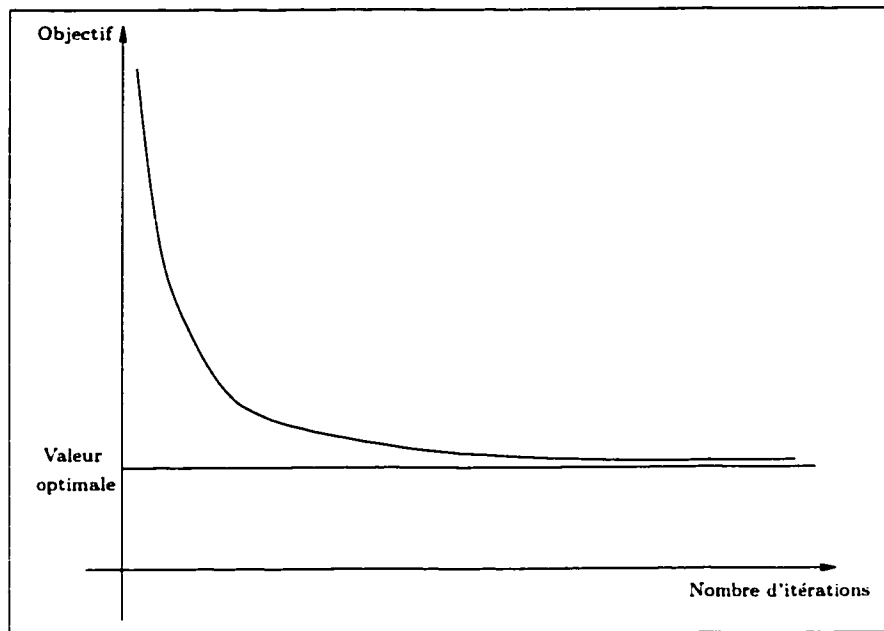


Figure 1.1 – Comportement typique de l'algorithme de génération de colonnes.

La non-croissance de la borne inférieure fournie par le point dual courant est également un défaut de la génération de colonnes. En fait, la borne supérieure primale est décroissante alors que la borne inférieure (lagrangienne) peut croître ou décroître d'une itération à l'autre. De plus, la distance entre le point dual courant et l'optimum de la fonction Θ n'est pas décroissante et on remarque de l'instabilité dans son évolution. Un exemple célèbre a été présenté dans [37]. Il s'agit de minimiser la fonction $\Theta(u) = \frac{1}{2}u^2$ sur \mathbb{R} . Si $u_1 = 1$ et $u_2 = -\varepsilon$, alors $u_3 = \frac{1}{2} - \frac{1}{2}\varepsilon$. Si ε diminue, u_2 est plus proche de l'optimum 0. Mais, le point suivant u_3 est alors plus loin de 0. Ainsi, si le point courant est meilleur (plus proche de l'optimum), le point suivant est pire (plus loin de l'optimum).

1.4 Méthodes de résolution stabilisées

Plusieurs méthodes ont été développées dans le but de remédier à l'instabilité des méthodes des plans coupants. Nous commençons par un survol des méthodes de centres avant de présenter plus en détails les méthodes de faisceaux qui ont des ressemblances avec l'approche unifiée que nous proposons. Nous présentons ensuite les principales approches récentes de stabilisation de l'algorithme de génération de colonnes.

1.4.1 Méthodes de centres

Ces méthodes profitent de la stabilité affichée par les méthodes des points intérieurs dans la résolution des programmes linéaires. À la différence de l'algorithme de Kelley, les méthodes de centres déterminent un ensemble de localisation (borné, convexe et fermé) et calculent un point à l'intérieur de cet ensemble. Les points calculés sont appelés centres. L'ensemble (ou domaine) de localisation, noté \mathcal{L} , est déterminé par les approximations externes courantes de la fonction objectif et du domaine réalisable, et par une borne supérieure sur la valeur optimale du problème.

Les méthodes élaborées diffèrent principalement par les propriétés du centre calculé. La méthode d'Elzinga et Moore [16] trouve le prochain point comme étant le centre de la plus grande hypersphère contenue dans \mathcal{L} . La méthode du centre de gravité [60], comme son nom l'indique, trouve le prochain point en calculant le centre de gravité de l'ensemble de localisation \mathcal{L} . Cette méthode possède un taux de convergence optimal à une ε -solution. Mais, vu que le calcul d'un centre de gravité est plus difficile que la résolution du problème original, cette méthode n'est pas efficace en pratique. La méthode volumétrique [86] calcule le prochain point

(centre volumétrique) en minimisant une fonction barrière logarithmique (strictement convexe). Initialement conçue pour un problème de réalisabilité, cette méthode converge en $O(nL)$ itérations, où L est la constante de Lipschitz de la fonction optimisée. Pour contourner la difficulté qui résulte de la résolution optimale de la fonction barrière, ce problème est résolu approximativement en $O(30\ln(2\varepsilon^{-4.5}))$ ou $O(30\ln(4\varepsilon^{-3}))$. Enfin, la méthode ACCPM (*Analytic Center Cutting Plane Method*) [32, 49] trouve le point suivant en calculant un centre analytique (*Analytic Center*) de l'ensemble de localisation \mathcal{L} . Les études de la convergence vers une solution ε -optimale par l'algorithme ACCPM ont été faites en imposant quelques restrictions sur l'implantation. Goffin *et al.* [31] ont établi des résultats de convergence pour le problème de réalisabilité sous certaines conditions de centrage. De plus, le centre analytique n'est pas calculé exactement. La complexité estimée de l'algorithme est $O(\frac{n^2}{\varepsilon^2})$. Ce résultat est généralisé quand σ coupes sont générées par l'oracle. La complexité devient $O(\frac{\sigma^2 n^2}{\varepsilon^2})$ [95]. Kiwiel [47] a étudié une variante plus restreinte et a montré que la complexité est $O(\frac{nL^2}{\varepsilon^2})$.

1.4.2 Méthodes de faisceaux

Les méthodes de faisceaux (*Bundle Methods*) [37] ont été proposées par Lemaréchal [56, 57] et Wolfe [94] dans le but de remédier aux carences observées sur la méthode des plans coupants de Kelley et sur les méthodes du sous-gradient. Les travaux de Kiwiel ([45], [46], [47]) complétèrent ces études. Dans le premier cas, on a remarqué qu'un grand nombre des contraintes (ou colonnes) générées n'améliorent pas réellement l'approximation de la fonction au voisinage de l'optimum. De plus, la qualité du prochain point généré peut évoluer dans le sens inverse de la qualité du point actuel [37]. Dans le cas des méthodes du sous-gradient, même si on choisit le "meilleur" sous-gradient, la direction choisie, très bonne du point de vue local,

devient de plus en plus mauvaise du point de vue global (maximisation de la fonction). Les méthodes de faisceaux visent à stabiliser les méthodes de plans coupants en considérant à chaque fois la meilleure solution trouvée, appelée centre de stabilité (*Stability Center*). À partir du centre de stabilité, un nouveau point est calculé en résolvant un problème quadratique à contraintes linéaires. Si le point trouvé est assez prometteur, il devient le nouveau centre de stabilité. Dans le cas contraire, le centre de stabilité reste le même mais l'approximation linéaire de la fonction est enrichie.

Pour présenter la méthode, nous utilisons le problème suivant :

$$\text{Max}_{u \geq 0} \Theta(u).$$

Les méthodes de faisceaux requièrent que le nouveau point maximise le modèle linéaire $\hat{\Theta}^K$, mais également qu'il ne s'éloigne pas trop du centre de stabilité. Ainsi, cette approche requiert que la relaxation linéaire constitue une bonne approximation de la fonction Θ au voisinage du centre de stabilité. Soit u_K le centre de stabilité courant. Le problème résolu est obtenu en ajoutant à la relaxation linéaire un terme quadratique qui correspond à la distance au centre courant. Ce qui définit le problème quadratique $(QB)_K$

$$\text{sc} \begin{cases} \text{Max} \quad (\eta - \frac{1}{2}\rho_K \|u - u_K\|^2) \\ \Theta(u_k) + g_k^T(u - u_k) \geq \eta, \quad \forall k = 1, \dots, K \\ u \geq 0. \end{cases} \quad (1.30)$$

$\rho_K > 0$ est un paramètre choisi de façon à ce que le nouveau point v_{K+1} qui résoud $(QB)_K$ reste dans le voisinage de u_K où la relaxation linéaire approche assez bien la fonction $\Theta(u)$. Si l'objectif a suffisamment augmenté ($\Theta(v_{K+1}) - \Theta(u_K) \geq \nu(\eta_{K+1} - \eta_K)$ pour une tolérance prédéfinie $\nu > 0$), alors $u_{K+1} = v_{K+1}$ devient le nouveau centre de stabilité. On parle alors de "pas de montée" (*Serious Step*). Sinon, le centre de stabilité ne change pas ($u_{K+1} = u_K$), mais un nouveau plan coupant défini par (v_{K+1}, g_{K+1}) est généré et l'approximation linéaire de Θ est enrichie. Dans ce cas, on parle de "pas nul" (*Null Step*).

Le choix du paramètre ρ_K à chaque itération est déterminant dans le comportement et l'efficacité de la méthode. Un poids constant $\rho_K = \bar{\rho}$ pourrait conduire à des cas extrêmes où la convergence est très lente. Si $\bar{\rho}$ est très grand, il y a toujours des pas réels. Si $\bar{\rho}$ est petit, il y a une multitude de pas nuls. D'une manière générale, il serait intéressant de choisir ρ_{K+1} de façon à conserver une interpolation quadratique de la courbure de Θ entre u_K et v_{K+1} . Cette approche a permis de réduire de manière significative le nombre d'itérations (Kiwiel [46]).

Plusieurs points de vue sont possibles pour des méthodes de faisceaux [37]. Sous la forme présentée ci-dessus, une méthode de faisceaux représente une méthode de plans coupants stabilisée par pénalisation quadratique. Mais, elle peut être vue comme une méthode de plans coupants avec région de confiance (*Trust Region*) où le point v_{K+1} est restreint à rester dans une boule centrée en u_K et d'un rayon prédéterminé. Le point de vue de relaxation considère une méthode de faisceaux comme une méthode de plans coupants avec stabilisation de niveau (*Level Stabilization*), c'est-à-dire qu'elle correspond à l'optimisation d'un problème où la fonction Θ doit être au dessus d'un niveau spécifié à l'avance. Enfin, d'un point de vue dual, une méthode de faisceaux est une méthode de plans coupants avec stabilisation duale (*Dual Stabilization*). Cette approche est utile quand le problème de recherche de la direction de déplacement comporte un grand nombre de contraintes. Nous présentons plus en détails ce dernier point de vue ci-après.

Pour déterminer le dual lagrangien du problème $(QB)_K$, on dualise toutes les contraintes (y compris les contraintes de non-négativité) et on écrit les conditions

d'optimalité du premier ordre :

$$\begin{aligned} \sum_{k=1}^K \lambda_k &= 1 \\ -\rho_K(u - u_K) + \sum_{k=1}^K \lambda_k g_k + \alpha &= 0 \\ \lambda &\geq 0, \alpha \geq 0. \end{aligned}$$

La première relation élimine la variable η de l'expression du lagrangien et la deuxième relation donne l'expression explicite de u à l'optimalité du lagrangien (par rapport à η et u). En substituant dans (1.30), on obtient la formulation suivante pour le dual :

$$\begin{aligned} \text{Min } & \frac{1}{2\rho_K} \left\| \sum_{k=1}^K \lambda_k g_k + \alpha \right\|^2 + \sum_{k=1}^K \lambda_k (e_k^K - \Theta(u_K)) + \alpha^T u_K \\ \text{sc } & \begin{cases} \sum_{k=1}^K \lambda_k = 1 \\ \lambda \geq 0, \alpha \geq 0, \end{cases} \end{aligned} \quad (1.31)$$

où $e_k^K = \Theta(u_K) + (\Theta(u_k) + g_k^T(u_K - u_k))$ est l'erreur résultant de la linéarisation de $\Theta(u)$ au point u_K .

En multipliant l'objectif par ρ_K et en remarquant que le terme $\rho_K(\sum_{k=1}^K \lambda_k (e_k^K - \Theta(u_K)) + \alpha^T u_K)$ est la dualisation d'une contrainte du type

$$\sum_{k=1}^K \lambda_k (e_k^K - \Theta(u_K)) + \alpha^T u_K \leq \varepsilon,$$

le problème dual peut s'écrire

$$\begin{aligned} \text{Min } & \frac{1}{2} \left\| \sum_{k=1}^K \lambda_k g_k + \alpha \right\|^2 \\ \text{sc } & \begin{cases} \sum_{k=1}^K \lambda_k = 1 \\ \sum_{k=1}^K \lambda_k (e_k^K - \Theta(u_K)) + \alpha^T u_K \leq \varepsilon \\ \lambda \geq 0, \alpha \geq 0. \end{cases} \end{aligned} \quad (1.32)$$

Ce problème est appelé problème de faisceaux. En fait, il utilise le faisceau d'information $F_K = \{(g_k, e_k^K, u_k), k = 1, \dots, K\}$ afin de déterminer une direction de montée (ou ε -montée [36]) dans une approximation du sous-différentiel (ou ε -sous-différentiel [36]). Dans une approche plus générale, les méthodes de faisceaux se déplacent dans la direction trouvée d'un certain pas t_K calculé pour l'itération K . Ensuite, suivant la qualité du nouveau point obtenu, le centre de stabilité est changé (*Serious Step*) ou seulement le faisceau est enrichi (*Null Step*).

Pour cette approche, les choix de ε et du pas de descente t_K sont critiques pour l'efficacité de la résolution. Plusieurs schémas sont possibles [37]. Enfin, ce point de vue dual a été appliqué par Tachefine [84] dans le cadre de la planification des opérations dans une mine à ciel ouvert. Les résultats obtenus sont très satisfaisant du point de vue de la qualité de la solution et du temps de calcul.

1.4.3 Approches de stabilisation récentes

Les premières approches de stabilisation peuvent être regroupées en deux classes [43]. Les méthodes du type faisceaux suivent le schéma de l'algorithme de faisceaux. Elles consistent à pénaliser l'approximation courante de la fonction duale $\hat{\Theta}$ par une fonction autre que la norme $\|\cdot\|_2$ et à résoudre le problème résultant. En fait, elles modifient le problème sur lequel est exécutée une itération de génération de colonnes. Un pas de descente (*Serious Step*) est exécuté chaque fois que la croissance de la valeur de la fonction objectif est jugée suffisante. Cette croissance est assurée au moyen de plusieurs pas nuls (*Null Steps*). Cette classe contient les méthodes de faisceaux, l'approche de Schramm et Zowe 92 [79] et la méthode hybride [64]. Les méthodes du type proximal suivent quand à elles le schéma de l'algorithme du point proximal. Elles consistent à pénaliser la fonction duale Θ et à calculer un optimum du problème résultant qui sera le prochain centre de stabilité. Une itération d'optimisation d'un

problème pénalisé sera appelée itération majeure (*Major Iteration*). Les itérations de plans coupants à l'intérieur de chaque itération majeure seront appelées itérations mineures (*Minor Iterations*). Cette classe contient la méthode *Box Step* [61, 62] et la méthode de Kim *et al.*[43].

Ces dernières méthodes peuvent être vues comme des méthodes de type faisceaux. Il suffit de garder la même fonction de pénalité pour les pas nuls (*Null Steps*) à l'intérieur d'un pas de descente (*Serious Step*) donné. Un tel pas est alors exécuté seulement à l'optimalité du problème stabilisé.

Nous considérons le problème de maximisation d'une fonction concave

$$\text{Max}_u \Theta(u). \quad (1.33)$$

Les itérations des approches de type proximal seront indexées par l alors que celles des approches de type faisceaux seront indexées par K .

La méthode Box Step : (Marsten *et al.*1975 [61, 62])

Le problème (1.33) est résolu sur un hyper-cube centré au point courant u_l . La longueur de l'arête de l'hypercube reste constante le long des itérations. Le problème résolu à chaque itération l est le suivant :

$$\begin{aligned} u^{l+1} \in \text{Argmax}_u \Theta(u) \\ \text{sc } \{ \|u - u_l\|_\infty \leq B. \end{aligned} \quad (1.34)$$

Il est également possible de faire varier la taille de l'hypercube en définissant à chaque itération un terme B_l . La résolution peut être ramenée à la résolution d'un programme linéaire par génération de colonnes grâce à l'utilisation de cette norme.

Une variante des méthodes de faisceaux : (Schramm et Zowe 1992 [79])

La variante *Trust Region* des méthodes des faisceaux (cf. 1.4.2) proposée par Schramm et Zowe résoud le problème suivant à chaque itération :

$$u^{K+1} \in \operatorname{Argmax}\{\hat{\Theta}^K(u) - \frac{1}{\alpha_K} \|u - u^K\|_2^2\}. \quad (1.35)$$

Le paramètre de proximité α_K est mis à jour au fil des itérations de façon à mieux estimer la courbure de la fonction.

Pénalisation par norme linéaire :

Kim *et al.* 1995 [43] proposent l'utilisation d'une norme $\|\cdot\|_p$ ($p \in [0, +\infty]$). Le problème résolu à chaque itération l est

$$u^{l+1} \in \operatorname{Argmax}\{\Theta(u) - \frac{1}{\varepsilon_l} \|u - u^l\|_p\}. \quad (1.36)$$

Dans le cas $p \in \{1, \infty\}$, le problème (1.36) se reformule comme un programme linéaire. Dans le cas de la norme $\|\cdot\|_1$, le problème à l'itération K s'écrit sous la forme suivante

$$\begin{aligned} & \operatorname{Max}_{\eta, u, \omega} \quad \eta - \varepsilon_l \sum_{j=1}^m \omega_j \\ \text{sc} \quad & \begin{cases} u_j - u_j^l \leq w_j, \quad \forall j = 1, \dots, m \\ u_j^l - u_j \leq w_j, \quad \forall j = 1, \dots, m \\ \eta + \sum_{k \in I} (M_k - b) \leq d_k, \quad \forall k \in I. \end{cases} \end{aligned} \quad (1.37)$$

Notons que cette approche part de l'idée de l'algorithme du point proximal (*Proximal Point*) proposé par Rockafellar 76 [76].

Une méthode hybride : (du Merle *et al.* 1998 [64])

Cette approche tient compte parallèlement des deux problèmes primal et dual, c'est-à-dire le problème maître restreint (génération de colonnes) et son dual (algo-

arithme de Kelley). On notera la ressemblance entre la fonction de pénalité utilisée dans cette méthode et celle utilisée dans la méthode de Kim *et al.*[43] quoique les deux méthodes n'utilisent pas la même approche de stabilisation. Le problème primal pénalisé se formule comme suit :

$$\begin{aligned}
 & Min_{\lambda, y^-, y^+} \left(\sum_{k \in I_K} d_k \lambda_k \right) - \delta_{-,K}^T y^- + \delta_{+,K}^T y^+ \\
 & sc \left\{ \begin{array}{l} \sum_{k \in I_K} M_k \lambda_k - y^- + y^+ = b \\ y^- \leq \varepsilon_{-,K} \\ y^+ \leq \varepsilon_{+,K} \\ \sum_{k \in I_K} \lambda_k = 1 \\ \lambda, y^-, y^+ \geq 0. \end{array} \right. \quad (1.38)
 \end{aligned}$$

y^- et y^+ sont des variables de surplus et d'écart, $\varepsilon_{-,K}$ et $\varepsilon_{+,K}$ sont les bornes supérieures sur ces variables et $\delta_{-,K}$ et $\delta_{+,K}$ sont les coûts associés aux variables de surplus et d'écart.

Le dual de ce problème est alors

$$\begin{aligned}
 & Max_{\eta, u, \omega^-, \omega^+} \eta - \varepsilon_{-,K}^T \omega^- - \varepsilon_{+,K}^T \omega^+ \\
 & sc \left\{ \begin{array}{l} \eta + M_k^T u - b^T u \leq d_k, \forall k \in I_K \\ -u - \omega^- \leq \delta_{-,K} \\ u - \omega^+ \leq \delta_{+,K} \\ \omega^+, \omega^- \geq 0. \end{array} \right. \quad (1.39)
 \end{aligned}$$

Du point de vue dual, $[\delta_{-,K}, \delta_{+,K}]$ est le pavé dans lequel la pénalité est nulle. $\varepsilon_{-,K}$ et $\varepsilon_{+,K}$ sont les vecteurs des pénalités correspondant à chacune des composantes de u suivant qu'on se trouve à gauche ou à droite de l'intervalle associé.

Si $\delta_{+,K} = \delta_{-,K} = u_K$ et $\varepsilon_{+,K} = \varepsilon_{-,K} = (\epsilon_K, \dots, \epsilon_K)$, on obtient une méthode de faisceaux utilisant la norme linéaire $\|\cdot\|_1$. Par contre, si $\delta_{+,K} - u_K = u_K - \delta_{-,K} = (B, \dots, B)$ et $\varepsilon_{+,K} = \varepsilon_{-,K} = (\infty, \dots, \infty)$, on obtient une variante de la version *Trust Region* des méthodes de faisceaux utilisant la norme $\|\cdot\|_\infty$. C'est pour cela que la méthode est dite hybride.

1.4.4 Approches unifiées

D'autres approches plus générales ont été proposées par Neame 99 [67], Kiwiel 99 [51] et Frangioni 2001 [20].

Neame [67] propose de pénaliser le modèle linéaire résolu à chaque itération de l'algorithme de Kelley à l'aide d'une fonction convexe qui s'annule au centre de stabilité courant. Le terme ajouté à l'objectif du problème (1.29) est $-g^{r,t}(u - u^{r,t})$ où $g^{r,t}$ est une fonction convexe non-négative vérifiant $g^{r,t}(0) = 0$ et $u^{r,t}$ est le centre de stabilité courant. Les indices r et t représentent deux types d'itérations exécutées par l'algorithme (r : itération majeure (*Null Step*); t : itération mineure (*Serious Step*)). Ces dernières itérations sont exécutées lorsque la valeur de la fonction Θ au point calculé coïncide avec son approximation actuelle $\hat{\Theta}$. Le point obtenu est alors optimal pour la fonction Θ pénalisée par la fonction $g^{r,t}$ à l'itération courante. Cependant, la fonction de pénalité peut changer d'une itération majeure (*Null Step*) à l'autre. La convergence finie de l'algorithme est démontrée pour les fonctions de pénalité suffisamment plates en 0 ($g^{r,t}$ ξ -plates au voisinage de 0, i.e., $\exists \varepsilon > 0 : \forall u \in B(0, \varepsilon), \partial g(u) \in B(0, \xi)$, où ξ dépend de la fonction optimisée). Les voisinages correspondant doivent être assez larges pour contenir une solution optimale du problème (1.33).

Kiwiel [51] propose une généralisation des méthodes Bregman-proximales en permettant la résolution approximative des problèmes pénalisés. Les fonctions de stabilisation étudiées sont strictement convexes. L'utilisation de fonctions du type $\frac{1}{t_K}g(\hat{u}^K; u)$, où t_K est le paramètre de proximité, permet d'obtenir les résultats de convergence sous des conditions plus faibles. Ce travail généralise plusieurs études de l'algorithme du point proximal au moyen de fonctions spéciales (e.g. les D -fonctions et les ϕ -divergences)[6, 15, 39, 50].

Frangioni [20] propose une généralisation des méthodes des faisceaux qui englobe la plus part des approches de stabilisation citées ci-haut. Il propose deux algorithmes. Le premier algorithme, à deux niveaux, représente le schéma général des méthodes de faisceaux. Cet algorithme généralise la plus part des approches de stabilisation connues ainsi que d'importantes classes des méthodes d'optimisation non-différentiable de type faisceaux. Le deuxième algorithme, à trois niveau, permet d'affaiblir les conditions requises sur les fonctions de stabilisation pour assurer la convergence de l'algorithme. Une des caractéristiques intéressantes des algorithmes proposés est qu'ils assurent la convergence vers une paire de solutions optimales primale et duale dans le cas où la fonction de stabilisation est différentiable en 0. Pour plus de détails sur la description de l'algorithme, ainsi que pour une étude des liens des algorithmes proposés avec les méthodes de faisceaux connues, le lecteur est référé au travail de l'auteur [20].

L'idée de stabilisation que nous traitons rentre dans le cadre des approches proposées dans [67] (dans le cas des fonctions polyédrales) et [20]. L'algorithme proposé est du type proximal et utilise des fonctions de stabilisation (éventuellement non-différentiables) ayant une région de confiance de pleine dimension définie autour du centre de stabilisation courant. La convergence de l'algorithme est assurée grâce à cette caractéristique. Cette dernière permet d'établir la convergence finie de l'algorithme même dans le cas où la fonction optimisée n'est pas polyédrale, en supposant que les problèmes stabilisés sont résolus exactement. L'extension de l'idée aux méthodes de faisceaux proximales fera l'objet de recherche futures. L'approche proposée sera présentée plus en détails au chapitre 3.

1.5 Conclusion

Nous avons présenté une synthèse des méthodes classiques de résolution des problèmes (convexes) de grande taille. Nous avons d'abord décrit l'approche primale composée de la décomposition de Dantzig-Wolfe et de la génération de colonnes. Une revue des travaux concernant la généralisation du principe de décomposition aux problèmes à variables entières a également été faite. Puis, nous avons présenté l'approche duale qui part de la relaxation lagrangienne pour obtenir des reformulations des problèmes originaux qui peuvent être résolues par les méthodes d'optimisation non-différentiables classiques. Nous y avons mis les développements les plus récents concernant la relaxation lagrangienne et ces méthodes. Ensuite, nous avons fait un survol des difficultés rencontrées par ces approches de résolution ainsi que les aspects d'instabilité qu'elles affichent. Enfin, nous avons présenté les approches récentes de stabilisation de l'algorithme de Kelley. Ces méthodes simples peuvent être regroupées en deux classes dépendamment du schéma général de l'algorithme proposé. Nous avons fait également un survol des approches généralisant les principes des méthodes de faisceaux et de l'algorithme du point proximal. Les méthodes de faisceaux proximales généralisées ainsi obtenues englobent plusieurs classes importantes des méthodes de résolution stabilisées.

Deuxième partie

Stabilisation des méthodes des plans coupants

Chapitre 2

Une méthode proximale pour l'optimisation convexe non-différentiable

Les méthodes proximales sont des généralisations de l'algorithme du point proximal introduit par Rockafellar en 1976 [76, 55, 33]. La plupart de ces méthodes utilisent des fonctions de Bregman [15] ou des fonctions ϕ -divergences [39]. Leur application à l'optimisation convexe produit des régulations de Moreau-Yosida [66] utilisant des fonctions différentiables strictement convexes comme terme de stabilisation. Kiwiel [45] généralisa ces idées au cas où les fonctions utilisées sont non-différentiables mais toujours strictement convexes. Lorsque les problèmes stabilisés sont résolus par une méthode de plans coupants, ces méthodes produisent des algorithmes de plans coupants stabilisés. De tels algorithmes furent proposés par Marsten [61] et par Kim *et al.* [43]. Ils pénalisent le problème dual original et utilisent sa solution comme prochain centre de stabilité ou pour déterminer une direction de montée pour l'objectif. Notons que ces algorithmes sont des cas particuliers des méthodes de faisceaux généralisées [20].

Nous étudions une méthode proximale pour la maximisation d'une fonction concave. Une application de cette méthode à l'algorithme de Kelley sera traitée au chapitre suivant. Le schéma général est semblable à celui proposé par Kim *et al.* [43] pour lequel les fonctions de pénalité sont positives partout sauf au point courant (centre de stabilité) où elles s'annulent. Les fonctions que nous proposons

utilisent une région de confiance contenant le centre de stabilité. Les pénalités sont encourues à l'extérieur de cette région. Nous considérons des régions de confiance de pleine dimension au centre de stabilisation courant. L'étude de la convergence suppose la résolution exacte des problèmes stabilisés à chaque itération (majeure). La convergence finie est établie même pour des fonctions non-polyédrales.

Nous présentons d'abord le principe général de la méthode ainsi que les idées qui ont motivé notre étude de cette approche. Nous étudions la convergence de la méthode pour des régions de confiance de pleine dimension et donnons quelques cas particuliers des fonctions de pénalité. Nous discutons ensuite de l'extension de la méthode au cas contraint, du cas où les problèmes stabilisés sont résolus approximativement ainsi que des avantages de l'utilisation de régions de confiance de pleine dimension. Enfin, nous regardons plus en détails le cas où la fonction à optimiser est affine par morceaux et le domaine réalisable est un polyèdre et nous motivons notre choix de fonctions de pénalité linéaires par morceaux pour ce genre de problèmes. Ce cas correspond à la résolution d'un programme linéaire par génération de colonnes pour lequel nous proposons, au prochain chapitre, une approche de stabilisation basée sur la méthode que nous développons dans ce chapitre.

2.1 Principe

Nous considérons le problème de maximisation d'une fonction concave sur \mathbb{R}^m que nous notons (D) :

$$\begin{aligned} \text{Max } & \Theta(u) \\ & u \in \mathbb{R}^m. \end{aligned} \tag{2.1}$$

La méthode se généralise facilement au cas où le domaine réalisable est un ensemble convexe Ω . Normalement, ce problème est résolu par une méthode de plans coupants

de Kelley. À chaque itération K , un modèle linéaire par morceaux de la fonction Θ noté $\hat{\Theta}^K$ est maximisé. La résolution de ce problème restreint est facile et revient à la résolution d'un programme linéaire. Cependant, le processus de convergence vers une solution optimale de (D) présente beaucoup d'instabilité. L'évolution des variables duales d'une itération à l'autre n'assure pas une amélioration stricte de l'objectif de (D) .

L'idée que nous proposons consiste à résoudre une suite de problèmes dits stabilisés définis autour d'un point appelé centre de stabilité. Ce centre est déplacé à la fin de la résolution du problème stabilisé tout en s'assurant d'une amélioration stricte de l'objectif de (D) . Un autre problème stabilisé est défini avec une nouvelle fonction de pénalité, et ainsi de suite jusqu'à la convergence de l'algorithme. Ainsi, la méthode profite de la simplicité de l'algorithme de Kelley [42] tout en s'assurant de l'amélioration de l'objectif défini par (2.1) à chaque déplacement du centre de stabilité.¹ Les fonctions que nous utilisons seront définies à l'aide d'un ensemble appelé région de confiance (*Trust Region*). À l'extérieur de cette région, une pénalité sera appliquée afin d'empêcher un grand déplacement inutile dans l'espace dual.

Chaque itération de résolution d'un problème stabilisé est appelé itération majeure (*Major Iteration*). Chaque problème stabilisé est résolu par un algorithme de plans coupants. Les itérations de cet algorithme pour résoudre le problème stabilisé sont appelées itérations mineures (*Minor Iterations*). Il faut noter qu'un problème stabilisé peut être résolu par une méthode de plans coupants stabilisée de type faisceaux. Cependant, nous allons voir plus loin que dans le cas linéaire, ces problèmes

¹Quand le problème pénalisé est résolu par l'algorithme de Kelley, la méthode peut être vue comme une méthode de type faisceaux dont les pas de montée (*Serious Step*) sont exécutés seulement à l'optimalité des problèmes pénalisés. Toutefois dans le cas où les problèmes stabilisés sont non-linéaires et/ou difficiles, leur résolution par une méthode de plans coupants stabilisée de type faisceaux serait d'un apport intéressant à l'efficacité de l'optimisation. En utilisant une seule itération majeure, on obtient alors une méthode de plans coupants stabilisée de type faisceaux.

pénalisés sont souvent plus faciles à résoudre que le problème original et que la résolution par l'algorithme de Kelley moyennant quelques astuces d'implantation est satisfaisante. Nous nous intéressons donc plus aux itérations majeures et à la gestion des centres de stabilité.

2.2 Pénalisation du problème dual

Nous définissons la fonction de pénalité et le problème stabilisé pour une itération majeure l . \hat{u}^l est le centre de stabilité courant et T_l est la région de confiance courante. Nous supposons que T_l est convexe fermé et que $\hat{u}^l \in T_l$. Le problème (D) sera pénalisé par un terme $-g^l(u)$ où g^l est une fonction convexe qui vérifie les propriétés suivantes :

$$\begin{cases} g^l(u) = 0, & \text{si } u \in T_l \\ g^l(u) > 0, & \text{si } u \notin T_l. \end{cases} \quad (2.2)$$

Le problème stabilisé à l'itération l , noté (SD_l) , est alors défini comme suit

$$\begin{aligned} \text{Max } & \Theta(u) - g^l(u) \\ & u \in \mathbb{R}^m. \end{aligned} \quad (2.3)$$

Notons d'abord que si Θ est bornée, et admet donc un ensemble de solutions optimales non-vide, le problème stabilisé (SD_l) admet un ensemble de solutions optimales également non-vide. Cependant, il est possible que (D) soit non-borné alors que (SD_l) ait un ensemble de solutions optimales non-vide. Nous ferons notre étude en supposant que le problème (D) est fini et que son ensemble des solutions optimales est non-vide. Pour un programme mathématique (P) , nous notons sa valeur optimale par $v(P)$. Sous les hypothèses que nous venons d'énoncer, nous pouvons affirmer que

$$v(SD_l) \leq v(D). \quad (2.4)$$

Nous noterons également une solution optimale de (D) par u^* et une solution optimale de (SD_l) par u^l .

2.3 Étude de la convergence

Chaque itération majeure de la méthode proposée correspond à la recherche d'une direction de montée pour la fonction Θ . La proposition suivante montre que si la solution de (SD_l) n'est pas dans la région de confiance, alors elle fournit une direction de montée pour le problème (D) .

Proposition 2.1 *Soit u^l la solution optimale de (SD_l) . Si $u^l \notin T_l$, alors*

- i) $\Theta(u^l) > \Theta(\hat{u}^l)$
- ii) $d^l = u^l - \hat{u}^l$ est une direction de montée pour Θ .

Preuve:

- i) Puisque u^l est une solution optimale de (SD_l) , alors

$$\Theta(u^l) - g^l(u^l) \geq \Theta(\hat{u}^l) - g^l(\hat{u}^l).$$

Comme $\hat{u}^l \in T_l$ et $u^l \notin T_l$, on a nécessairement $g^l(\hat{u}^l) = 0$ et $g^l(u^l) > 0$.

Par conséquent,

$$\Theta(u^l) > \Theta(\hat{u}^l).$$

- ii) Θ étant concave, on déduit directement de i) que $d^l = u^l - \hat{u}^l$ est une direction de montée pour la fonction Θ . ■

Pour les propositions qui suivent, l'hypothèse que T_l est de pleine dimension est nécessaire. Elle peut être représentée par la condition suivante :

$$\text{int}(T_l) \neq \emptyset. \quad (2.5)$$

L'intérêt de cette condition est que toutes les directions possibles sont réalisables en u^l pour (SD_l) . Une conséquence de cette condition est que pour tout $u \in \text{int}(T_l)$, il existe une boule centrée en u et incluse dans T_l . Et ainsi toutes les directions seront admissibles en u .

La prochaine proposition permet de conclure à l'optimalité de u^l pour (D) s'il est à l'intérieur de T_l .

Proposition 2.2 *Soit u^l la solution obtenue par l'optimisation de (SD_l) . Si $u^l \in \text{int}(T_l)$, alors u^l est une solution optimale de (D) .*

Preuve: La condition (2.5) assure que $\text{int}(T_l)$ n'est pas vide. Supposons que $u^l \in \text{int}(T_l)$. Comme $g^l(u) = 0 \ \forall u \in T_l$, alors u^l maximise Θ sur T_l .

De plus, on peut définir $\eta > 0$ tel que $B(u^l, \eta) \subset T_l$. Donc, u^l maximise Θ sur $B(u^l, \eta)$. On en déduit que u^l est un maximum local de Θ . Cette dernière étant concave, u^l maximise Θ sur \mathbb{R}^m . ■

La proposition suivante traite le cas où u^l est sur la frontière de T_l .

Proposition 2.3 *Supposons que la solution optimale u^l obtenue par l'optimisation de (SD_l) est sur la frontière de T_l . Alors, l'une des deux assertions suivantes est vraie :*

- i) $\Theta(u^l) > \Theta(\hat{u}^l)$ et $d^l = u^l - \hat{u}^l$ est une direction de montée pour (D) .
- ii) $\Theta(u^l) = \Theta(\hat{u}^l)$ et tout u appartenant au segment (\hat{u}^l, u^l) est une solution optimale de (D) .

Preuve: Soit u^l la solution optimale obtenue par la résolution de (SD_l) . Supposons que u^l appartient à la frontière de T_l , c'est-à-dire que $u^l \in T_l - \text{int}(T_l)$. Sachant que $v(SD_l) \leq v(D)$ et que $g^l(\hat{u}^l) = g^l(u^l) = 0$, on peut écrire

$$\Theta(u^l) \geq \Theta(\hat{u}^l).$$

Le cas i) est trivial à montrer. Nous supposons donc que $\Theta(u^l) = \Theta(\hat{u}^l)$. Par conséquent, u^l et \hat{u}^l maximisent Θ sur T_l . Donc, \hat{u}^l maximise Θ sur $B(\hat{u}^l, \rho)$ et est une solution optimale de (D) .

Sachant que Θ est concave, $\forall \alpha \in [0, 1]$

$$\Theta(\alpha \hat{u}^l + (1 - \alpha)u^l) \geq \alpha \Theta(\hat{u}^l) + (1 - \alpha)\Theta(u^l) = v(SD_l) = v(D).$$

Donc, tout point du segment (\hat{u}^l, u^l) est aussi une solution optimale de (D) . ■

Ces deux propositions affirment que soit u^l est optimal soit qu'il définit une direction de montée pour (D) . La proposition qui suit examine ce qui se passe quand T_l contient une solution optimale de (D) .

Proposition 2.4 Si T_l contient une solution optimale u^* de (D) , alors $v(SD_l) = v(D)$ et toute solution optimale u^l de (SD_l) est optimale pour (D) et vérifie $u^l \in T_l$.

Preuve: Sachant que $v(SD_l) \leq v(D)$ et que $u^* \in T_l$, on conclut que $v(SD_l) = \Theta(u^*) - g^l(u^*) = v(D)$.

De plus tout $u \notin T_l$ vérifie $\Theta(u) - g^l(u) < \Theta(u) \leq \Theta(u^*)$. D'où toute solution optimale de (SD_l) sera dans T_l et sera optimale pour (D) . ■

Ce résultat permet d'affirmer que si la région de confiance ne contient pas de solution optimale, la résolution du problème stabilisé détermine une direction de montée pour le problème (D) . Par contre, si la région de confiance contient une solution optimale différente de \hat{u}^l , on ne peut pas conclure à l'optimalité de u^l après la résolution de (SD_l) . La proposition suivante affirme que la résolution du problème stabilisé (SD_{l+1}) défini autour du centre de stabilisation $\hat{u}^{l+1} = u^l$ permet de conclure à l'optimalité de u^{l+1} pour (D) .

Proposition 2.5 *Supposons que T_l contient une solution optimale u^* de (D) et soit u^l la solution optimale obtenue par la résolution de (SD_l) .*

La résolution de (SD_{l+1}) défini autour de $\hat{u}^{l+1} = u^l$ produit une solution u^{l+1} qui vérifie

i) soit $u^{l+1} \in \text{int}(T_l)$,

ii) soit u^{l+1} est sur la frontière de T_l et $\Theta(u^{l+1}) = \Theta(\hat{u}^{l+1})$.

Dans les deux cas, on peut conclure à l'optimalité de u^{l+1} pour (D) .

Preuve: L'hypothèse que T_l contient une solution optimale de (D) permet d'affirmer (proposition 2.4) que u^l est une solution optimale de (D) et que $u^l \in T_l$. Puisque le problème (SD_{l+1}) est défini autour d'une solution optimale de (D) , d'après la proposition 2.4, $\Theta(u^{l+1}) = \Theta(\hat{u}^{l+1})$ et $u^{l+1} \in T_l$.

Si $u^{l+1} \in \text{int}(T_l)$, la proposition 2.2 permet de conclure à l'optimalité de u^{l+1} pour (D) .

Si u^{l+1} est sur la frontière de T_l , le fait que $\Theta(u^{l+1}) = \Theta(\hat{u}^{l+1})$ permet de conclure à l'optimalité de u^{l+1} pour (D) (proposition 2.3 ii). ■

L'intérêt de cette proposition est qu'elle traduit l'aptitude de l'algorithme à trouver une solution optimale de (D) en au plus 2 itérations majeures si la région de confiance contient une solution optimale de (D) .

Nous sommes donc en possession de tous les éléments nécessaires pour écrire un algorithme qui converge vers une solution optimale de (D) .

2.3.1 Algorithme

Nous donnons l'algorithme général pour la maximisation d'une fonction concave. Nous discuterons après des étapes non détaillées.

Étape 0 : Initialisation

Choisir un point initial \hat{u}^0 , une région de confiance T_0 ($\hat{u}^0 \in \text{int}(T_0)$) et une fonction de pénalité g^0 ; $l = 0$

Calculer $\Theta(\hat{u}^0)$

Étape 1 : Itération majeure

1.1 Résoudre (SD_l)

$$u^l \in \text{Argmax}\{\Theta(u) - g^l(u)\}$$

1.2 Calculer $\Theta(u^l)$

1.3 Si $u^l \in \text{int}(T_l)$ ou si $u^l \in T_l - \text{int}(T_l)$ et $\Theta(u^l) = \Theta(\hat{u}^l)$, u^l est une solution optimale pour (D) ; **STOP**.

Sinon, aller à l'étape 2

Étape 2 : Déplacement du centre de stabilité

2.1 $d^l = u^l - \hat{u}^l$ est une direction de montée pour (D)

2.2 Déterminer un pas $\alpha_l > 0$

2.3 $\hat{u}^{l+1} = \hat{u}^l + \alpha_l d^l$.

2.4 Définir T_{l+1} et g^{l+1} . Aller à l'étape 1

À partir d'un point initial quelconque, une direction de montée est déterminée en résolvant un problème stabilisé autour du centre courant \hat{u}^l . Si on ne peut conclure à l'optimalité de u^l , le centre est alors déplacé dans la direction déterminée.

La convergence de l'algorithme est assurée par la condition suivante :

$$\exists \rho > 0 : \forall l, B(\hat{u}^l, \rho) \subset T_l, \quad (2.6)$$

Théorème 2.1 *Si*

- i) *les problèmes stabilisés (SD_l) sont résolus exactement à chaque étape 1.1 de l'algorithme,*
- ii) *la condition (2.6) est vérifiée et*
- iii) *le problème (D) possède une solution optimale finie,*

alors

l'algorithme, implanté avec $\alpha_l = 1$ à l'étape 2.3, converge vers une solution optimale de (D) en un nombre fini d'itérations (majeures).

Preuve: Nous prouvons le théorème en deux étapes. D'abord, nous prouvons en a) que la suite $\Theta(\hat{u}^l)$ converge vers la valeur optimale de (D). Ensuite, nous prouvons la convergence finie en b).

- a) Si l'algorithme s'arrête après un nombre fini d'itérations majeures en 1.3, alors il fournit une solution optimale de (D).

Considérons la suite $\Theta(\hat{u}^l)$ ($l = 1, 2, \dots$). D'après les propositions 2.1-2.3, cette suite est strictement croissante (tant que l'algorithme ne s'arrête pas en 1.3).

L'hypothèse iii) permet d'affirmer que cette suite est bornée. Donc elle converge vers une valeur finie notée $\bar{\Theta}$. Si Θ^* est la valeur optimale de (D) , alors

$$\bar{\Theta} \leq \Theta^*.$$

Définissons l'ensemble de niveau

$$\bar{L} = \{u \in \mathbb{R}^m : \Theta(u) = \bar{\Theta}\} \neq \emptyset.$$

Pour un entier l suffisamment grand et un certain $\bar{u} \in \bar{L}$,

$$d(\hat{u}^l, \bar{u}) < \rho.$$

L'hypothèse ii) (i.e la condition (2.6)) permet d'écrire

$$\bar{u} \in \text{int}T_l.$$

En utilisant la proposition 2.4 (et les précédentes), on peut affirmer que : si \bar{u} n'est pas optimal pour (D) (i.e. $\bar{\Theta} < \Theta^*$), alors $\hat{u}^{l+1} = u^l$ calculé à l'étape 1.1 est tel que

$$\Theta(\hat{u}^{l+1}) > \Theta(\bar{u}) = \bar{\Theta}.$$

Ceci contredit le fait que la suite $\Theta(\hat{u}^l)$ est croissante et converge vers $\bar{\Theta}$.

On en déduit que la suite $\Theta(\hat{u}^l)$ converge vers la valeur optimale de (D) Θ^* .

b) Un raisonnement similaire avec l'ensemble de niveau

$$L^* = \{u \in \mathbb{R}^m : \Theta(u) = \Theta^*\} \neq \emptyset$$

conduit à l'existence d'un entier l et de $u^* \in L^*$ (i.e. u^* solution optimale de (D)) tel que

$$d(\hat{u}^l, u^*) < \rho,$$

et par conséquent

$$u^* \in \text{int}(T_l).$$

La proposition 2.5 assure alors que l'algorithme terminera au plus à l'itération $l + 1$ avec une solution optimale de (D) . ■

La convergence finie de l'algorithme peut être prouvée pour tout choix des pas α_l qui assure une augmentation stricte de la valeur de la fonction Θ (i.e. $\Theta(\hat{u}^{l+1}) > \Theta(\hat{u}^l)$). Ceci est vrai en particulier si $\alpha_l \in]0, 1]$, $\forall l$. Une variante d'intérêt pratique de la direction de déplacement choisie est étudiée au chapitre suivant.

2.3.2 Remarques

Conditions de convergence de l'algorithme

La condition (2.6) est relativement forte pour assurer la convergence de l'algorithme. Afin d'assurer la convergence finie, il suffit que T_l soit de pleine dimension pour tout l et que la convergence de la suite $(\Theta(\hat{u}^l))_l$ soit plus rapide que la convergence du diamètre des régions de confiance T_l vers 0 (si c'est le cas).

D'autres conditions assurant la convergence de la suite $(\hat{u}^l)_l$ (éventuellement en un nombre infini d'itérations) vers une solution optimale de (D) existent dans [67] et [20]. Remarquons également que la condition (2.6) peut être liée à conditions sur les pas de déplacements utilisées dans les méthodes procimales (voir par exemple [15] et [51]).

Choix du point initial :

L'algorithme converge indépendamment du choix de \hat{u}^0 et T_0 en autant que $\hat{u}^0 \in \text{int}(T_0)$. Cependant en pratique, une bonne estimation d'une solution optimale est d'une grande utilité pour l'efficacité de la méthode. Une telle estimation dépend généralement du problème résolu. Nous verrons dans les prochains chapitres que ceci peut jouer un rôle dans l'efficacité de l'optimisation.

Choix du pas :

Un choix trivial pour α_l est la valeur 1. Le prochain centre de stabilité est alors l'optimum obtenu pour (SD_l) . Il est également possible d'effectuer une recherche linéaire pour le meilleur pas dans la direction d^l . Autrement, toute valeur de $\alpha_l \in]0, 1]$ assure une augmentation de l'objectif.

Résolution des problèmes stabilisés :

Chaque problème stabilisé est lui-même un problème de maximisation d'une fonction concave. Mais, ce problème pénalisé est supposément plus facile à résoudre grâce à la pénalité imposée. La pénalisation a un effet de restriction de l'espace des points admissibles ce qui réduit le déplacement nécessaire pour atteindre l'optimalité. Le choix de la fonction g^l est déterminant pour l'efficacité de la résolution de (SD_l) . Ce choix dépend bien sûr de la fonction Θ et de la méthode de résolution utilisée. Le problème stabilisé est généralement résolu par une méthode de plans coupants. L'utilisation d'une pénalité linéaire par morceaux permettra de résoudre un programme linéaire à chaque itération mineure.

Résolution approximative :

Si la fonction Θ n'est pas affine par morceaux, la résolution des problèmes stabilisés produira des solutions approchées. Généralement, si on désire une solution ϵ -optimale du problème original, les problèmes stabilisés doivent être résolus avec une précision $\eta \ll \epsilon$. Mais, l'utilisation de régions de confiance de pleine dimension permettrait de s'assurer que l'optimum d'un problème stabilisé est à l'intérieur de la région de confiance du problème stabilisé suivant. Les problèmes stabilisés pourront

alors être résolus avec une précision égale à la précision désirée pour la solution du problème original (i.e. $\eta = \epsilon$). Des résultats apparentés concernant les méthodes de faisceaux proximales peuvent être trouvés dans [50, 51]. Notons enfin que l'étude de convergence dans ce cas peut être faite en suivant les schémas des méthodes de faisceaux généralisées ([20]).

Cas contraint :

Si le problème à résoudre est contraint par $u \in \Omega$ convexe, la région de confiance T_l doit être de même dimension que Ω . Notons qu'il n'est pas nécessaire que chaque centre de stabilité appartienne à l'intérieur de Ω . Toutefois, si on veut se déplacer à l'intérieur du domaine réalisable, il faut choisir \hat{u}^0 pour qu'il soit à l'intérieur de Ω et que le pas α_l soit choisi de façon que le prochain centre \hat{u}^{l+1} soit à l'intérieur de Ω . Un pas $\alpha_l \in]0, 1[$, par exemple, permet de vérifier cette condition.

2.4 Maximisation d'une fonction concave affine par morceaux

Dans le cas où Θ est une fonction concave par morceaux (avec un nombre fini de morceaux), la convergence finie de l'algorithme est assurée par le nombre fini de morceaux de la fonction et par la résolution exacte des problèmes stabilisés à chaque itération. De plus, le cas linéaire présente beaucoup de particularités qui font que les conditions requises pour la convergence de l'algorithme sont plus faibles. Nous proposons dans le chapitre suivant l'utilisation d'une fonction de pénalité linéaire par morceaux pour stabiliser l'algorithme de génération de colonnes pour la résolution des programmes linéaires. L'utilisation de telles fonctions de pénalité permet de garder

des problèmes stabilisés linéaires dont la résolution par génération de colonnes est plus facile que celle du problème original. Les conditions sur le déplacement du centre de stabilité sont plus souples et l'enrichissement de la direction de montée trouvée est possible sans compromettre la convergence de l'algorithme.

La fonction de pénalité linéaire que nous proposons dans le prochain chapitre est en fait une combinaison de la fonction utilisée dans la méthode *Box Step* [61] et de la fonction de norme linéaire proposée par Kim *et al.* [43]. La fonction de la méthode *Box Step* est la suivante

$$g^l(u) = \begin{cases} 0 & \text{si } \|u - \hat{u}^l\|_\infty \leq \beta \\ \infty & \text{si } \|u - \hat{u}^l\|_\infty > \beta, \end{cases}$$

alors que la fonction utilisée par Kim *et al.* est définie par

$$g^l(u) = \begin{cases} 0 & \text{si } u = \hat{u}^l \\ \frac{1}{\epsilon_l} \|u - \hat{u}^l\|_1 & \text{si } u \neq \hat{u}^l. \end{cases}$$

La méthode utilise conjointement le concept de région de confiance autour du centre de stabilité et l'idée de pénalisation de l'éloignement du centre courant. La figure 2.1 représente les trois fonctions de pénalité et montre comment le type de fonctions que nous utilisons combine les fonction de pénalité de la méthode *Box Step* et la méthode de la norme linéaire. Notre fonction est différentiable autour du centre de stabilité et représente une meilleure approximation extérieure de la norme euclidienne que la fonction de la méthode *Boxstep*. La fonction de Kim *et al.* est quant à elle une approximation intérieure (localement).

Le chapitre suivant est consacré à l'étude complète de la méthode définie par une fonction linéaire à cinq morceaux. L'utilisation de cinq morceaux vise à mieux approcher une pénalité quadratique.

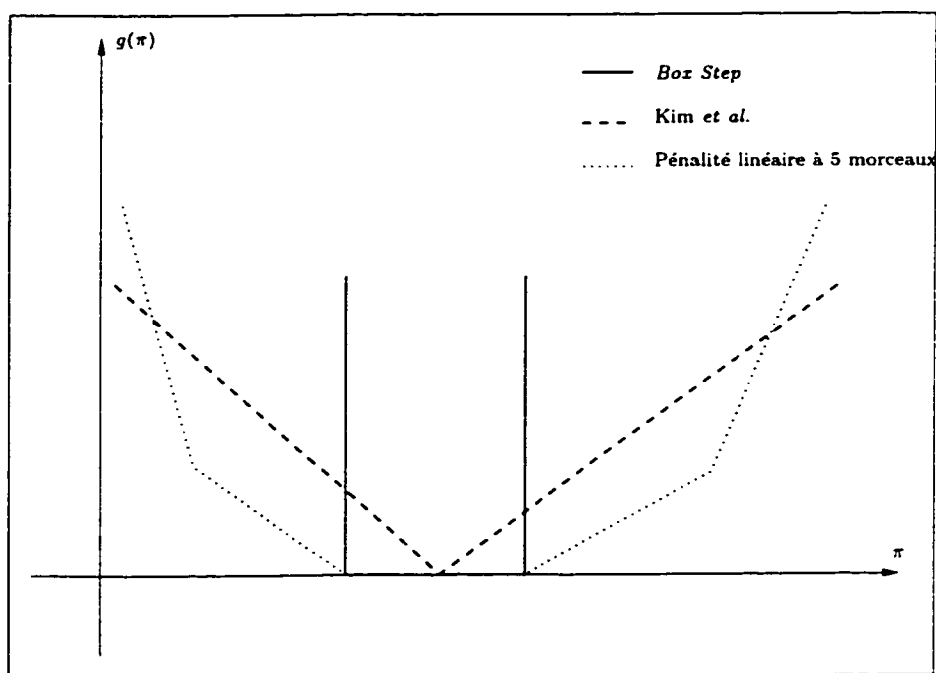


Figure 2.1 – Stabilisation : Illustration du type de fonction de pénalité utilisée.

2.5 Conclusion

L'approche de stabilisation présentée dans ce chapitre est une généralisation de l'idée de l'algorithme du point proximal pour la maximisation d'une fonction concave. L'idée est basée sur les concepts de région de confiance et de pénalisation. La région de confiance correspond à un ensemble défini autour du centre de stabilité courant, où aucune pénalité n'est appliquée. À l'extérieur de cette région, une pénalité est encourue. Le but est d'empêcher le prochain centre de stabilité de trop s'éloigner du centre courant tout en procurant une certaine flexibilité à l'algorithme. La résolution d'un problème stabilisé correspond à une itération majeure (étape de montée). Cette résolution est effectuée par une méthode de plans coupants. Les itérations de plans coupants sont appelées itérations mineures. Notons enfin que la différence principale avec l'approche de Kim *et al.* 95 [43] est que dans cette dernière la région de confiance

est réduite au singleton formé par le centre de stabilité. Dans notre cas, le fait que la région de confiance soit de plus grande dimension peut donner de la flexibilité à l'algorithme surtout dans le cas où les problèmes stabilisés sont résolus approximativement. Ce choix permet aussi d'assurer la convergence vers une solution primale comme nous le verrons dans le chapitre suivant pour la résolution des programmes linéaires par génération de colonnes.

La convergence de l'algorithme est étudiée sous la condition que la région de confiance soit de pleine dimension. Ceci assure la convergence finie de l'algorithme même quand la fonction optimisée est non-polyédrale à condition que les problèmes stabilisés soient résolus exactement. Dans le cas contraint, la dimension doit être la même que la dimension du domaine réalisable. Dans le cas linéaire (maximisation d'une fonction concave affine par morceaux sur un polyèdre), la pénalisation par une fonction affine par morceaux permet la convergence finie sans conditions supplémentaires. Nous traitons le cas d'une pénalité linéaire à cinq morceaux dans le chapitre suivant. L'étude peut être généralisée à toute fonction de pénalité linéaire par morceaux.

Chapitre 3

Stabilisation par une fonction de pénalité linéaire par morceaux

Dans une procédure de génération de colonnes et dans l'algorithme de Kelley, le problème maître qu'il faut résoudre est un programme linéaire. Ce problème est généralement le résultat de la décomposition d'une certaine formulation dite originale. Le dual de ce problème peut être aussi obtenu par relaxation lagrangienne de la même formulation originale. Il correspond alors à la maximisation d'une fonction concave linéaire par morceaux sur un polyèdre. Les contraintes qui définissent ce polyèdre sont des colonnes du problème primal.

Dans la suite de ce texte, nous utilisons une formulation générale pour un programme linéaire sous forme matricielle. La matrice A est de dimension $m \times n$, $c \in \mathbb{R}^n$ et $b \in \mathbb{R}^m$. Le problème maître primal est noté (P) et son dual est noté (D) .

$$\begin{array}{cc}
 (P) & \qquad \qquad \qquad (D) \\
 \text{Min } c^T x & \qquad \qquad \qquad \text{Max } b^T \pi \\
 \text{sc } \left\{ \begin{array}{l} Ax = b \\ x \geq 0 \end{array} \right. & \qquad \qquad \qquad \text{sc } \left\{ A^T \pi \leq c \right.
 \end{array}$$

L'approche que nous développons dans ce chapitre s'applique à tout programme linéaire. Cependant, elle est surtout utile dans le cas d'un programme linéaire qui est difficile à résoudre. C'est souvent le cas des problèmes résolus par génération de colonnes où le nombre de colonnes est très élevé et où toutes ces colonnes ne peuvent être connues préalablement à la résolution du problème.

À chaque itération de génération de colonnes, un problème maître restreint est résolu. Ce problème est défini à l'aide d'une sous-matrice de A notée A_K ($m \times n_K$) où K est le numéro de l'itération courante de génération de colonnes. Les formulations du problème maître restreint et de son dual sont alors les suivantes :

$$\begin{array}{c|c}
 (P_K) & (D_K) \\
 \hline
 \begin{array}{l} \text{Min } c^T x \\ \text{sc } \left\{ \begin{array}{l} A_K x = b \\ x \geq 0 \end{array} \right. \end{array} & \begin{array}{l} \text{Max } b^T \pi \\ \text{sc } \left\{ \begin{array}{l} A_K^T \pi \leq c \end{array} \right. \end{array}
 \end{array}$$

Le problème (P_K) est une *restriction* de (P) . La fonction objectif est la même pour les deux problèmes alors que le domaine réalisable de (P_K) est une *restriction* du domaine réalisable de (P) : certaines colonnes de la matrice A ne sont pas prises en compte dans la formulation de (P_K) . Par contre, le problème (D_K) est une *relaxation* de (D) étant donné que seulement un sous-ensemble des contraintes de (D) sont prises en compte. Après chaque résolution de (P_K) , on obtient une paire de solutions optimales primale et duale (x^K, π^K) . x^K est une solution réalisable pour le problème (P) alors que π^K ne l'est pas pour (D) , sauf à l'optimalité. $c^T x^K$ fournit une borne supérieure pour (P) qui décroît à chaque itération, alors que la borne inférieure fournie par π^K peut croître ou décroître d'une itération de génération de colonnes à l'autre (cf. chapitre 2).

La méthode que nous proposons est un cas particulier de l'approche unifiée présentée au chapitre 2. Elle consiste à résoudre une suite de problèmes dits stabilisés jusqu'à la convergence de l'algorithme vers une paire de solutions optimales primale et duale de la paire de problèmes $((P), (D))$. Chaque paire de problèmes stabilisés primal et dual est obtenue en pénalisant le dual (D) par une fonction concave linéaire par morceaux. Il est important de noter dès le début que la procédure de stabilisation proposée ne modifie pas directement la résolution du problème maître restreint ou la résolution du sous-problème comme les méthodes de faisceaux [37]. Elle consiste à appliquer plusieurs fois la même procédure de génération de colonnes à des

problèmes plus faciles dans le but de converger plus rapidement vers des solutions optimales des problèmes primal et dual. Chaque résolution complète des problèmes stabilisés est appelée itération majeure (*Major Iteration*). À l'intérieur d'une itération majeure, plusieurs itérations de génération de colonnes sont nécessaires pour atteindre l'optimalité des problèmes stabilisés. Ces itérations sont appelées itérations mineures (*Minor Iterations*). L'utilisation d'une fonction linéaire par morceaux permet de préserver la linéarité des problèmes résolus à chaque itération. La première contribution importante de ce chapitre est la preuve d'un résultat puissant sous des hypothèses assez faibles qui fournit une manière simple de calculer une solution optimale primale de base pour tout programme linéaire connaissant un multiplicateur optimal dual. L'efficacité de cette technique, justifiée par des arguments mathématiques, est vérifiée sur une instance de grande taille du problème de tournées de véhicules multi-dépôts (*MDVSP*). L'analyse des solutions des problèmes stabilisés établit l'amélioration de l'objectif dual à chaque itération majeure. Ceci assure la convergence vers une solution optimale duale. L'optimalité primale est obtenue grâce au résultat mentionné ci-dessus. Ceci est dû à l'utilisation d'une région de confiance de pleine dimension autour des centres de stabilité. La convergence de l'algorithme prouvée, nous discutons de quelques aspects de l'implantation. Nous montrons également qu'il est possible d'enrichir la direction de montée par n'importe quelle direction sans compromettre la convergence de l'algorithme sous des conditions simples et assez faibles.

Nous commençons ce chapitre par la présentation des idées principales derrière notre méthode et la définition de la fonction de pénalité proposée. Puis, nous étudions les propriétés des problèmes stabilisés en montrant leurs liens avec les problèmes originaux (P) et (D). Nous étudions ensuite le cas particulier où une solution optimale duale est connue. Par la suite, nous examinons les propriétés des solutions optimales des problèmes stabilisés, avant de présenter l'algorithme de stabilisation proposé

pour lequel nous étudions la convergence et l'implantation.

3.1 Pénalité linéaire à cinq morceaux

Vu l'instabilité observée dans la convergence des variables duales dans les algorithmes de génération de colonnes et de Kelley (*cf.* chapitre 2), nous allons essayer de guider l'évolution des variables duales afin d'assurer une convergence plus rapide. Les idées principales derrière l'approche de stabilisation sont décrites ci-après.

- Résoudre une suite de problèmes stabilisés $((SP_l), (SD_l))$ dont les solutions convergent vers des solutions optimales de (P) et de (D) . Chaque itération l (*itération majeure*) correspond à l'application d'une procédure de génération de colonnes complète qui permet de résoudre les problèmes stabilisés.
- La paire de problèmes stabilisés est formée par le primal et le dual obtenus suite à la pénalisation du dual (D) . La pénalité, linéaire par morceaux, est construite autour d'une estimation d'une solution optimale duale (centre de stabilisation ou de stabilité)¹. Ceci permet d'éviter les oscillations et les grands déplacements inutiles et nuisibles dans l'espace dual vu que l'information disponible n'est pas forcément pertinente loin du centre de stabilisation.
- S'assurer de l'amélioration de la qualité de la solution duale après la résolution du problème stabilisé. Le nouveau centre de stabilisation présente une croissance stricte de l'objectif de (D) par rapport au précédent. Ainsi, la pénalité sera placée autour d'un meilleur ensemble de multiplicateurs de (P) .
- La pénalité linéaire assure que les problèmes stabilisés sont des programmes linéaires et on n'introduit pas ainsi une difficulté supplémentaire dans la résolution de ces problèmes.

¹L'appellation *centre de stabilité* est utilisée dans les méthodes de faisceaux. C'est cette appellation que nous utiliserons en général quoique nous pensons que l'appellation *centre de stabilisation* est aussi convenable.

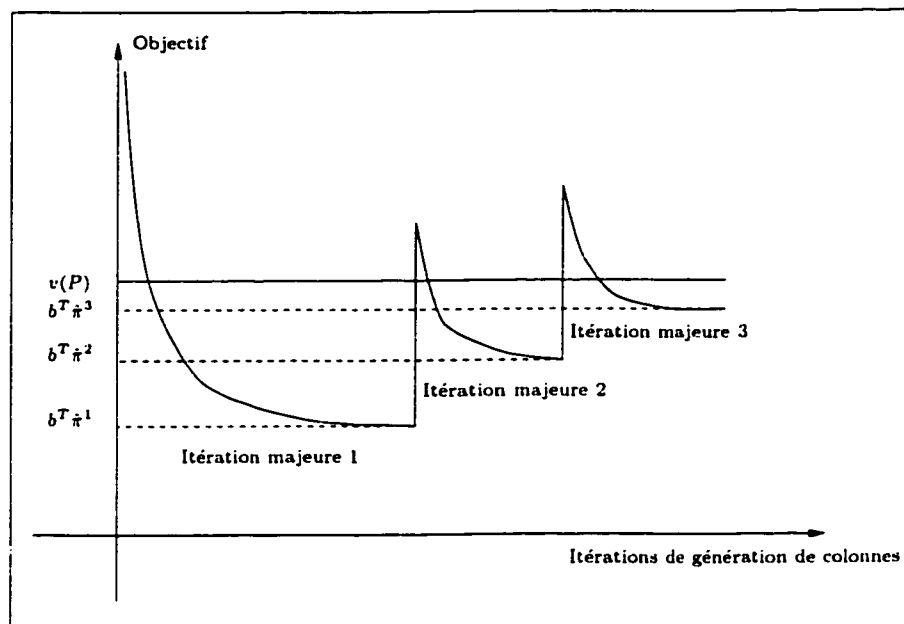


Figure 3.1 – Évolution de l'objectif primal et des bornes inférieures.

- Les problèmes pénalisés sont plus faciles à résoudre que le problème original. La pénalisation entraîne que la résolution du dual se fait sur un domaine “plus restreint” et que la solution obtenue n'est pas très loin du centre de stabilisation. Le primal est ainsi une relaxation de (P) .

La figure 3.1 montre l'évolution de l'objectif primal (des problèmes stabilisés) dans un processus à trois itérations majeures. La valeur optimale de chaque itération est une borne inférieure sur la valeur optimale de (P) . La figure illustre la croissance stricte de cette borne avec les itérations majeures.

3.1.1 Fonction de pénalité

Nous présentons la fonction de pénalité utilisée avec un indice l qui correspond à l'itération majeure courante. La fonction, notée $-g^l(\pi)$ ($\pi = [\pi_1, \dots, \pi_m]^T$), est

définie par :

$$g^l(\pi) = \sum_{i=1}^m g_i^l(\pi_i) \quad (3.1)$$

où

$$g_i^l(\pi_i) = \begin{cases} (\zeta_{-,i}^l + \varepsilon_{-,i}^l)(\gamma_{-,i}^l - \pi_i) & \text{si } -\infty \leq \pi_i \leq \gamma_{-,i}^l \\ \varepsilon_{-,i}^l(\delta_{-,i}^l - \pi_i) & \text{si } \gamma_{-,i}^l \leq \pi_i \leq \delta_{-,i}^l \\ 0 & \text{si } \delta_{-,i}^l \leq \pi_i \leq \delta_{+,i}^l \\ \varepsilon_{+,i}^l(\pi_i - \delta_{+,i}^l) & \text{si } \delta_{+,i}^l \leq \pi_i \leq \gamma_{+,i}^l \\ (\varepsilon_{+,i}^l + \zeta_{+,i}^l)(\pi_i - \gamma_{+,i}^l) & \text{si } \gamma_{+,i}^l \leq \pi_i \leq +\infty \end{cases} \quad (3.2)$$

Afin que la définition de la fonction soit cohérente, on doit avoir

$$\gamma_+^l \geq \delta_+^l \geq \delta_-^l \geq \gamma_-^l. \quad (3.3)$$

Pour que, g_i^l ($i = 1, \dots, m$) soit concave, les conditions suivantes doivent être vérifiées :

$$\begin{aligned} \zeta_-^l &\geq \varepsilon_-^l > 0 \\ \zeta_+^l &\geq \varepsilon_+^l > 0. \end{aligned} \quad (3.4)$$

Nous verrons plus tard que les fonctions les plus utiles vérifient en plus la condition $\delta_-^l < \delta_+^l$.

La région de confiance correspond au pavé défini par les vecteurs δ_-^l et δ_+^l , c'est-à-dire l'ensemble $\{\pi \in \mathbb{R}^m : \delta_-^l \leq \pi \leq \delta_+^l\}$. Ainsi pour chaque composante π_i , on tolère sans pénalité les valeurs à l'intérieur de l'intervalle $[\delta_{-,i}^l, \delta_{+,i}^l]$. Une pénalité unitaire égale à $-\varepsilon_{+,i}^l$ (respectivement $-\varepsilon_{-,i}^l$) est appliquée dans l'intervalle $[\delta_{+,i}^l, \gamma_{+,i}^l]$ (respectivement $[\gamma_{-,i}^l, \delta_{-,i}^l]$). Pour $\pi_i > \gamma_{+,i}^l$ (respectivement $\pi_i < \gamma_{-,i}^l$), la pénalité est $-(\varepsilon_{+,i}^l + \zeta_{+,i}^l)$ (respectivement $-(\varepsilon_{-,i}^l + \zeta_{-,i}^l)$).

La figure 3.2 illustre la fonction $g_i^l(\pi)$ autour de la composante $\hat{\pi}_i^l$ du centre de stabilisation $\hat{\pi}^l$.

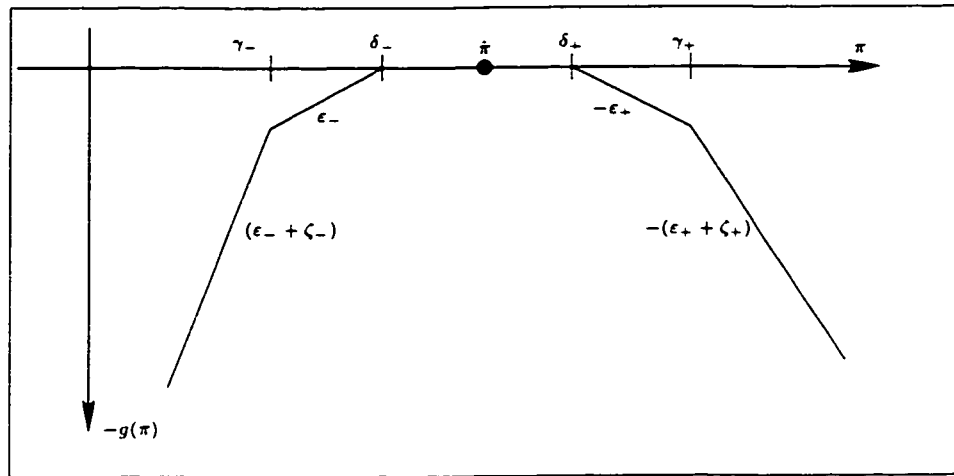


Figure 3.2 - Fonction de pénalité linéaire à cinq morceaux.

3.1.2 Formulation des problèmes stabilisés

Le problème dual pénalisé $((SD_l))$ est formulé comme suit :

$$\begin{aligned} \text{Max } & b^T \pi - g^l(\pi) \\ \text{sc } & \{ A^T \pi \leq c \end{aligned} \quad (3.5)$$

La fonction $g^l(\pi)$ étant linéaire par morceaux et convexe, (SD_l) peut être écrit comme un programme linéaire. Son dual est appelé problème primal stabilisé (SP_l) .

Les formulations des deux problèmes sont les suivantes :

$$\begin{array}{c}
 (SP_l) \\
 \\
 \text{Min } c^T x - \gamma_-^T z^- - \delta_-^T y^- + \delta_+^T y^+ + \gamma_+^T z^+ \\
 \text{sc } \begin{cases} Ax - z^- - y^- + y^+ + z^+ = b \\ z^- \leq \zeta_-, y^- \leq \varepsilon_- \\ y^+ \leq \varepsilon_+, z^+ \leq \zeta_+ \\ x, z^-, y^-, y^+, z^+ \geq 0 \end{cases} \\
 \\
 (SD_l) \\
 \\
 \text{Max } b^T \pi - \zeta_-^T v^- - \varepsilon_-^T u^- - \varepsilon_+^T u^+ - \zeta_+^T v^+ \\
 \text{sc } \begin{cases} A^T \pi \leq c \\ \delta_- - u^- \leq \pi \leq \delta_+ + u^+ \\ \gamma_- - v^- \leq \pi \leq \gamma_+ + v^+ \\ v^-, u^-, u^+, v^+ \geq 0. \end{cases}
 \end{array}$$

Notons que (SP_l) a $4m$ variables et $4m$ contraintes de borne de plus que (P) et que (SD_l) a $4m$ variables et $4m$ contraintes de plus que (D) .

Nous étudions maintenant le comportement des problèmes stabilisés par rapport au comportement des problèmes (P) et (D) .

Relations entre (P) et (SP_l) : Si (P) est réalisable et fini, (SP_l) l'est aussi. Par contre, (SP_l) peut être réalisable et fini alors que (P) est non-réalisable.

Si (SP_l) est non-borné, alors (P) est forcément non-borné. La réciproque est vraie si $\varepsilon_{\pm}^l + \zeta_{\pm}^l < \infty$.

Si (SP_l) est non-réalisable, (P) est nécessairement non-réalisable. Par contre, (P) peut être non-réalisable alors que (SP_l) est réalisable.

Relations entre (D) et (SD_l) : Le domaine réalisable de (SD_l) est le même que celui de (D) (voir la formulation (3.5) de (SD_l)). Donc, (D) est réalisable si et seulement si (SD_l) est réalisable.

De plus, si (SD_l) est non-borné, (D) est aussi non-borné. Le contraire n'est pas forcément vrai. Le résultat dépend des paramètres de la pénalité.

Si $\min_i \{\varepsilon_{-,i}^l + \zeta_{-,i}^l\} \geq \max_i \{-b_i : b_i \leq 0\}$ et $\min_i \{\varepsilon_{+,i}^l + \zeta_{+,i}^l\} \geq \max_i \{b_i : b_i \geq 0\}$, alors même si (D) est non-borné, (SD_l) est réalisable et fini.

Si $\max_i \{\varepsilon_{-,i}^l + \zeta_{-,i}^l\} < \min_i \{-b_i : b_i < 0\}$ et $\max_i \{\varepsilon_{+,i}^l + \zeta_{+,i}^l\} < \min_i \{b_i : b_i > 0\}$, alors si (D) est non-borné, (SD_l) l'est aussi.

3.2 Propriétés des problèmes stabilisés

Dans cette section, nous supposons que (P) et (D) sont réalisables et finis. Nous noterons par π^l et x^l des solutions optimales de (SD_l) et (SP_l) respectivement.

3.2.1 Analyse duale

Il est clair que le problème (SD_l) diffère du problème (D) seulement par la pénalisation de l'objectif. Nous notons par $v(P)$ la valeur optimale d'un problème (P) donné. On peut alors écrire l'inégalité

$$v(P) = v(D) \geq v(SD_l).$$

De plus toute solution réalisable π de (SD_l) est réalisable pour (D) et vérifie

$$b^T \pi \geq b^T \pi - g^l(\pi).$$

Donc, si π^l est une solution optimale de (SD_l) , alors $b^T \pi^l$ est une borne inférieure sur $v(D) = v(P)$ meilleure que $v(SD_l) = b^T \pi^l - g^l(\pi^l)$. En résumé, nous avons la relation suivante :

$$v(P) = v(D) \geq b^T \pi^l \geq v(SD_l) = v(SP_l). \quad (3.6)$$

Le résultat suivant montre que sous certaines conditions relativement faibles sur la fonction g^l , les deux problèmes (D) et (SD_l) ont la même valeur optimale.

Proposition 3.1 : Soit π^* une solution optimale de (D) qui vérifie $\delta_-^l \leq \pi^* \leq \delta_+^l$. Alors (D) et (SD_l) ont la même valeur optimale, i.e. $v(D) = v(SD_l)$. De plus, toute solution optimale de (SD_l) est aussi optimale pour (D) .

Preuve: D'après la relation (3.6), on a $v(D) \geq v(SD_l)$. De plus, nous savons qu'une solution optimale de (SD_l) π^l est réalisable pour (D) . Étant donné que $\pi^* \in [\delta_-^l, \delta_+^l]$, $g^l(\pi^*) = 0$. Donc, $v(D) = b^T \pi^* = v(SD_l)$. De même, toutes les solutions optimales de (SD_l) étant réalisables pour (D) , elles sont optimales pour (D) . ■

Le résultat est aussi vrai si on sait que D^* , l'ensemble des solutions optimales duales de (D) , est inclu dans la boîte $[\delta_-^l, \delta_+^l]$. Mais dans ce cas, on peut avoir un résultat plus fort en élargissant les bornes du pavé de façon à englober toutes les solutions duales optimales.

Proposition 3.2 : Si $D^* \in]\delta_-^l, \delta_+^l[$, alors les solutions optimales de (SD_l) et (SP_l) sont optimales pour (D) et (P) .

Preuve: Soient $(\pi^l, v^{+,l}, u^{+,l}, u^{-,l}, v^{-,l})$ une solution optimale de (SD_l) ainsi que $(x^l, z^{+,l}, y^{+,l}, y^{-,l}, z^{-,l})$ une solution optimale de (SP_l) . D'après la proposition précédente, on sait que π^* est une solution optimale de (D) . En plus, puisque $D^* \subset]\delta_-^l, \delta_+^l[$,

$$\delta_-^l < \pi^l < \delta_+^l,$$

et $v^{+,l} = u^{+,l} = u^{-,l} = v^{-,l} = 0$; par complémentarité, on déduit que $z^{+,l} = y^{+,l} = y^{-,l} = z^{-,l} = 0$. Donc, x^* est réalisable pour (P) et en est par conséquent une solution optimale. ■

Un cas particulier de cette proposition correspond à avoir $\delta_-^l = \gamma_-^l = -M$, $\delta_+^l = \gamma_+^l = +M$ et $\varepsilon_-^l = \varepsilon_+^l = +M$, avec $M \rightarrow \infty$. En pratique, ceci correspond à pénaliser la non-réalisabilité par des coûts très élevés (méthode du grand M). Mais, quoique cette méthode est fréquemment utilisée pour la vérification de la réalisabilité d'un problème, elle n'est point efficace pour la résolution des problèmes à l'optimalité.

3.2.2 Analyse primale

Le domaine réalisable de (SP_l) est une relaxation de celui de (P) . En effet, on permet une relaxation limitée par les valeurs de $\varepsilon_+^l, \varepsilon_-^l, \zeta_-^l$ et ζ_+^l . Par contre, la fonction objectif de (SP_l) peut être supérieure ou inférieure à celle de (P) . Si $\gamma_-^l > 0$, $\delta_-^l > 0$, $\delta_+^l < 0$ et $\gamma_+^l < 0$, l'objectif est relaxé et la valeur optimale de (SD_l) est en général strictement inférieure à celle de (P) . Elle fournit une borne inférieure et la solution obtenue x^l n'est généralement pas réalisable pour (P) . Si $\gamma_-^l < 0$, $\delta_-^l < 0$, $\delta_+^l > 0$ et $\gamma_+^l > 0$, l'objectif est pénalisé, ce qui ne permet pas de conclure directement vu que le domaine est relaxé.

La formulation du problème primal (SP_l) utilise des variables qui modélisent la

non-réalisabilité. Notre étude va donc se baser sur l'analyse de la non-réalisabilité du problème.

Nous formulons d'abord le lagrangien de (P) sous une forme particulière qui sera utile à l'analyse des solutions de (SP_l) et (SD_l) .² En effet, nous verrons que la formulation de (SP_l) revient à pénaliser le lagrangien de (P) autour de $\{x : b - Ax = 0\}$.

Le lagrangien du problème (P) est défini par

$$L(x; \pi) = c^T x + \pi^T (b - Ax), \quad x \geq 0, \quad \pi \in \mathbb{R}^m.$$

On peut l'écrire de manière équivalente comme suit :

$$\begin{aligned} \mathcal{L}(x, z^+, y^+, y^-, z^-; \pi) &= c^T x + \pi^T (z^+ + y^+ - y^- - z^-) \\ b - Ax &= z^+ + y^+ - y^- - z^- \\ x, z^+, y^+, y^-, z^- &\geq 0, \quad \pi \in \mathbb{R}^m. \end{aligned} \quad (3.7)$$

En utilisant cette notation, on peut écrire la relation suivante qui traduit le fait que pour x^* solution optimale de (P) et π^* solution optimale de (D) , (x^*, π^*) est un point de selle pour le lagrangien du problème (P) :

$$\begin{aligned} \mathcal{L}(x^*, 0, 0, 0, 0; \pi) &\leq \mathcal{L}(x^*, 0, 0, 0, 0; \pi^*) \leq \mathcal{L}(x, z^+, y^+, y^-, z^-; \pi^*) \\ z^+ + y^+ - y^- - z^- &= b - Ax \\ \forall x \geq 0, z^+ \geq 0, y^+ \geq 0, y^- \geq 0, z^- \geq 0, \pi &\in \mathbb{R}^m. \end{aligned} \quad (3.8)$$

Et par conséquent on peut écrire

$$\begin{aligned} b^T \pi^* &= \max \{ \mathcal{L}(x^*, 0, 0, 0, 0; \pi) : \pi \in \mathbb{R}^m \} \\ c^T x^* &= \min \{ \mathcal{L}(x, z^+, y^+, y^-, z^-; \pi^*) : x, z^+, y^+, y^-, z^- \geq 0, \\ &\quad b - Ax = z^+ + y^+ - y^- - z^- \}. \end{aligned} \quad (3.9)$$

²Nous faisons la même analyse que celle faite au chapitre 2 mais nous modélisons la non-réalisabilité par 4 variables.

Notons que pour $\hat{\pi} \in \mathbb{R}$, le problème (SP_l) peut s'écrire

$$\begin{aligned} \text{Min} \quad & \mathcal{L}(x, z^+, y^+, y^-, z^-; \hat{\pi}) + \\ & (\gamma_+^l - \hat{\pi})^T z^+ + (\delta_+^l - \hat{\pi})^T y^+ + \\ & (\hat{\pi} - \delta_-^l)^T y^- + (\hat{\pi} - \gamma_-^l)^T z^- \\ \text{sc} \quad & \begin{cases} Ax - z^- - y^- + y^+ + z^+ = b \\ z^- \leq \zeta_-, y^- \leq \varepsilon_- \\ y^+ \leq \varepsilon_+, z^+ \leq \zeta_+ \\ x, z^-, y^-, y^+, z^+ \geq 0 \end{cases} \end{aligned}$$

Il est intéressant de noter que pour $\hat{\pi} \in]\delta_-^l, \delta_+^l[$, l'objectif correspond à une pénalisation de $\mathcal{L}(x, z^+, y^+, y^-, z^-; \hat{\pi})$ lorsque $Ax \neq b$. De plus, ceci est vrai pour tous les multiplicateurs dans la boîte $]\delta_-^l, \delta_+^l[$.

Le prochain théorème établit l'équivalence entre la résolution des paires de problèmes $((P), (D))$ et $((SP_l), (SD_l))$ sous certaines conditions assez faibles sur la fonction de stabilisation g^l .

Théorème 3.1 : *Si la fonction g^l vérifie les conditions (3.3) et (3.4) et $\delta_-^l < \delta_+^l$, et si $\exists \pi^*$ solution optimale de (D) qui vérifie $\delta_-^l < \pi^* < \delta_+^l$, alors toute solution optimale de (SP_l) $(x^l, z^{+,l}, y^{+,l}, y^{-,l}, z^{-,l})$ vérifie $z^{+,l} = y^{+,l} = y^{-,l} = z^{-,l} = 0$; ce qui implique que x^l est une solution optimale de (P) .*

Preuve: Soit x^* une solution optimale de (P) . La relation (3.9) stipule que

$$c^T x^* = \min \{ \mathcal{L}(x, z^+, y^+, y^-, z^-; \pi^*) : \begin{aligned} & x, z^+, y^+, y^-, z^- \geq 0, \\ & b - Ax = z^+ + y^+ - y^- - z^- \end{aligned} \}.$$

Soit maintenant $(x^l, z^{+,l}, y^{+,l}, y^{-,l}, z^{-,l})$ une solution optimale de (SP_l) ; on a :

$$x^l, z^{+,l}, y^{+,l}, y^{-,l}, z^{-,l} \geq 0$$

et

$$b - Ax^l = z^{+,l} + y^{+,l} - y^{-,l} - z^{-,l}.$$

Par conséquent,

$$c^T x^* = \mathcal{L}(x^*, 0, 0, 0, 0; \pi^*) \leq \mathcal{L}(x^l, z^{+,l}, y^{+,l}, y^{-,l}, z^{-,l}; \pi^*).$$

Or,

$$\begin{aligned} v(SP_l) &= \mathcal{L}(x^l, z^{+,l}, y^{+,l}, y^{-,l}, z^{-,l}; \pi^*) \\ &+ (\gamma_+^l - \pi^*)^T z^{+,l} + (\delta_+^l - \pi^*)^T y^{+,l} + (\pi^* - \delta_-^l)^T y^{-,l} + (\pi^* - \gamma_-^l)^T z^{-,l}. \end{aligned}$$

Donc, si $\gamma_-^l \leq \delta_-^l \leq \pi^* \leq \delta_+^l \leq \gamma_+^l$ et sachant que $(x^*, 0, 0, 0, 0)$ est une solution réalisable de (SP_l) , on a les relations suivantes :

$$v(P) = c^T x^* = \mathcal{L}(x^*, 0, 0, 0, 0; \pi^l) \leq \mathcal{L}(x^l, z^{+,l}, y^{+,l}, y^{-,l}, z^{-,l}; \pi^l) \leq v(SP_l) \leq c^T x^*.$$

Il en résulte que $v(P) = v(SP_l)$ et

$$(\gamma_+^l - \pi^*)^T z^{+,l} + (\delta_+^l - \pi^*)^T y^{+,l} + (\pi^* - \delta_-^l)^T y^{-,l} + (\pi^* - \gamma_-^l)^T z^{-,l} = 0$$

Donc si $\gamma_-^l \leq \delta_-^l < \pi^* < \delta_+^l \leq \gamma_+^l$, alors $(\gamma_+^l - \pi^*) \geq (\delta_+^l - \pi^*) > 0$ et $(\pi^* - \gamma_-^l) \geq (\pi^* - \delta_-^l) > 0$.

On peut en déduire que $z^{+,l} = y^{+,l} = y^{-,l} = z^{-,l} = 0$.

Par conséquent x^l est une solution optimale réalisable pour (P) . ■

3.2.3 Remarques importantes

Afin de mieux comprendre la preuve présentée et de voir graphiquement les implications de la formulation de (SP_l) , nous faisons les remarques suivantes :

1. Comme nous l'avons noté dans la preuve, la fonction objectif de (SP_l) peut s'écrire comme

$$L(x, z^+, y^+, y^-, z^-; \pi) + [(\gamma_+^l - \pi)^T z^+ + (\delta_+^l - \pi)^T y^+ + (\pi - \delta_-^l)^T y^- + (\pi - \gamma_-^l)^T z^-].$$

et ce pour tout $\pi \in \mathbb{R}^m$. Sachant que z^+, y^+, y^- et z^- représentent une mesure de la non-réalisabilité de x ($b - Ax = y^+ - y^- + z^+ - z^-$), si $\gamma_-^l \leq \delta_-^l < \pi < \delta_+^l \leq \gamma_+^l$, le terme entre crochets représente alors une pénalité convexe de la non-réalisabilité de x . C'est-à-dire que si $b - Ax \neq 0$, les valeurs correspondantes (> 0) de z^+, y^+, y^- et z^- sont prises en compte dans le coût, en plus du lagrangien. La pénalité correspondante est illustrée par la figure 3.3.

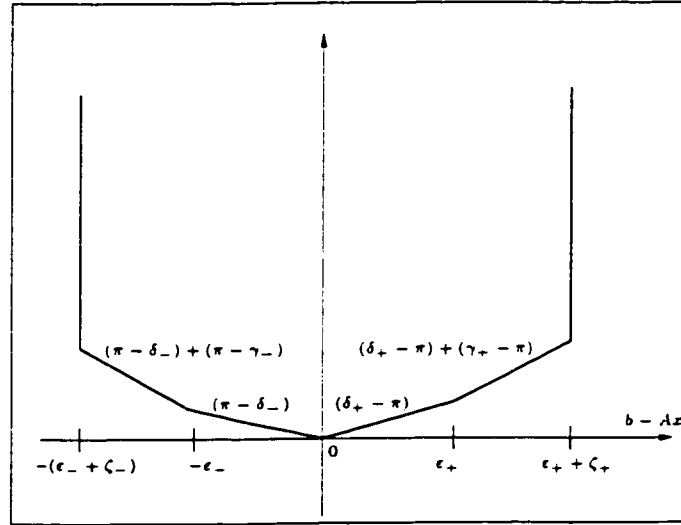


Figure 3.3 – Pénalité ajoutée au lagrangien de (P) autour de $\pi \in]\delta_-^l, \delta_+^l[$.

2. Les modifications résultantes sur (SP_l) peuvent aussi être interprétées en fonction des solutions optimales duales. Pour une solution optimale π^* de (D) , l'ensemble des solutions x pour lesquelles $L(x, z^+, y^+, y^-, z^-; \pi^*) = b^T \pi^* = v(P)$ est l'intersection de $\{x : c^T x = b^T \pi^*\}$ avec $\{x : \pi^{*T}(b - Ax) = 0\}$. La figure 3.4 montre bien que si x n'est pas réalisable ($b - Ax \neq 0$), une quantité strictement positive est ajoutée à $b^T \pi^* = v(P)$. Ainsi, un x vérifiant $c^T x = b^T \pi^*$ et $\pi^{*T}(b - Ax) = 0$ ne peut être optimal pour (SP_l) s'il n'est pas réalisable pour (P) .
3. Il est important de noter que le résultat du théorème 3.1 est vrai si un π^* quelconque est dans $]\delta_-^l, \delta_+^l[$. C'est-à-dire que c'est vrai même si le polyèdre des solutions optimales duales n'est pas borné.

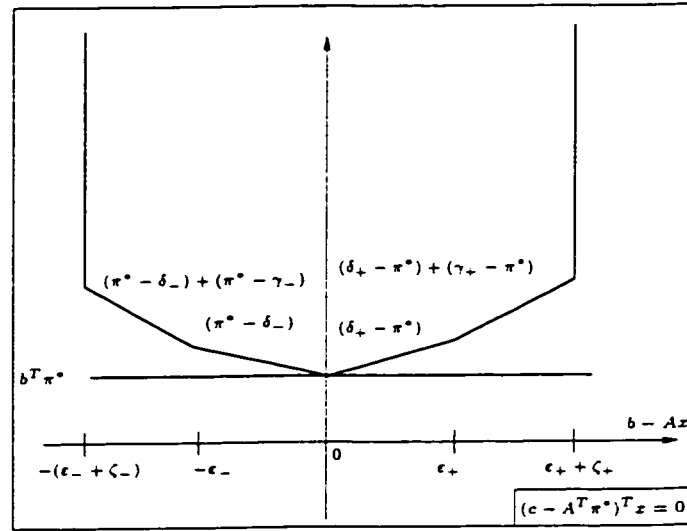


Figure 3.4 – Quantité ajoutée à $b^T \pi^*$ en fonction de la réalisabilité avec (> 0) ou sans stabilisation ($= 0$) pour les x tels que $\pi^{*T}(b - Ax) = 0$ et $c^T x = b^T \pi^*$.

Idéalement, il faut localiser ou estimer n'importe quelle solution optimale duale et on pourra obtenir une solution primale optimale (réalisable) pour (P) .

4. Contrairement à la résolution d'un problème par génération de colonnes qui prend en considération tout l'ensemble des réels afin d'y trouver un multiplicateur optimal, la résolution de (SP_l) revient à considérer seulement les multiplicateurs qui sont dans $[\delta_-^l, \delta_+^l]$. Le multiplicateur trouvé (optimal pour (SP_l)), s'il n'est pas optimal pour (P) , est meilleur que les multiplicateurs qui sont dans $[\delta_-^l, \delta_+^l]$.

Si $[\delta_-^l, \delta_+^l]$ contient une solution optimale π^* de (D) , la résolution de (SP_l) revient à faire tout le processus de relaxation lagrangienne en pénalisant les multiplicateurs qui sont trop loin de π^* (à l'extérieur de l'intervalle $[\delta_-^l, \delta_+^l]$) et fournira une solution primale optimale de (P) .

5. La résolution de la paire de problèmes $((SP_l), (SD_l))$ devrait être plus facile que celle de $((P), (D))$ pour deux raisons principales. La première est que la pénalisation du dual empêche les grands déplacements et oscillations inutiles des multiplicateurs durant la résolution en plus de restreindre l'ensemble des

multiplicateurs admissibles. La deuxième est la facilité de résolution de (SP_l) vu que l'objectif descend plus vite étant donné qu'on tolère la non-réalisabilité. Ceci est illustré par la figure 3.5.

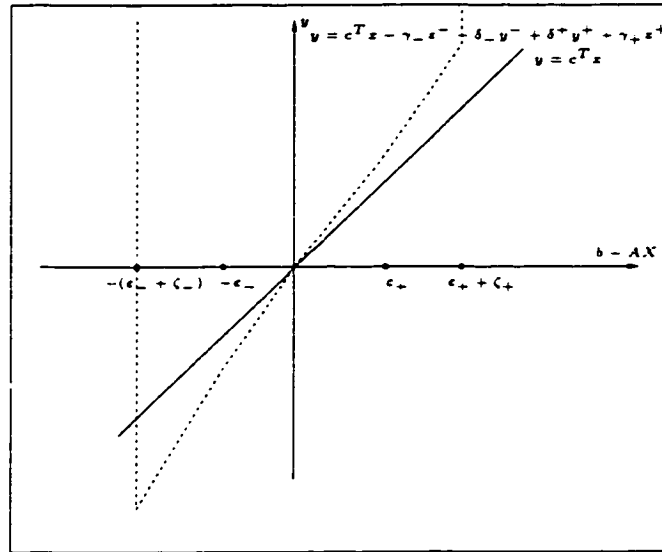


Figure 3.5 – Effet de la non-réalisabilité sur l'objectif du primal $c^T x$ dans le cas où $\gamma_- > 0, \delta_- > 0, \delta_+ > 0$ et $\gamma_+ > 0$.

6 Il faut noter que notre approche de stabilisation devrait être utilisée pour des problèmes ayant un grand nombre de contraintes. Donc si le nombre de contraintes m est élevé, on risque de se trouver à ajouter un nombre assez élevé de colonnes, soit $4m$. Ceci peut diminuer l'efficacité de la résolution du simplexe durant la génération de colonnes. Il sera donc important de trouver un moyen de limiter le nombre de colonnes ajoutées.

3.3 Cas où une solution duale optimale est connue

Dans la présente section, nous montrons d'abord comment la résolution de (SP_l) est très avantageuse par rapport à la résolution de (P) quand on connaît une so-

lution optimale duale. Ceci constitue la première motivation pour l'algorithme de stabilisation que nous présenterons par la suite.

Soit π^* une solution optimale de (D) . On choisit les paramètres de la fonction g^l de manière à avoir

$$\delta_-^l < \pi^* < \delta_+^l.$$

Ainsi, à l'optimalité de (SP_l) , on obtient une paire de solutions (x^l, π^l) optimales pour (P) et (D) respectivement (Théorème 3.1).

Regardons d'abord le cas où la fonction g^l consiste juste à imposer que π soit dans la boîte $[\delta_-^l, \delta_+^l]$. Ceci revient à avoir des pénalités $\varepsilon_-^l = \zeta_-^l = \zeta_+^l = \varepsilon_+^l \rightarrow +\infty$. La proposition suivante donne des arguments qui plaident en faveur de l'efficacité de la résolution de (SD_l) ainsi défini par génération de colonnes.

Proposition 3.3 *Supposons que g^l est définie autour de π^* de façon que :*

$$\delta_-^l < \pi^* < \delta_+^l \\ \varepsilon_-^l = \zeta_-^l = \zeta_+^l = \varepsilon_+^l \rightarrow +\infty.$$

Alors,

- *Pendant le processus de génération de colonnes, seules les colonnes qui coupent la boîte $[\delta_-^l, \delta_+^l]$ sont générées.*
- *La solution primale de départ est $x = 0$, $y_i^+ = b_i$ si $b_i \geq 0$, et $y_i^- = -b_i$ si $b_i \leq 0$.*
- *Les multiplicateurs sont définis par $\pi_i = \delta_{-,i}^l$ si $b_i < 0$ et $\pi_i = \delta_{+,i}^l$ si $b_i > 0$.*
- *La valeur de l'objectif au départ est*

$$\sum_{i:b_i>0}^m \delta_{+,i}^l b_i - \sum_{i:b_i<0} \delta_{-,i}^l b_i.$$

- *Le saut d'optimalité initial est*

$$\sum_{i:b_i>0}^m (\delta_{+,i}^l - \pi_i^*) |b_i| + \sum_{i:b_i<0} (\pi_i^* - \delta_{-,i}^l) |b_i|$$

Ainsi, on peut dire que plus la boîte est petite plus la résolution est efficace. On peut également réduire graduellement la largeur de la boîte, au fur et à mesure que les colonnes sont générées.

Remarque : Le comportement est le même pour le cas où la fonction g^l possède cinq morceaux, avec $\varepsilon_{-,i}^l = \varepsilon_{+,i}^l \geq |b_i|$ ($i = 1, \dots, m$). On peut aussi répartir les quantités $|b_i|$ de façon à avoir $\varepsilon_{-,i}^l + \zeta_{-,i}^l \geq |b_i|$ et $\varepsilon_{+,i}^l + \zeta_{+,i}^l \geq |b_i|$. Enfin, une fois que la réalisabilité est atteinte, ces bornes n'ont plus aucun effet.

Cette dernière approche permet d'éliminer graduellement les variables de stabilisation. Ceci est d'autant plus utile que dans le cas général, on ne connaît pas une solution optimale duale. On doit alors utiliser une fonction de pénalisation plus souple afin de pouvoir avancer plus loin quand c'est vraiment pertinent.

Nous allons vérifier ceci sur une instance d'un problème de tournées de véhicules pour lequel la résolution de la relaxation linéaire est très coûteuse en temps de calcul. La description détaillée de ce problème se trouve dans le chapitre consacré à l'application de la présente approche de stabilisation sur ce même problème.

Les tests sont effectués sur une instance du problème de tournées de véhicules multi-dépôts (*MDVSP*)[75]. Les tests utilisent une fonction g^l linéaire à cinq morceaux. Nous avons utilisé une instance à 800 clients (contraintes) et 4 dépôts. Le tableau 3.1 présente les résultats obtenus par une procédure de génération de colonnes standard (qui s'est avérée très efficace pour la majorité de problèmes résolus avec un nombre de contraintes réduit : ≤ 300), noté standard. Pour chacune des versions stabilisées, nous donnons la largeur de la boîte $[\delta_-, \delta_+]$. La ligne *Sol Opt*

donne la valeur optimale du problème résolu alors que la colonne *Sol Init* donne la solution initiale avec laquelle commence la procédure de génération de colonnes. *cpu* est le temps de calcul total en secondes, *itr GC* est le nombre d'itérations de génération de colonnes nécessaires pour atteindre l'optimum, *col SP* est le nombre total de colonnes générées, et *itr MP* est le nombre total d'itérations du simplexe effectuées. Les résultats sont aussi présentés en pourcentage.

Tableau 3.1 – Stabilisation : résultats obtenus quand π^* est connu.

Méthode	Sol Init	cpu(s)	itr GC	col SP	itr MP
standard	800000000	4178.4	509	37579	926161
$\delta_+ - \delta_-$					
200.0	2035590.5	835.5	119	9368	279155
20.0	1927590.5	117.9	35	2789	40599
2.0	1915710.5	52.0	20	1430	8744
0.2	1915710.5	47.5	19	1333	8630
Sol. Opt.	1915589.5				
$\delta_+ - \delta_-$	Saut initial (%)	(%)	(%)	(%)	(%)
200.0	6.26	20.0	23.4	24.9	30.1
20.0	0.63	2.8	6.9	7.4	4.4
2.0	0.063	1.2	3.9	3.8	0.9
0.2	0.0063	1.1	3.7	3.5	0.9

Le tableau montre très clairement l'effet bénéfique que peut avoir la connaissance d'information très précise sur une solution optimale du dual (D) pour l'obtention d'une solution optimale primale de (P). Notons que dans la formulation de (SP_l) il n'est point question de π^* . Seulement les valeurs de δ_-^l et δ_+^l sont nécessaires. Donc si on peut encadrer assez précisément une solution optimale duale, l'obtention d'une solution optimale primale devient facile. Dans l'exemple précédent, une boîte de ± 100 autour d'une solution optimale accélère le temps de calcul par un facteur de 5 alors que le facteur atteint presque 100 pour une boîte à ± 0.1 .

3.4 Analyse des solutions des problèmes stabilisés dans le cas général

Considérons maintenant une fonction g^l quelconque et soit $(x^l, z^{+,l}, y^{+,l}, y^{-,l}, z^{-,l})$ et $(\pi^l, v^{+,l}, u^{+,l}, u^{-,l}, v^{-,l})$ les solutions optimales obtenues pour (SP_l) et (SD_l) . Nous allons étudier les propriétés de ces solutions ainsi que les informations qu'elles peuvent fournir. Notre étude sera basée sur la solution optimale π^l de (SD_l) obtenue à la fin de l'itération majeure l . Dépendamment de la position de π^l par rapport à la boîte $[\delta_-^l, \delta_+^l]$, nous déduirons soit l'optimalité des solutions x^* et π^l pour (P) et (D) , soit l'amélioration stricte de la borne inférieure $b^T \pi^l$.

La première proposition traite du cas trivial où x^l est réalisable pour (P) .

Proposition 3.4 : *Si $z^{+,l} = y^{+,l} = y^{-,l} = z^{-,l} = 0$, alors x^l et π^l sont des solutions optimales pour (P) et (D) respectivement.*

Preuve: Si $z^{+,l} = y^{+,l} = y^{-,l} = z^{-,l} = 0$, alors x^l est réalisable pour (P) . De plus, les contraintes de bornes supérieures sur les variables de stabilisation sont satisfaites strictement. Par complémentarité, $v^{+,l} = u^{+,l} = u^{-,l} = v^{-,l} = 0$. D'où π^l est réalisable pour (D) . Ainsi $c^T x^l = b^T \pi^l$ et on déduit que x^l et π^l sont optimales pour (P) et (D) , respectivement. ■

La proposition qui suit concerne un cas idéal pour la solution π^l , c'est-à-dire quand $\delta_-^l < \pi^l < \delta_+^l$.³

Proposition 3.5 : *Si π^l est tel que $\delta_-^l < \pi^l < \delta_+^l$, alors π^l et x^l sont des solutions*

³Cette proposition a été montrée pour le cas d'une pénalité à trois morceaux dans [61] et [64]. Le principe est le même dans notre cas.

optimales pour (D) et (P) respectivement.

Preuve: Si $\delta_-^l < \pi^l < \delta_+^l$, $v^{+,l} = u^{+,l} = u^{-,l} = v^{-,l} = 0$ et les contraintes reliant $\pi^l, v^{+,l}, u^{+,l}, u^{-,l}, v^{-,l}$ ne sont pas actives. Par la complémentarité linéaire, $z^{+,l} = y^{+,l} = y^{-,l} = z^{-,l} = 0$. Par conséquent, x^l et π^l sont réalisables pour (P) et (D) et vérifient $c^T x^l = b^T \pi^l$. Le résultat en découle directement. ■

Si une ou plusieurs composantes de π^l sont sur la frontière de la boîte $[\delta_-, \delta_+]$, alors les composantes correspondantes de $z^{+,l}, y^{+,l}, y^{-,l}, z^{-,l}$ peuvent ne pas être nulles. Dans ce cas, x^* n'est pas réalisable pour (P). Par contre, π^* peut être optimal pour (D). L'exemple suivant en donne une illustration.

Exemple 3.1 Supposons que $\varepsilon_+ > b > 0$ et $[\delta_-, \delta_+]$ touche le polyèdre des solutions optimales en δ_+ . δ_+ est alors la solution optimale obtenue pour (D). Dans ce cas, la solution primale de (SP_l) consistant à avoir $y^{+,*} = b$ et les autres variables nulles est optimale pour (SP_l) , mais $x^* = 0$ n'est pas réalisable pour (P). Cependant, d'après le théorème 3.1, aucune solution optimale duale ne peut se trouver à l'intérieur de $]\delta_-, \delta_+[$. ■

Regardons maintenant ce qu'on peut tirer comme information des solutions obtenues pour (SP_l) et (SD_l) si les conditions des propositions précédentes ne sont pas vérifiées. Une première information est que $b^T \pi^l$ est une borne inférieure sur la valeur optimale $v(P) = v(D)$.

Proposition 3.6 : Si π^l est une solution optimale de (SD_l) , alors

$$b^T \pi^l \leq v(D) = v^* = v(P),$$

c'est-à-dire que $b^T \pi^l$ est une borne inférieure sur $v(P)$.

Preuve: Le résultat découle directement du fait que π^* est réalisable pour (D) . ■

Le théorème suivant résume les résultats précédents et montre que si π^l n'est pas un multiplicateur optimal pour (P) , il est un meilleur multiplicateur pour (P) que tous les $\pi \in]\delta_-^l, \delta_+^l[$.

Théorème 3.2 : Soit $\hat{\pi} \in]\delta_-^l, \delta_+^l[$ réalisable pour (D) et π^l la solution optimale obtenue pour (SD_l) . Alors

i) si $\pi^l \in]\delta_-^l, \delta_+^l[$, alors π^l est une solution optimale de (D) .

ii) si $\pi^l \notin [\delta_-^l, \delta_+^l]$, alors

$$b^T \pi^l > b^T \hat{\pi}.$$

iii) si $\pi^l \in [\delta_-^l, \delta_+^l]$ mais $\pi^l \notin]\delta_-^l, \delta_+^l[$ (au moins une composante de π^l est sur la frontière), alors exactement l'une des deux propositions suivantes est vraie :

- π^l est optimal pour (D) .

- $b^T \pi^l > b^T \hat{\pi}$.

iv) si $\pi^l \notin]\delta_-^l, \delta_+^l[$, alors

$$\|\pi^l - \hat{\pi}\|_\infty \geq \Delta^l = \text{Min}\{\Delta_-, \Delta_+\} > 0$$

où

$$\begin{aligned} \Delta_+ &= \text{Min}_{i=1}^m \{\delta_{+,i}^l - \hat{\pi}_i\}, \\ \Delta_- &= \text{Min}_{i=1}^m \{\hat{\pi}_i - \delta_{-,i}^l\}. \end{aligned}$$

Preuve:

i) Le résultat découle directement de la proposition 3.5.

ii) π^l est un maximum de (SD_l) , donc

$$b^T \pi^l + g^l(\pi^l) \geq b^T \hat{\pi} + g^l(\hat{\pi}).$$

Mais $\hat{\pi} \in [\delta_-^l, \delta_+^l]$, donc $g^l(\hat{\pi}) = 0$. D'où

$$b^T \pi^l + g^l(\pi^l) \geq b^T \hat{\pi}.$$

Or, $\pi^l \notin [\delta_-^l, \delta_+^l]$. Donc, au moins une composante π_i^l est à l'extérieur de l'intervalle $]\delta_{-,i}^l, \delta_{+,i}^l[$. Ce qui veut dire que

$$g_i^l(\pi_i^l) < 0.$$

Par conséquent,

$$g^l(\pi^l) < 0.$$

D'où on peut déduire que

$$b^T \pi^l > b^T \hat{\pi}.$$

- iii) On sait que $g^l(\pi^l) = g^l(\hat{\pi}) = 0$ et $b^T \pi^l \geq b^T \hat{\pi}$. L'hypothèse $\pi^l \in [\delta_-^l, \delta_+^l]$ veut dire que la résolution de $((SP_l), (SD_l))$ est équivalente à l'utilisation d'une pénalité infinie à l'extérieur de la boîte $[\delta_-^l, \delta_+^l]$ (restriction du dual à la boîte $[\delta_-^l, \delta_+^l]$).

Pour démontrer les résultats escomptés, nous séparons les deux cas suivants :

- cas 1 : $b^T \pi^l > b^T \hat{\pi}$

Rien à démontrer.

- cas 2 : $b^T \pi^l = b^T \hat{\pi}$

Notons d'abord que dans ce cas, b (vecteur coût du dual) est orthogonal au segment $[\bar{\pi}, \pi^l] \subset \{\pi : A^T \pi \leq c\}$.

Étant donné la convexité du polyèdre $\{\pi : A^T \pi \leq c\}$, scit le segment $[\hat{\pi}, \pi^l]$ est sur une face optimale, soit il existe une direction de montée à partir d'un point du segment vers l'intérieur du polyèdre. Dans ce dernier cas et vu que $\delta_-^l < \hat{\pi} < \delta_+^l$, il existerait un point réalisable dans $]\delta_-^l, \delta_+^l[$ meilleur que π^l . Ce qui contredit l'optimalité de π^l pour (SD_l) .

On conclut alors que π^l et $\hat{\pi}$ sont des solutions optimales pour (D) .

- iv) Le résultat découle directement de la condition de *iv*). ■

Supposons qu'à l'itération majeure l , on a construit la fonction de pénalisation g^l autour du centre de stabilisation $\hat{\pi}^l$. Si π^l n'est pas un multiplicateur optimal, on a

nécessairement une croissance stricte de $b^T \pi$.

De plus, on s'assure qu'il y a eu un déplacement d'au moins

$$\Delta^l = \text{Min}\{\Delta_-^l, \Delta_+^l\} > 0$$

où

$$\begin{aligned}\Delta_+^l &= \text{Min}_{i=1}^m \{\delta_{+,i}^l - \hat{\pi}_i^l\}, \\ \Delta_-^l &= \text{Min}_{i=1}^m \{\hat{\pi}_i^l - \delta_{-,i}^l\}.\end{aligned}$$

On a maintenant en possession tous les résultats qu'il faut pour écrire un algorithme de stabilisation et prouver sa convergence dans le cadre d'une procédure de génération de colonnes.

3.5 Algorithme

L'algorithme donné ci-dessous représente la forme la plus simple de l'approche de stabilisation proposée. Nous y ferons référence par A_0 . Certaines étapes peuvent paraître triviales. Cependant elles sont incluses par souci de complétion et dans le but de pouvoir expliquer leurs détails et les modifications apportées dans la suite du chapitre. Nous ne présentons pas d'algorithme générique vu que mis à part quelques précisions spécifiées pour chaque étape, l'algorithme serait le même que A_0 .

L'algorithme effectue deux types d'itérations : majeures et mineures. Une itération majeure l consiste à résoudre la paire de problèmes $((SP_l), (SD_l))$ à l'optimalité. Les itérations mineures K sont les itérations de génération de colonnes nécessaires pour effectuer une itération majeure l . À la fin d'une itération majeure l , si x^l est réalisable pour (P) , l'optimalité est atteinte et l'algorithme s'arrête. L'algorithme s'arrête aussi si l'un des problèmes (SP_l) ou (SD_l) est non-réalisable ou non-borné.

Si les deux problèmes sont réalisables et finis, mais que l'optimalité n'est pas atteinte, le centre de stabilisation ainsi que la fonction de pénalisation sont mis à jour.

Algorithme A_0

0. Initialisation : Définir $\hat{\pi}^0$ et $g^l(\pi)$; $l = 0$.

1. Itération majeure l

Résoudre $((SP_l), (SD_l))$ par génération de colonnes :

1.1 itération mineure K :

Résoudre le K ème problème maître restreint $(RSP_{l,K})$

→ les solutions obtenues sont $(x^{l,K}, \pi^{l,K})$.

1.2 test d'arrêt de l'itération mineure k :

Résoudre le sous-problème avec le multiplicateur $\pi^{l,K}$

Si aucune colonne de coût réduit négatif n'est générée, **Retourner** x^l et π^l .

Aller à 2.

1.3 mise à jour du problème maître restreint :

Ajouter les colonnes générées au problème maître restreint → $(RSP_{l,K+1})$.

Aller à 1.1.

2. Test d'arrêt de l'algorithme

2.1 Si (SP_l) et (SD_l) sont réalisables et finis, alors

Si $z^{+,l} = y^{+,l} = y^{-,l} = z^{-,l} = 0$, x^l et π^l sont optimales pour (P) et (D) . **STOP**

Sinon, **Aller à 3.**

2.2 Sinon, (P) ou (D) est non-réalisable ou non-borné.

3. Déplacement et mise à jour de la pénalité

3.1 $\hat{\pi}^{l+1} = \pi^l$

3.2 Définir g^{l+1} .

3.3 **Aller à 1.**

3.5.1 Étude de la convergence

Nous allons traiter les quatre cas possibles :

- i) (P) et (D) réalisables et finis
- ii) (P) non-réalisable et (D) non-borné
- iii) (P) et (D) non-réalisables
- iv) (P) non-borné et (D) non-réalisable.

- i) Dans ce cas (D) possède au moins une solution optimale π^* telle que $b^T \pi^* < +\infty$.

Nous reprenons la condition (2.6) ici :

$$\exists \rho > 0 : \forall l, B(\hat{u}^l, \rho) \subset [\delta_-^l, \delta_+^l]. \quad (3.10)$$

En suivant le même raisonnement que la preuve du théorème 2.1, on prouve que sous l'hypothèse i) et la condition (3.10), l'algorithme converge vers une solution optimale duale π^* après un nombre fini d'itérations majeures.

Nous aurons, au bout d'un nombre fini d'itérations majeures l ,

$$\|\pi^* - \hat{\pi}^l\|_\infty < \rho.$$

Sachant que $B(\hat{u}^l, \rho) \subset [\delta_-^l, \delta_+^l]$, on peut affirmer que

$$\delta_-^l < \pi^* < \delta_+^l.$$

Et d'après le théorème 3.1, cette itération retournera x^l et π^l qui sont des solutions optimales pour (P) et (D) , respectivement.

- ii) Si (P) est non-réalisable et (D) est non-borné, (SP_l) et (SD_l) peuvent être réalisables et finis. Nous montrons dans ce qui suit que si la suite de fonctions g^l vérifie une certaine condition, l'algorithme pourra conclure correctement que (P) est non-réalisable et (D) est non-borné.

Nous traitons le cas où $b \neq [0, \dots, 0]^T$. Le cas trivial $b = [0, \dots, 0]^T$ ne pose aucun problème de ce genre, car le problème (P) est alors réalisable et (D) est alors borné.

Soit α_i (respectivement β_i) ($i = 1, \dots, m$) la limite de $(\varepsilon_{-,i}^l + \zeta_{-,i}^l)$ (respectivement $(\varepsilon_{+,i}^l + \zeta_{+,i}^l)$) quand $l \rightarrow \infty$.

Supposons que

$$\begin{aligned} \alpha_i &< -b_i, & b_i &< 0 \\ \beta_i &< +b_i, & b_i &> 0. \end{aligned} \tag{3.11}$$

Après un nombre fini d'itérations majeures, nous aurons

$$\begin{aligned} \varepsilon_{-,i}^l + \zeta_{-,i}^l &< -b_i, & b_i &< 0, \\ \varepsilon_{+,i}^l + \zeta_{+,i}^l &< +b_i, & b_i &> 0. \end{aligned}$$

Alors, (D) non-borné $\Rightarrow (SD_l)$ non-borné. (SP_l) est alors non-réalisable.

Ceci permet à l'algorithme de conclure correctement que (P) est non-réalisable et que (D) est non-borné.

iii) Si (P) et (D) sont non-réalisables, la première itération majeure donnera (SD_l) non-réalisable. Donc, soit que (SP_l) est non-réalisable soit qu'il est non-borné. Le deuxième cas voudrait dire que (P) est aussi non-borné. La première itération majeure donnera que (SP_l) et (SD_l) sont non-réalisables.

On conclut alors que (P) et (D) sont tous les deux non-réalisables.

iv) Si (P) est non-borné, il en est de même pour (SP_l) . Si (D) est non-réalisable, (SD_l) l'est aussi. La première itération majeure permettra alors de conclure que (P) est non-borné et que (D) est non-réalisable.

■

Le théorème suivant résume l'analyse de la convergence de l'algorithme qui vient d'être présenté.

Théorème 3.3 :

- i) Si (D) et (P) sont réalisables et finis, et si la condition (3.10) est vérifiée, alors l'algorithme A_0 termine en 2.1 avec des solutions optimales pour (P) et (D) .
- ii) Si (P) est non-réalisable et (D) est non-borné et si les α_i et β_i ($i = 1, \dots, m$) vérifient (3.11), alors l'algorithme A_0 termine en 2.2 avec la conclusion que (P) est non-réalisable et (D) est non-borné.
- iii) Si (P) et (D) sont non-réalisables, alors l'algorithme termine après la première itération majeure avec la conclusion que (P) et (D) sont non-réalisables.
- iv) Si (P) est non-réalisable et (D) est non-borné, alors l'algorithme termine après la première itération majeure à l'étape 2.2 avec la conclusion que (P) est non-réalisable et que (D) est non-borné.

■

3.5.2 Stratégies de mise à jour de la fonction de pénalité

Toutes les stratégies vérifiant les conditions du théorème 3.3 assurent la convergence de l'algorithme. Cependant, la question de savoir si (P) est réalisable ne présente aucune difficulté. Les problèmes résolus sont réalisables et finis. En fait, la non-réalisabilité du problème (P) est détectée facilement après un petit nombre d'itérations de génération de colonnes (phase I). Donc, l'implantation de l'algorithme doit tenir compte surtout du cas i). Plusieurs stratégies de mise à jour de g^l sont possibles. On peut se baser sur des idées intuitives et suivre l'une des lignes directrices suivantes :

- garder la fonction de pénalisation constante autour des centres de stabilité ;
- modifier g^l indépendamment de π^l ;
- modifier g^l en fonction de la position de π^l par rapport à la région de confiance.

Il ne faut pas négliger l'importance du problème à résoudre dans l'élaboration de la stratégie. Nous verrons un exemple quand nous aborderons l'application au problème de tournées de véhicules multi-dépôts.

Dans l'algorithme proposé, le nouveau centre de stabilisation $\hat{\pi}^{l+1}$ est toujours la solution optimale π^l de (SD_l) . Cependant, on tient compte seulement des paramètres de la fonction g^{l+1} dans l'itération majeure suivante. En effet, ce qui est important est que π^* soit à l'intérieur de la boîte $]\delta_-^l, \delta_+^l[$. Cette dernière condition peut n'être imposée qu'à partir d'un certain nombre d'itérations majeures. Ceci est vrai si on remplace l'étape 3.3 de A_0 par

$$3.1') \quad \hat{\pi}^{l+1} = \pi^l + \alpha_l d_l,$$

où d_l est une direction de déplacement et $\alpha_l \geq 0$ est le pas de déplacement correspondant. Ces suites doivent cependant vérifier

$$\|\alpha_l d_l\| \longrightarrow 0, \quad l \rightarrow \infty. \quad (3.12)$$

Nous donnons quelques exemples de directions intéressantes :

- $d_l = b$ et $\alpha_l \longrightarrow 0$.
- $d_l = \pi^l - \hat{\pi}^l$ et $\alpha_l \longrightarrow 0$.
- $d_l = (\pi^l - \hat{\pi}^l)^+$ et $\alpha_l \longrightarrow 0$.
- $d_l = b - Ax^l$ et $\alpha_l \longrightarrow 0$.
- $d_l = (b - Ax^l)^+$ et $\alpha_l \longrightarrow 0$.

Ceci nous permet d'énoncer le théorème suivant :

Théorème 3.4 : *Si on remplace l'étape 3.1 par 3.1' dans l'algorithme A_0 , en respectant la condition (3.12), alors les résultats de convergence sont les mêmes que pour le théorème 3.3. ■*

3.5.3 Quelques aspects de l'implantation

Initialisation :

L'algorithme converge quelque soit le vecteur initial $\hat{\pi}^0$. Même si $\hat{\pi}^0$ n'est pas réalisable, $\hat{\pi}^1$ sera réalisable. Et pour tout $l \geq 1$, $\hat{\pi}^l$ sera réalisable.

Le choix d'un vecteur initial assez proche de l'optimum permet d'accélérer la résolution de manière significative. Par contre, un mauvais choix nécessiterait plus d'itérations majeures et donc plus de temps de calcul.

Un choix trivial est le vecteur nul $(0, \dots, 0)^T$. Dans certains cas, ce vecteur est réalisable pour (D) . Il peut constituer une assez bonne approximation initiale comme tout vecteur réalisable de (D) . Il faut cependant noter que pour l'objectif, il présente une erreur relative de 100%, et pour des problèmes où l'ordre de l'objectif est grand, il est généralement très loin de la face optimale.

Si on peut obtenir une estimation, réalisable ou non, d'un optimum de (D) par une approche heuristique, la résolution peut être accélérée significativement. On peut aussi obtenir juste un encadrement heuristique, vu que la valeur de $\hat{\pi}^0$ n'intervient pas dans l'expression de g^0 .

Arrêt prématuré du processus de génération de colonnes :

Vu que la définition de g^l autour de $\hat{\pi}^l$ utilise une boîte d'intérieur non vide, il n'est pas nécessaire de se rendre exactement jusqu'à l'optimalité de $((SP_l), (SD_l))$ avant la fin de l'algorithme, notamment lors des premières itérations majeures. La première itération majeure en particulier peut nécessiter beaucoup d'itérations mi-

neures et de temps de calcul. Il serait bénéfique de fixer un nombre maximum d'itérations mineures par itération majeure. Les dernières itérations majeures prennent très peu d'itérations mineures et donc, l'algorithme convergera en pratique avec cette règle. On peut également imposer un nombre maximum d'itérations mineures pour les premières itérations majeures seulement.

Gestion des colonnes de stabilisation :

La procédure de stabilisation proposée vise à résoudre efficacement des problèmes ayant un nombre élevé m de contraintes. (SP_l) aurait alors $4m$ colonnes de stabilisation présentes dans le problème maître restreint. Ceci peut allourdir la résolution du simplexe durant les itérations mineures. Plusieurs idées peuvent être exploitées pour éviter ces inconvénients éventuels.

- **utilisation d'un seul côté de la pénalité** : Pour une itération majeure l , si la composante π_i^l augmente, la pénalisation du côté gauche est "inutile". Seule la pénalisation du côté droit a influencé l'optimum de (SD_l) . Ainsi si on utilise seulement un côté de la pénalité, on diminuera le nombre de colonnes de stabilisation. Ceci peut être implanté en vérifiant si la composante π_i^l évolue dans le même sens pour un certain nombre d'itérations majeures successives.
- **élimination des colonnes non-utilisées** : Si à la fin d'une itération majeure l , $\pi_i^l \in]\delta_{-,i}^l, \delta_{+,i}^l[$, on peut estimer que la composante est stabilisée, au moins momentanément, et éliminer toutes les variables de stabilisation associées. Bien sûr si on remarque une grande variation dans la valeur de π_i^l dans les itérations suivantes, la stabilisation serait réintroduite dans le problème. Ceci peut être appliqué pour les autres intervalles, mais il faut tenir compte des valeurs des pénalités correspondantes.
- **élimination des variables de stabilisation inutilisées** : Nous avons noté que le nombre de variables de stabilisation utilisées dans les solutions optimales

des (SP_l) décroît strictement avec les itérations majeures l . Ce qui suggère d'éliminer toutes ou certaines colonnes non-utilisées par la solution optimale de (SP_l) .

Mise à jour de la fonction g^l :

Une modification de la fonction g^l modifie le problème (SP_l) qui est résolu à chaque itération majeure. La réoptimisation du problème maître restreint pourrait être coûteuse en temps de calcul.

Pour ce qui est des coûts des colonnes de stabilisation, ils sont définis autour du nouveau centre de stabilisation. Ils seront donc modifiés à chaque itération majeure. Concernant les bornes supérieures sur les colonnes de stabilisation, on a le choix de les modifier ou non.

3.6 Conclusion

Nous avons proposé une approche de stabilisation des algorithmes de génération de colonnes et de Kelley à l'aide d'une fonction de pénalité linéaire par morceaux du dual. Ceci permet de préserver la linéarité des problèmes à résoudre à chaque itération de génération de colonnes. Nous résolvons une suite de problèmes stabilisés dont les solutions convergent vers une paire de solutions primale et duale.

Nous avons d'abord montré que si le pavé de confiance contient strictement un multiplicateur optimal, la résolution des problèmes stabilisés est équivalente à la résolution des problèmes originaux. L'application de ce résultat à la résolution d'une instance de *MDVSP* par génération de colonnes a démontré une très grande effi-

cacité. Cette efficacité était prévisible grâce aux résultats théoriques liés au saut d'optimalité initial et aux colonnes générées durant cette résolution. Ensuite, nous avons montré que la résolution des problèmes stabilisés fournit un vecteur de multiplicateurs meilleur que le précédent, s'il n'est pas optimal. Ceci nous a permis de prouver la convergence de l'algorithme que nous avons proposé, indépendamment du vecteur initial de multiplicateurs. Il est important de noter que le fait que la région de confiance contient le centre de stabilité dans son intérieur est indispensable pour la convergence de la méthode vers une solution primale du problème original. Nous avons également expliqué pourquoi la résolution des problèmes stabilisés devrait être plus facile que celle des problèmes originaux. Enfin, nous avons proposé plusieurs stratégies de mise à jour de la fonction de pénalité à chaque itération majeure qui assurent la convergence de l'algorithme ainsi que plusieurs techniques d'accélération. En particulier, nous avons montré comment la direction de montée peut être enrichie par n'importe quelle autre direction sans compromettre la convergence de l'algorithme. Cependant, il faut choisir ces directions de façon qui tend à augmenter l'objectif dual ou à se rapprocher d'une solution optimale pour espérer accélérer la convergence de l'algorithme en pratique.

L'application de l'approche de stabilisation proposée au problème de tournées de véhicules multi-dépôts (*MDVSP*) fera l'objet d'un chapitre dans la suite de la thèse. Nous y montrerons l'apport bénéfique de la stabilisation pour une résolution plus efficace de ce problème.

Chapitre 4

Inégalités valides sur le polyèdre optimal dual

L'approche que nous traitons dans ce chapitre se base sur la même idée que les inégalités valides utilisées pour les problèmes à variables entières. L'objectif de la résolution du problème est de trouver un point du polyèdre des solutions optimales duales (avec une solution optimale primale bien sûr). Ainsi, si nous connaissons une inégalité valide sur ce polyèdre ou un sous-ensemble non-vide de ce polyèdre, on peut l'ajouter à la formulation du dual tout en garantissant l'obtention d'une solution optimale du dual original et de la même valeur optimale. Ceci permettra de restreindre le domaine dual et par conséquent, réduira l'ensemble des multiplicateurs possibles durant la génération de colonnes. Cependant, la réalisabilité primale n'est plus garantie vu que le domaine du primal est relaxé. Une difficulté importante qui doit être surmontée par une méthode utilisant des inégalités valides optimales-duales est de trouver une façon d'assurer l'obtention d'une solution optimale primale de base.

Valério de Carvalhò 2000 [88] a proposé l'utilisation de coupes duales (*Dual Cuts*) afin d'accélérer la résolution du problème de découpe unidimensionnelle (*CSP*). La notion de validité des coupes est déduite des travaux de Gilmore et Gomory 60 [28] sur le même problème. Elle est introduite d'un point de vue primal et aucun lien n'est fait avec le polyèdre des solutions optimales duales. De plus, contrairement à ce qu'affirme l'auteur, le fait d'avoir le second membre des contraintes nul ne garantit

pas la valeur optimale pour les problèmes originaux et les problèmes modifiés. Nous reviendrons sur ces travaux quand nous traiterons la stabilisation de *CSP*.

Nous analysons deux cas de figure possibles des inégalités valides sur le polyèdre optimal dual. Le premier correspond à la restriction du polyèdre des solutions optimales duales. C'est le cas où l'on sait qu'il existe au moins une solution optimale duale qui vérifie certaines contraintes. Ces inégalités sont appelées inégalités partiellement valides duales-optimales (*Dual-Optimal Partial Valid Inequalities*). Nous les modélisons par des contraintes linéaires d'égalité. Le deuxième cas de figure correspond à une meilleure connaissance du problème. Les contraintes ajoutées sont vérifiées par tout le polyèdre des solutions optimales duales. Elles sont appelées inégalités valides duales-optimales (*Dual-Optimal Valid Inequalities*). Nous utiliserons des contraintes d'inégalités, et ainsi nous aurons traité les deux types de contraintes linéaires possibles.

Nous analysons d'abord l'effet de l'ajout des contraintes dans le dual sur la résolution du problème dual et sur la résolution du primal. Pour le premier type d'inégalités valides, nous montrons la convergence vers une solution optimale duale et proposons une méthode générale pour calculer une solution optimale primale de base. Cette méthode peut être appliquée aussi au deuxième type de contraintes. Nous proposons aussi une modification des contraintes ajoutées au dual qui profite de la qualité de l'information duale disponible afin d'obtenir directement une solution optimale primale de base. La fin du chapitre est consacrée à la discussion de l'implantation des contraintes afin de garantir une résolution la plus efficace possible.

4.1 Inégalités partiellement valides duales-optimales

Nous considérons la formulation d'un programme linéaire avec contraintes d'égalité et son dual. Ceci correspond à l'application au problème de découpe binaire (*BCSP*) que nous traitons dans les chapitres suivants. Nous redéfinissons les formulations considérées ici par souci de complétion. Le primal sera noté (P) et le dual (D) .

Il s'agit d'un cas particulier de la fonction de pénalité définie dans l'approche unifiée. La région de confiance sera définie à l'aide d'égalités vérifiées par certaines solutions optimales duales. L'objectif de la stabilisation est alors de guider ou de pousser les multiplicateurs vers cet ensemble de solutions optimales duales. Ceci permettra d'éviter certaines oscillations dans l'évolution des multiplicateurs duaux durant la génération de colonnes, en plus de restreindre l'espace des solutions duales admissibles. L'approche sera présentée dans une version très simple par souci de clarté. Plus loin dans le chapitre, nous reviendrons sur les aspects plus complexes reliés à l'implantation.

Nous commençons d'abord par la présentation détaillée de cette approche de stabilisation ainsi que l'effet sur les problèmes (P) et (D) . Ensuite, nous étudions la convergence du processus de génération de colonnes vers des solutions des problèmes originaux.

Dans certains problèmes modélisés par (P) , certaines solutions optimales de (D) ont des propriétés particulières. L'idée que nous proposons ici consiste à utiliser ces propriétés afin d'accélérer le processus de convergence vers des solutions optimales primale et duale. Nous modélisons ces propriétés sous la forme de contraintes

d'égalité. D'autres formes sont, bien sûr, possibles mais, cela dépend des propriétés considérés et des problèmes étudiés.

L'approche de stabilisation proposée consiste à ajouter ces contraintes d'égalité dans la formulation du dual (D). Ceci se traduira alors par la restriction du domaine de (D). Mais, il est important de noter que l'ensemble de solutions optimales de (D) peut aussi être restreint. Nous présentons cette approche sous forme statique, c'est-à-dire que toutes les contraintes seront ajoutées à la formulation du dual au début de la résolution. Les problèmes stabilisés seront notés (SP) et (SD).

4.1.1 Effets sur les problèmes originaux

Supposons que les contraintes que nous considérons sont définies à l'aide d'une matrice E et d'un vecteur colonne e de la manière suivante :

$$E^T \pi = e.$$

La fonction de pénalisation correspondante est définie par

$$g(\pi) = \begin{cases} 0, & E^T \pi = e \\ +\infty, & E^T \pi \neq e \end{cases}$$

Les formulations résultantes pour les problèmes stabilisés sont données par :

(SP) $\begin{aligned} & \text{Min } c^T \tilde{x} + e^T y \\ \text{sc } & \begin{cases} A\tilde{x} + Ey = b \\ \tilde{x} \geq 0 \end{cases} \end{aligned}$	$ $	(SD) $\begin{aligned} & \text{Max } b^T \tilde{\pi} \\ \text{sc } & \begin{cases} A^T \tilde{\pi} \leq c \\ E^T \tilde{\pi} = e \end{cases} \end{aligned}$
--	-----	--

Si les contraintes ajoutées sont trop générales, le domaine dual ne changera pas. Afin que le domaine dual soit restreint, les contraintes ajoutées doivent être vérifiées seulement par un sous-ensemble des solutions duales. Dans le cas de contraintes

vérifiées par un sous-ensemble des solutions optimales duales, l'ensemble des solutions optimales ainsi que le domaine réalisable seront restreints. Ces inégalités sont appelées *inégalités partiellement valides duales-optimales*. Il est important de noter qu'il n'est pas sûr que de telles inégalités restreignent l'ensemble des solutions optimales duales. En effet, c'est l'information disponible qui est incomplète.

La figure 4.1 montre de quelle manière le domaine réalisable dual est restreint sur un exemple dans \mathbb{R}^3 . Le polyèdre représenté est l'ensemble des solutions optimales duales. L'hyperplan qui supporte la contrainte ajoutée coupe le domaine dual et aussi le polyèdre des solutions optimales.

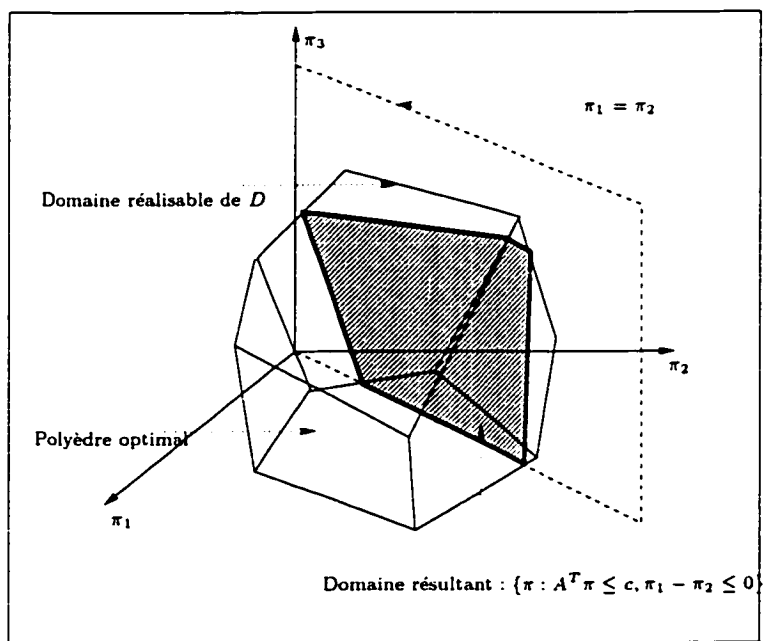


Figure 4.1 – Restriction du domaine réalisable dual par des contraintes d'égalité valides sur un sous-ensemble du polyèdre optimal dual.

La figure 4.2 montre comment la contrainte

$$\pi_1 = \pi_2$$

coupe le polyèdre des solutions optimales D^* . Ce polyèdre est représenté dans le plan $\pi_1\pi_2$ dans le but d'avoir une meilleure représentation.

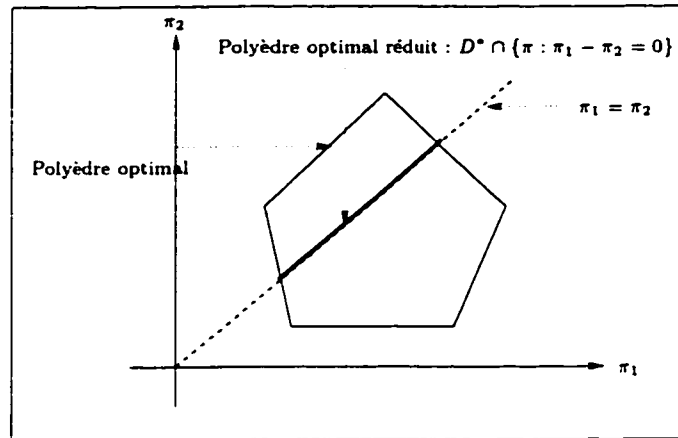


Figure 4.2 – Restriction du polyèdre optimal dual par des contraintes d'égalité valides sur un sous-ensemble des solutions optimales duales.

Il faut noter également que puisque les contraintes ajoutées restreignent l'ensemble des solutions duales optimales des problèmes maîtres, l'ensemble des multiplicateurs possibles durant la génération de colonnes est également restreint.

L'ajout de ces contraintes au dual résulte en l'ajout de variables (colonnes statiques) au même nombre que ces contraintes dans le primal. Ces variables peuvent être positives ou négatives et leur coût est donné par e . Ceci peut affecter la réalisabilité de la solution primale obtenue à la fin de la résolution de (SP) et (SD) . Nous reviendrons sur ce point dans la section suivante.

4.1.2 Étude de la convergence

Nous considérons la résolution des problèmes stabilisés (SP) et (SD) . Nous supposons évidemment que l'ensemble $\{\pi \in \mathbb{R}^m : A^T \pi \leq c, E^T \pi = e\}$ n'est pas vide. En fait, cet ensemble, par la définition de la matrice E et du vecteur e , contient au moins une solution optimale de (D) . Ceci sous-entend aussi que (P) et (D) sont réalisables et finis.

La première proposition énonce le résultat trivial qui dit que si les contraintes ajoutées sont vérifiées par au moins une solution optimale, le dual original et le dual stabilisé ont la même valeur optimale.

Proposition 4.1 : *Si l'ensemble $D^* \cap \{\pi : E^T \pi = e\} \neq \emptyset$, alors (D) et (SD) ont la même valeur optimale et toute solution optimale de (SD) est optimale pour (D) . ■*

Les propositions qui suivent montrent des cas particuliers où la solution optimale obtenue pour (SP) fournit aussi une solution optimale au problème (P) .

Proposition 4.2 : *Si la solution optimale (\tilde{x}^*, y^*) obtenue pour (SP) vérifie $y^* = 0$, alors \tilde{x}^* est une solution optimale pour (P) . ■*

Proposition 4.3 : *Soit (\tilde{x}^*, y^*) la solution optimale obtenue pour (SP) . Si on peut obtenir à partir de cette solution un vecteur x^* réalisable pour (P) dont le coût vérifie $c^T x^* \leq c^T \tilde{x}^* + e^T y^*$, alors x^* est une solution optimale pour (P) .*

Preuve: Notons d'abord que toute solution réalisable pour (P) est aussi réalisable pour (SP) . Donc

$$c^T \tilde{x}^* + e^T y^* \leq v(P).$$

Puisque x^* est réalisable pour (P) et $c^T x^* \leq c^T \tilde{x}^* + e^T y^*$, on déduit que

$$c^T x^* \leq v(P),$$

et par conséquent, x^* est optimal pour (P) . ■

La proposition qui suit résume les précédentes et donne un cas général où la résolution de (SP) et (SD) fournit des solutions optimales pour (P) et (D) .

Proposition 4.4 : *Si la matrice E et le vecteur e vérifient*

i) l'ensemble des solutions optimales de (D) vérifiant $E^T \pi = e$ n'est pas vide

ii) de toute solution (\tilde{x}, y) réalisable pour (SD) on peut déduire une solution x réalisable pour (P) de moindre coût (i.e. $c^T \tilde{x} + e^T y \geq c^T x$),¹

alors, en notant (\tilde{x}^, y^*) et $\tilde{\pi}^*$ les solutions optimales obtenues pour (SP) et (SD) , on peut affirmer que :*

i) $\tilde{\pi}^$ est une solution optimale de (D)*

ii) la solution x^ obtenue de (\tilde{x}^*, y^*) suivant ii) est optimale pour (P) . ■*

La deuxième condition du problème paraît forte, mais en pratique pour certains problèmes, les contraintes ajoutées au dual se traduisent dans le primal par des colonnes très particulières et qui ont un lien très étroit avec les colonnes générées par l'oracle. Nous verrons ceci dans le cas des problèmes *BCSP* et *CSP* dans les chapitres qui suivront.

¹V.de Carvalhò utilise une telle condition pour montrer la validité des coupes duales qu'il propose pour *CSP* [88].

Si cette condition n'est pas vérifiée, il est possible de retrouver une solution optimale primale à l'aide du théorème 3.1. Il suffit alors de construire une boîte très petite autour de la solution optimale duale obtenue et de résoudre le problème restreint correspondant.

Si $\tilde{\pi}^*$ est la solution optimale obtenue pour (SD) et $\eta \in \mathbb{R}^m, \eta > 0$ raisonnablement petit, le nouveau primal stabilisé et son dual sont formulés comme suit :

$$\begin{array}{c}
 (NSP) \\
 Min \quad c^T \tilde{x} - (\tilde{\pi}^* - \eta)^T y^- + (\tilde{\pi}^* + \eta)^T y^+ \\
 sc \left\{ \begin{array}{l} A\tilde{x} - y^- + y^+ = b \\ \tilde{x}, y^+, y^- \geq 0 \end{array} \right. \\
 \\
 \text{-----} \\
 (NSD) \\
 Max \quad b^T \tilde{\pi} \\
 sc \left\{ \begin{array}{l} A^T \tilde{\pi} \leq c \\ \tilde{\pi}^* - \eta \leq \tilde{\pi} \leq \tilde{\pi}^* + \eta. \end{array} \right.
 \end{array}$$

Le théorème qui suit énonce le fait que la résolution de la paire $((NSP), (NSD))$ fournit une solution optimale de (P) .

Théorème 4.1 : Soit (\tilde{x}^*, y^*) et $\tilde{\pi}^*$ les solutions optimales obtenues pour (SP) et (SD) . Et soit $(\tilde{x}^*, y^{-,*}, y^{+,*})$ une solution optimale de (NSP) .

Alors, $y^{-,*} = y^{+,*} = 0$ et \tilde{x}^* est optimale (de base) pour (P) .

Preuve: Comme $\eta > 0$, la stabilisation correspond à une fonction de pénalisation du type introduit dans le chapitre 3. De plus, $\tilde{\pi}^*$ qui est une solution optimale de (D) (d'après la proposition 4.1) est strictement compris dans la boîte utilisée. D'après le théorème 3.1, toute solution optimale de (NSP) , en particulier \tilde{x}^* , est optimale (de base) pour (P) . ■

Cette deuxième phase est très efficace (*cf.* chapitre 5), d'autant plus que le problème maître est déjà riche en colonnes générées lors de la résolution de (SP) et (SD) .

Nous reviendrons à la gestion dynamique des contraintes ajoutées ainsi que leur interprétation plus loin dans le chapitre après la présentation du cas où les contraintes ajoutées sont vérifiées par toutes les solutions optimales duales.

4.2 Inégalités valides duales-optimales

L'approche de stabilisation proposée dans cette section est très proche de la précédente. Elle est moins générale dans le sens où elle vise tout l'espace dual optimal au lieu d'un sous-ensemble. Elle s'applique quand l'information duale disponible est plus complète. Vu que ces inégalités sont vérifiées par toutes les solutions optimales duales, elles seront appelées *inégalités valides duales-optimales*. Ces inégalités peuvent être traitées de la même manière que les égalités dans la section précédente. Cependant, dans la présente approche, nous profitons de la complétion de l'information afin d'obtenir une solution optimale primale directement. Nous redéfinissons les problèmes primal et dual (P) et (D) de la même manière.

$$\begin{array}{cc}
 (P) & \qquad \qquad \qquad (D) \\
 \begin{array}{l} \text{Min } c^T x \\ \text{sc } \left\{ \begin{array}{l} Ax = b \\ x \geq 0 \end{array} \right. \end{array} & \left| \begin{array}{l} \text{Max } b^T \pi \\ \text{sc } \left\{ \begin{array}{l} A^T \pi \leq c \end{array} \right. \end{array} \right.
 \end{array}$$

L'approche de stabilisation sera présentée de manière simple et les aspects plus complexes reliés à l'implantation seront discutés à la fin de la présentation. Nous commencerons par la présentation de l'idée de stabilisation et de son effet sur les formulations de (P) et (D) . Ensuite, les résultats de convergence sont présentés.

Le polyèdre des solutions optimales duales est généralement petit par rapport au polyèdre des solutions réalisables. La résolution du problème se fait sur ce dernier et peut prendre en considération des multiplicateurs très éloignés du polyèdre optimal.

Si on dispose d'inégalités valides sur le polyèdre optimal, mais qui ne sont pas vérifiées par une partie du domaine réalisable, on peut limiter le nombre de multiplicateurs intermédiaires possibles dans une procédure de génération de colonnes. Et plus ces inégalités sont précises (serrées), plus on évite des itérations inutiles.

Ainsi, si on peut prouver que certaines contraintes sont satisfaites par l'ensemble des solutions duales optimales et qu'elles ne sont pas redondantes, on peut les ajouter à la formulation du dual afin de stabiliser l'évolution des multiplicateurs duaux pendant la génération de colonnes.

4.2.1 Effets sur les problèmes originaux

Soient la matrice E ($m \times q$) et le vecteur $e \in \mathbb{R}^q$ qui définissent les k contraintes qu'on désire ajouter à la formulation du dual. Nous supposons que toute solution optimale duale π vérifie

$$E^T \pi \leq e,$$

et que ces contraintes ne sont pas redondantes par rapport aux contraintes définies par $A^T \pi \leq c$. L'ajout de ces contraintes au dual revient à utiliser une pénalité linéaire constante définie par la fonction suivante :

$$g(\pi) = \begin{cases} 0, & E^T \pi \leq e \\ +\infty, & \text{sinon.} \end{cases}$$

Les formulations résultantes pour les problèmes stabilisés sont données par :

$$\begin{array}{cc}
 (SP) & (SD) \\
 \hline
 \begin{array}{l} \text{Min } c^T \tilde{x} + e^T y \\ \text{sc } \left\{ \begin{array}{l} A\tilde{x} + Ey = b \\ \tilde{x}, y \geq 0 \end{array} \right. \end{array} & \begin{array}{l} \text{Max } b^T \tilde{\pi} \\ \text{sc } \left\{ \begin{array}{l} A^T \tilde{\pi} \leq c \\ E^T \tilde{\pi} \leq e \end{array} \right. \end{array}
 \end{array}$$

Nous supposons que ces contraintes réduisent le domaine réalisable dual de manière assez significative car dans le cas contraire elles seraient inutiles, voire nuisibles. Ceci est le cas si ces contraintes sont des faces ou des facettes du domaine réalisable et/ou du polyèdre des solutions optimales duales.

La figure 4.3 montre un exemple où la contrainte $\pi_3 - 4\pi_2 \leq 0$ est vérifiée par toutes les solutions optimales et coupe le domaine dual considérablement. On voit clairement comment l'ensemble des solutions réalisables duales est réduit, alors que le polyèdre optimal dual est intact. La coupe ajoutée permet aussi de réduire l'ensemble des multiplicateurs duaux possibles durant la génération de colonnes.

L'ajout de ces contraintes au dual se traduit par l'ajout de colonnes au primal. Ces colonnes, dont les variables correspondantes doivent être non-négatives, joueront le même rôle que dans la section précédente. Elles donnent plus de flexibilité au primal et, en général, leur présence peut affecter la réalisabilité du primal.

Enfin il est important de noter que la résolution se fait en une seule itération majeure (cf. chapitre 3). C'est-à-dire qu'il s'agit de résoudre les problèmes stabilisés par génération de colonnes.

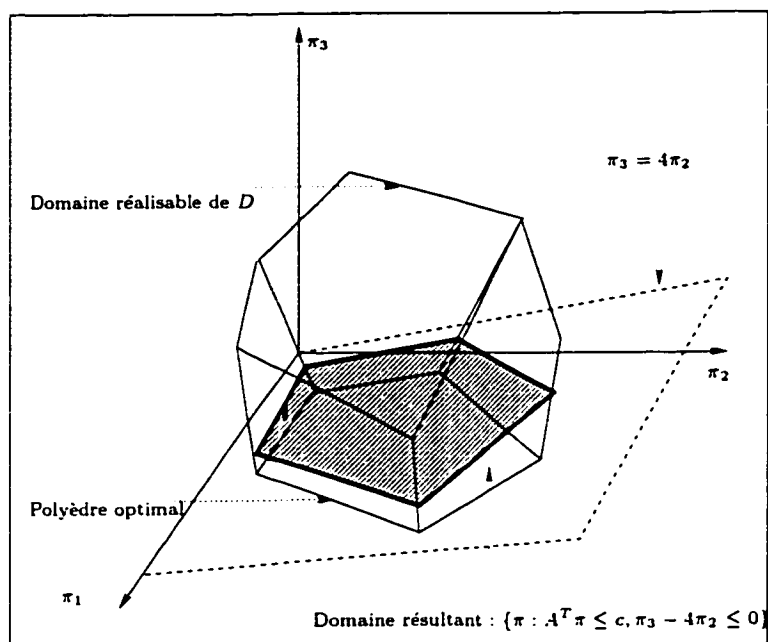


Figure 4.3 – Restriction du domaine réalisable à l'aide une inégalité valides sur le polyèdre optimal dual.

4.2.2 Étude de la convergence

Nous supposons que (P) et (D) sont réalisables et finis. Nous notons D^* le polyèdre des solutions optimales de (D) . Une hypothèse principale est que l'ensemble $\{\pi \in \mathbb{R}^m : A^T \pi \leq c, E^T \pi \leq e\}$ soit non-vide. Ainsi, (SD) sera réalisable et fini, de même que (SP) . Cette hypothèse découle directement du fait que toute solution optimale duale vérifie $E^T \pi \leq e$.

Nous énonçons d'abord une proposition sur l'égalité de la valeur optimale des problèmes originaux et celle des problèmes stabilisés dans le cas où l'ensemble des contraintes ajoutées au dual est tel que précisé plus haut.

Proposition 4.5 : Si $D^* \subset \{\pi \in \mathbb{R}^m : E^T \pi \leq e\}$, alors (D) et (SD) ont le même

ensemble de solutions optimales et la même valeur optimale. ■

Ainsi, si les contraintes ajoutées sont des inégalités valides sur le polyèdre des solutions optimales duales D^* , la résolution de la paire $((SP), (SD))$ fournit la même valeur optimale que la résolution de la paire $((P), (D))$. De plus, les solutions optimales $\tilde{\pi}^*$ de (SD) sont toutes des solutions optimales pour (D) et vice versa.

Par contre, la solution \tilde{x}^* de (SP) n'est pas forcément réalisable pour (P) , et par conséquent, peut ne pas être une solution optimale. Mais, si y^* est tel que (\tilde{x}^*, y^*) est une solution optimale de (SP) , on a l'égalité

$$c^T \tilde{x}^* + e^T y^* = v(P).$$

De plus, toute solution réalisable x de (P) est réalisable pour (SP) . Donc,

$$c^T \tilde{x}^* + e^T y^* \leq c^T x.$$

On peut alors énoncer la proposition suivante qui affirme que si \tilde{x}^* est réalisable pour (P) , il en est une solution optimale.

Proposition 4.6 : *Soit (\tilde{x}^*, y^*) la solution optimale obtenue pour (SP) . Si $y^* = 0$, alors \tilde{x}^* est une solution optimale pour (P) .* ■

La proposition qui suit donne un cas particulier où on peut obtenir une solution optimale de (P) à partir d'une solution optimale de (SP) . Les mêmes arguments qui ont précédé la proposition précédente permettent de prouver cette proposition.

Proposition 4.7 : *Soit (\tilde{x}^*, y^*) la solution optimale obtenue pour (SP) . Si on peut construire x^* réalisable pour (P) et vérifiant $c^T x^* \leq c^T \tilde{x}^* + e^T y^*$, alors x^* est une solution optimale pour (P) .* ■

La proposition qui suit résume les précédentes et donne un cas général où la résolution de (SP) et (SD) fournit des solutions optimales pour (P) et (D) .

Proposition 4.8 : *Si la matrice E et le vecteur e vérifient :*

- i) l'ensemble des solutions optimales de (D) vérifiant $E^T \pi \leq e$ n'est pas vide*
 - ii) de toute solution (\tilde{x}, y) réalisable pour (SD) , on peut déduire une solution x réalisable pour (P) de moindre coût (i.e. $c^T \tilde{x} + e^T y \leq c^T x$),*
- alors, en notant (\tilde{x}^*, y^*) et $\tilde{\pi}^*$ les solutions optimales obtenues pour (SP) et (SD) , on peut affirmer que :*

- i) $\tilde{\pi}^*$ est une solution optimale de (D)*
- ii) la solution x^* obtenue de (\tilde{x}^*, y^*) suivant ii) est optimale pour (P) . ■*

Les mêmes remarques faites pour la première approche reviennent ici. Ainsi, les colonnes ajoutées au primal lui procurent une certaine flexibilité, ce qui fait que sa résolution est plus facile. Mais, il y a aussi l'inconvénient que la solution obtenue pour (SP) peut ne pas être réalisable pour (P) . Le théorème 3.1 peut être utilisé afin d'obtenir une solution optimale de (P) de la même manière que pour les inégalités partiellement valides duales-optimales.

Cependant, pour cette approche l'information disponible est plus complète. Nous pouvons en profiter afin d'avoir un résultat plus fort pour la convergence vers une solution optimale primale.

Le théorème qui suit donne une façon d'appliquer cette approche de stabilisation tout en assurant l'obtention de solutions optimales de (P) et (D) sans hypothèse supplémentaire.

Théorème 4.2 : *Supposons que la matrice E et le vecteur e vérifient*

$$D^* \subset \{\pi \in \mathbb{R}^m : E^T \pi \leq e\}.$$

Si les contraintes ajoutées au dual sont de la forme $E^T \pi \leq e + \eta$, où $\eta > 0$ (raisonnablement petit), alors

toute solution (\tilde{x}^, y^*) optimale pour (SP) vérifie $y^* = 0$ et \tilde{x}^* est une solution optimale pour (P) .*

Preuve: Soient (\tilde{x}^*, y^*) et $\tilde{\pi}^*$ les solutions optimales primale et duale obtenues pour (SP) et (SD) avec les contraintes décrites dans l'énoncé du théorème. Comme $\tilde{\pi}^*$ est optimale pour (D) , alors

$$E^T \tilde{\pi}^* \leq e < e + \eta.$$

Par complémentarité, $y^* = 0$. Donc, \tilde{x}^* est réalisable et optimal pour (P) . ■

Remarque : Dans le cas de contraintes d'égalité, c'est-à-dire

$$E^T \pi = e,$$

on doit les modifier de la manière suivante :

$$e - \eta \leq E^T \pi \leq e + \eta,$$

où $\eta > 0$. Bien sûr, il faut noter que le nombre de contraintes (et colonnes) ajoutées est doublé. Cependant, dans le cadre d'une implantation efficace, ceci ne devrait pas poser de problèmes à l'efficacité de la stabilisation. ■

4.3 Remarques générales

1. Les deux approches étudiées sont similaires mathématiquement. Elles consistent toutes les deux à ajouter un ensemble de contraintes linéaires au dual dans le but de restreindre son domaine réalisable. Nous avons introduit chacune sur un type de contrainte différent (égalité ou inégalité) afin d'avoir une vue plus complète sur la modification du problème. L'approche développée pour les inégalités partiellement valides duales-optimales peut être utilisée pour des contraintes d'égalité valides sur tout le polyèdre optimal dual. Sous les conditions de la proposition 4.4, cette approche permet d'ajouter la moitié du nombre de colonnes au problème maître que la deuxième approche.

Toutefois, les idées derrière ces deux approches les différencient de manière claire. La première consiste à restreindre le polyèdre optimal dual. C'est le cas quand on réussit seulement à prouver qu'il existe des solutions optimales duales qui vérifient un ensemble de contraintes donné. Pour la deuxième approche, les contraintes sont vérifiées par toutes les solutions optimales duales. Ceci permet d'obtenir des solutions optimales pour le dual et le primal dès la première optimisation. Ceci n'est pas le cas pour la première approche qui nécessite une autre optimisation. Cette deuxième optimisation est néanmoins efficace comme nous l'avons déjà noté.

2. L'application de l'approche de stabilisation proposée est conditionnée par la connaissance de propriétés d'un sous-ensemble des solutions optimales duales exprimées par la matrice E et le vecteur e . C'est le cas pour quelques problèmes pratiques. Nous traiterons les problèmes *BCSP* et *CSP* dans les chapitres qui suivent et montrerons que la connaissance d'une seule propriété permet d'accélérer considérablement la résolution des problèmes.
3. L'utilisation de l'une ou l'autre de ces deux approches dépend tout d'abord du type d'information disponible. Si on peut prouver que les contraintes sont

des inégalités valides sur le polyèdre optimal dual, on utilisera la deuxième approche. Si on peut prouver seulement l'existence de variables duales vérifiant les contraintes, ce sera la première approche qu'il faudra utiliser.

Cependant, il est plus probable que, pour un même problème, les contraintes du deuxième type (inégalités valides duales-optimales) soient moins serrées que celle du premier type. Alors, dépendamment de la difficulté du problème, le choix de l'approche à utiliser doit être déterminé après avoir pris en compte plusieurs aspects reliés à l'optimisation du problème en question.

4. Le type de contraintes utilisées peut varier selon le problème résolu et selon les variables duales visées. Dans certains cas, ceci permettrait de briser la symétrie existante entre plusieurs solutions optimales duales. Ainsi, les multiplicateurs "intermédiaires" possibles ne seront plus quelconques et l'ajustement de ces derniers se fera tout en éliminant les symétries du dual.
5. Il est généralement possible que le nombre de contraintes soit très élevé. Cependant, toutes ces contraintes ne jouent pas le même rôle dans l'accélération de la convergence. Il serait intéressant soit d'ajouter les contraintes progressivement au problème maître soit de les imposer, si possible, implicitement dans la formulation du primal.

4.4 Conclusion

Nous avons proposé deux approches de stabilisation basées sur la disponibilité d'information sur le polyèdre des solutions optimales duales. Deux types d'inégalités valides sont introduits : inégalités partiellement valides duales-optimales et inégalités valides duales-optimales. Les premières sont des contraintes vérifiées par au moins une solution optimale duale et les dernières sont des contraintes vérifiées par toutes les solutions optimales duales. Ces contraintes traduisent la qualité de l'information

duale disponible. L'ajout de ces contraintes dans la formulation du dual se traduit par l'ajout de colonnes dans la formulation du primal. Nous montrons que dans les deux cas, l'obtention d'une solution optimale duale est garantie alors que la réalisabilité primale peut être compromise. Nous proposons alors une méthode générale simple et efficace pour obtenir une solution primale optimale de base. Cette méthode en deux phases peut être appliquée avec l'un ou l'autre des types d'inégalités valides. Cependant dans le deuxième cas, le fait que les contraintes sont vérifiées par toutes les solutions optimales duales nous permet d'obtenir une paire de solutions optimales primale (de base) en une seule phase : il suffit de modifier légèrement les contraintes utilisées. L'efficacité de la résolution n'est pas affectée vu que les contraintes sont modifiées très légèrement.

Enfin, nous avons discuté quelques aspects liés à l'implantation des approches proposées. Nous illustrerons ces approches sur les problèmes *BCSP* et *CSP* dans la suite de la thèse.

Troisième partie

Applications

Chapitre 5

Stabilisation du problème de tournées de véhicules multi-dépôts

Le problème de tournées de véhicules multi-dépôts (*MDVSP*) [75, 34] est la version de base d'une importante classe de problèmes dans le domaine de tournées et d'horaires. Les version les plus générales utilisent des fenêtres de temps et de ressources afin de modéliser diverses contraintes difficiles. Étant donné que le but de notre étude est la stabilisation de l'évolution des variables duales du problème maître, nous considérons le cas sans ressources ni fenêtres de temps afin d'avoir la version la plus simple des sous-problèmes. Une solution entière de *MDVSP* est généralement trouvée par une procédure de branchement [75, 34]. À chaque nœud de l'arbre de branchement, les contraintes d'intégrité sont relaxées et la relaxation linéaire résultante est résolue par génération de colonnes [75]. Cette résolution devient très difficile et coûteuse lorsque la taille du problème est assez grande. Nous nous proposons d'appliquer l'approche de stabilisation par une pénalité linéaire par morceaux à la relaxation linéaire obtenue au nœud 0 de l'arbre de branchement dans le but de rendre sa résolution plus efficace.

Nous donnons d'abord la formulation de *MDVSP* en problème mutiflats. Puis, nous présentons la formulation compacte en problème de partitionnement. Nous montrons comment obtenir la formulation compacte à partir de la formulation mutiflats en appliquant le principe de décomposition généralisé aux problèmes à variables entières (cf. chapitre 2). Nous présentons ensuite les résultats de la résolution de la re-

laxation linéaire de la formulation compacte par génération de colonnes. Après, nous décrivons en détails l'approche de stabilisation appliquée à *MDVSP*. Nous proposons, en premier lieu, une procédure d'initialisation basée sur une relaxation de *MDVSP* qui fournit une estimation initiale des variables duales, une borne inférieure sur la valeur optimale de la relaxation linéaire, une solution réalisable primale entière et une borne supérieure sur la valeur optimale du problème en nombres entiers. Par la suite, nous testons plusieurs types de stratégie de mise à jour et quelques astuces d'implantation basées sur l'arrêt prématuré des itérations majeures et sur l'élimination des colonnes de stabilisation inutilisées par les solutions optimales des problèmes stabilisés. Nous utilisons également plusieurs directions de déplacement supplémentaire des centres de stabilité dans l'espoir d'accélérer la convergence. Nous présentons enfin les résultats numériques obtenus sur six instances de *MDVSP* et discutons de l'efficacité de l'approche proposée.

5.1 Formulation

Il s'agit de couvrir un ensemble de tâches $\{T_1, T_2, \dots, T_m\}$ par des tournées des véhicules disponibles aux dépôts D_k ($k \in K$), tout en respectant un certain nombre de contraintes opérationnelles. Chaque tâche T_i commence au temps a_i et se termine au temps b_i . Si le temps de déplacement entre i et j est t_{ij} , la paire (T_i, T_j) est compatible si $b_i + t_{ij} \leq a_j$. D'autres contraintes peuvent être imposées dans le graphe de compatibilité. Les nœuds de ce graphe sont les tâches et les dépôts. Les arcs existent entre deux tâches compatibles et entre les tâches et les dépôts. Une tournée réalisable est une suite de tâches (couvertes par un même véhicule) où deux tâches consécutives forment une paire compatible. Elle doit commencer et se terminer au même dépôt.

Une affectation des tâches aux véhicules est dite réalisable si :

- le nombre de véhicules qui partent de chaque dépôt D_k ($k = 1, 2, \dots, |K|$) ne dépasse pas sa capacité n_k ,
- chaque tâche est couverte par une seule tournée de véhicule,
- et chaque véhicule qui part de D_k retourne au même dépôt après avoir effectué sa tournée.

Afin de formuler le problème *MDVSP* comme un problème multi-flots [75, 34], nous introduisons les notations suivantes :

- V : ensemble des sommets du graphe de compatibilité. Les sommets sont les tâches ($i = 1, \dots, n$) et les dépôts ($i = n + 1, \dots, n + |K|$).
- A : ensemble des arcs du graphe de compatibilité. Les arcs sont notés (i, j, k) où k est le numéro du dépôt d'où part la tournée qui utilise cet arc. L'arc peut relier deux tâches ($i = 1, \dots, n; j = 1, \dots, n$), le dépôt D_k à une tâche ($i = n + k; j = 1, \dots, n$) ou une tâche au dépôt D_k ($i = 1, \dots, n; j = n + k$).
- $\delta^+(i)$ ($i = 1, \dots, n$) : ensemble des successeurs de i , c'est-à-dire $\{j : \exists k, (i, j, k) \in A\}$.
- $\delta^-(i)$ ($i = 1, \dots, n$) : ensemble des prédécesseurs de i , c'est-à-dire $\{j : \exists k, (j, i, k) \in A\}$.
- c_{ij} ($i = 1, \dots, n; j = 1, \dots, n$) : coût encouru si la tâche j suit immédiatement la tâche i sur une même tournée. Ce coût est indépendant du dépôt origine du véhicule qui couvre la tournée.
- $c_{n+k,i}$ ($i = 1, \dots, n$) : coût encouru si T_i est la première tâche d'une tournée partant du dépôt D_k .
- $c_{i,n+k}$ ($i = 1, \dots, n$) : coût encouru si T_i est la dernière tâche d'une tournée se terminant au dépôt D_k .

Le problème consiste à trouver une affectation réalisable des tournées aux véhicules (dépôts) qui soit de coût minimum. Le coût d'une tournée est la somme des coûts de tous les arcs qui la composent. Afin d'écrire une formulation en nombres entiers

du problème *MDVSP*, on définit les variables binaires X_{ij}^k ($k = 1, \dots, |K|$; $(i, j) \in \{(s, t) : 1 \leq s, t \leq n\} \cup \{(s, n+k) : 1 \leq s \leq n\} \cup \{(n+k, t) : 1 \leq t \leq n\}$). X_{ij}^k prend la valeur 1 si et seulement si l'arc (i, j, k) est utilisée dans une tournée qui commence et se termine au dépôt D_k . On obtient alors la formulation suivante (Ribeiro et Soumis [75]) :

$$\text{Min} \quad \sum_{(i,j,k) \in A} c_{ij} X_{ij}^k \quad (5.1)$$

sc

$$\sum_{k \in K} \sum_{j \in \delta^+(i)} X_{ij}^k = 1 \quad i = 1, 2, \dots, n \quad (5.2)$$

$$\sum_{j=1}^n X_{n+k,j}^k \leq n_k \quad k \in K \quad (5.3)$$

$$\sum_{j \in \delta^-(i)} X_{ji}^k - \sum_{j \in \delta^+(i)} X_{ij}^k = 0 \quad k \in K, i = 1, 2, \dots, n, n+k \quad (5.4)$$

$$X_{ij}^k \in \{0, 1\} \quad \forall (i, j, k) \in A \quad (5.5)$$

L'objectif (5.1) calcule le coût d'une affectation des tournées aux véhicules en sommant les coûts de toutes les tournées. Les contraintes (5.2) stipulent que chaque tâche doit être couverte exactement une fois et le nombre maximum de véhicules par dépôt est respecté grâce aux contraintes (5.3). Les contraintes de conservation de flot (5.4) assurent la réalisabilité des tournées pour chaque dépôt k . Enfin, les contraintes (5.5) assurent l'intégrité de la solution.

Remarque : Lorsque $|K| = 1$, c'est-à-dire qu'il y a un seul dépôt, le problème est noté *SDVSP* [14]. Ce problème est en fait un problème de flot à coût minimum qui peut être résolu en temps polynomial. Il faut noter que le problème *MDVSP* peut être transformé pour donner un problème de *SDVSP* qui en est une relaxation. Nous reviendrons plus loin sur ce problème quand nous traiterons l'implantation de l'approche de stabilisation pour *MDVSP*.

5.2 Résolution par génération de colonnes

Le nombre de variables et de contraintes dans cette formulation est très élevé et une résolution directe ne serait pas réaliste pour n et $|K|$ assez grands. La formulation couramment utilisée est une formulation en un problème de partitionnement (*Set Partitionning*) [75]. Cette formulation est résolue par une procédure de séparation et dévaluation progressives. À chaque nœud de l'arbre branchement, une relaxation linéaire de ce problème est résolue par génération de colonnes.

Notons par Ω l'ensemble des tournées réalisables. Pour $p \in \Omega$, c_p est le coût de la tournée p , la constante binaire a_{ip} est égale à 1 si et seulement si la tournée p couvre la tâche T_i et la constante binaire b_p^k est égale à 1 si et seulement si la tournée p commence et se termine au dépôt D_k . On peut alors formuler le problème *MDVSP* comme suit :

$$\begin{array}{ll} \text{Min} & \sum_{p \in \Omega} c_p \Theta_p \\ \text{sc} & \end{array} \quad (5.6)$$

$$\sum_{p \in \Omega} a_{ip} \Theta_p = 1 \quad i = 1, 2, \dots, n \quad (5.7)$$

$$\sum_{p \in \Omega} b_p^k \Theta_p \leq n_k \quad k = 1, 2, \dots, |K| \quad (5.8)$$

$$\Theta_p \in \{0, 1\} \quad \forall p \in \Omega. \quad (5.9)$$

Il est facile de voir l'équivalence entre ces deux formulations en nombre entiers. Il est également possible de montrer directement l'équivalence entre leur relaxations linéaires. Nous montrons dans ce qui suit comment la formulation (5.8)-(5.9) s'obtient à partir de la formulation (5.1)-(5.5) en appliquant le principe de décomposition généralisé (*cf.* chapitre 2).

5.2.1 Décomposition en nombres entiers

Afin de décomposer la formulation (5.1)-(5.5), le domaine du sous-problème sera défini par les contraintes de conservation de flot (5.4). Ce domaine possède la propriété d'intégrité (flot à coût minimum). Donc, on peut tenir compte ou non des contraintes d'intégrité (5.5) dans la formulation du sous-problème. La décomposition en nombres entiers du problème *MDVSP* et la décomposition de sa relaxation linéaire fournissent la même relaxation linéaire et la même borne inférieure.

Si on tient compte de ces contraintes dans la définition du sous-problème, le domaine défini par (5.4)-(5.5) sera borné. Les colonnes sont alors les points extrêmes de ce domaine. Ces points extrêmes ne sont pas seulement les chemins réalisables originant et se terminant au même dépôt D_k . Il faut aussi tenir compte des points extrêmes formés par plusieurs chemins distincts. La formulation résultante n'est pas (5.6)-(5.9). Elle doit être transformée en agrégeant les colonnes pour donner cette formulation.

Si on ne tient pas compte des contraintes d'intégrité (5.5), le domaine du sous-problème sera défini seulement par les contraintes (5.4) en plus des contraintes de non-négativité. Ce domaine est non-borné et possède un seul point extrême qui correspond au flot nul. Les rayons extrêmes sont les chemins (réalisables) originant de et se terminant au même dépôt D_k . Les colonnes correspondant à des tournées sont ces rayons extrêmes (chemins).¹ Soient Ω l'ensemble indexant tous les chemins

¹D'un point de vue plus général, on peut différencier les deux choix du sous-problème par le type de contraintes d'intégrité incluses dans la formulation du sous-problème. La formulation qui décrit les colonnes utilise les contraintes X^k binaire ($k = 1, \dots, |K|$) alors que celle qui les décrit par des rayons extrêmes utilise les contraintes d'intégrité X^k entier ($k = 1, \dots, n$). Cette discussion est mieux illustrée dans le cas des problèmes de tournées de véhicules avec contraintes de ressources. Dans ce cas, on doit toujours tenir compte des contraintes d'intégrité dans le sous-problème. La description de son domaine par des points extrêmes ou par des rayons extrêmes dépendra du type des contraintes d'intégrité incluses dans le sous-problème ; c'est-à-dire inclure dans le sous-problème

réalisables x_p et les constantes binaires b_p^k ($k \in K, p \in \Omega$) qui prennent la valeur 1 si le chemin p commence et se termine au dépôt D_k . Alors, toute solution X^k ($k \in K$) est définie par

$$\begin{aligned} X^k &= \sum_{p \in \Omega} b_p^k \Theta_p x_p, \quad k \in K, \\ \Theta_p &\geq 0, \quad p \in \Omega. \end{aligned} \quad (5.10)$$

En substituant dans (5.1)-(5.3), on obtient (5.6)-(5.8) avec

$$\begin{aligned} c_p &= \sum_{(i,j,k) \in A} c_{ij} b_p^k x_{pij}, \quad p \in \Omega \\ a_{ip} &= \sum_{(i,j,k) \in A} b_p^k x_{pij}, \quad p \in \Omega, i = 1, \dots, m. \end{aligned}$$

Nous nous intéressons à la résolution de la relaxation linéaire de (5.6)-(5.9) qui intervient au nœud 0 de la procédure de branchement utilisée pour l'obtention d'une solution entière. Les formulations de cette relaxation (*MP*) (*Mdvsp Primal*) et de son dual (*MD*) (*Mdvsp Dual*) sont les suivantes :

$$\begin{aligned} & (MP) \\ & Min \sum_{p \in \Omega} c_p \Theta_p \\ & sc \left\{ \begin{aligned} & \sum_{p \in \Omega} a_{ip} \Theta_p = 1, \quad i = 1, \dots, m \\ & \sum_{p \in \Omega} b_p^k \Theta_p \leq n_k, \quad k = 1, 2, \dots, |K| \\ & \Theta_p \geq 0, \quad p \in \Omega \end{aligned} \right. \end{aligned}$$

$$\begin{aligned} & (MD) \\ & Max \sum_{i=1}^m b_i \pi_i - \sum_{k \in K} n_k \lambda^k \\ & sc \left\{ \begin{aligned} & \sum_{i=1}^m a_{ip} \pi_i - \sum_{k \in K} b_p^k \lambda^k \leq c_p, \quad p \in \Omega \\ & \lambda \geq 0. \end{aligned} \right. \end{aligned}$$

À partir de la relation (5.10) et sachant que les composantes des x_p ($p \in \Omega$) sont binaires et que chaque tâche doit être couverte exactement une fois, on peut affirmer X^k entier ou X^k binaire ($k = 1, \dots, |K|$).

que si X^k ($k \in K$) est réalisable pour (5.2)-(5.4) et Θ_p ($p \in \Omega$) est réalisable pour (5.7)-(5.8)², alors

$$X^k \text{ binaire}, k = 1, \dots, |K| \Leftrightarrow \Theta_p \text{ binaire}, p \in \Omega.$$

Dépendamment des décisions de branchement prises, la formulation du problème maître peut changer. Cependant, notre approche de stabilisation s'applique à n'importe quel programme linéaire et pourra tenir compte de ces changements dans une étude plus générale.

5.2.2 Expérimentation

Les problèmes utilisés pour les tests sont générés aléatoirement suivant une distribution uniforme [5, 75, 34]. Une fois le nombre de dépôts et le nombre de clients choisis, les coordonnées des clients sont générés dans un carré de côté 60. La procédure de choix des dépôts détermine deux types de problèmes. Pour les problèmes de type A, les coordonnées des dépôts sont générées aléatoirement. Pour les problèmes de type B, les 4 premiers dépôts sont placés sur les sommets du carré et les coordonnées des autres sont générées aléatoirement à l'intérieur du carré. Nous considérons 6 problèmes désignés par $pmTd.i$ où m est le nombre de clients ($m \in \{400, 800, 1000\}$), T est le type des problèmes ($T \in \{A, B\}$), d est le nombre de dépôts ($d \in \{4, 5\}$) et i est le numéro de l'instance générée. Le tableau 5.1 présente les instances utilisées et leur caractéristiques. Le nombre de contraintes par problème est $m + d$ dont m contraintes de couverture. Le nombre de sous-problèmes est d vu qu'une tournée valide doit débuter et se terminer au même dépôt. La colonne arcs donne le nombre d'arcs dans le réseau d'un sous-problème.

²Ce résultat est aussi vrai pour tout problème de partitionnement à condition que les composantes des colonnes soient binaires.

Tableau 5.1 – *MDVSP* : caractéristiques des instances résolues.

Problème	T	m	d	arcs
<i>p400A4_0</i>	A	400	4	206696
<i>p400B4_0</i>	B	400	4	210328
<i>p800A4_0</i>	A	800	4	785880
<i>p800B4_0</i>	B	800	4	816780
<i>p1000A5_0</i>	A	1000	5	1273435
<i>p1000B5_0</i>	B	1000	5	973260

Le tableau 5.2 donne les résultats obtenus en résolvant les relaxations linéaires des problèmes du tableau 5.1 par génération de colonnes non-stabilisée. La résolution est faite par l'optimiseur GENCOL(voir [13]). Les problèmes de plus court chemin sont résolus par un algorithme de programmation dynamique [14]. Les problèmes maîtres restreints sont résolus par l'optimiseur Cplex6.6 [7]. Plusieurs colonnes de coût réduit négatif sont ajoutées au problème maître à chaque itération de génération de colonnes. Les tests sont effectués sur une machine SUN Ultra-10/440 (int95=18.1, fp95=22.7, 640M).

Les résultats sont présentés dans le tableau (5.2). $lr(s)$, $mp(s)$ et $sp(s)$ donnent les temps de calcul nécessaires pour la résolution de la relaxation linéaire, du problème maître et du sous-problème, respectivement, pour chacun des problèmes considérés. itr est le nombre d'itérations de génération de colonnes nécessaires pour l'obtention d'une solution optimale de la relaxation linéaire.

Les résultats montrent que les problèmes ayant un grand nombre de tâches sont difficiles à résoudre. Le nombre d'itérations de générations de colonnes et le temps de calcul total sont plus grands. Le temps de résolution du sous-problème est quasiment le même à chaque itération. Ceci s'explique par l'absence de ressource et que l'algorithme est alors polynômial. Le temps de résolution des problèmes maîtres restreints

Tableau 5.2 – *MDVSP* : résolution par génération de colonnes.

Problème	lr(s)	mp(s)	sp(s)	itr
<i>p400A4.0</i>	203.5	125.9	77.6	149
<i>p400B4.0</i>	285.2	180.5	104.7	196
<i>p800A4.0</i>	4178.4	3149.2	1029.2	509
<i>p800B4.0</i>	3561.9	2676.2	885.7	422
<i>p1000A5.0</i>	2883.4	1641.3	1242.1	380
<i>p1000B5.0</i>	1428.9	779.7	649.2	259

est plus élevé que le temps de résolution des sous-problèmes. Il prend plus de 60% du temps total même pour les petits problèmes ($n = 400$). Le nombre d'itérations de génération de colonnes est très élevé pour tous les types de problèmes ce qui laisse présager un apport important de la stabilisation. Les problèmes ayant 800 tâches semblent être plus difficiles que ceux ayant 1000 tâches. Ceci n'est probablement pas vrai pour toutes les instances. Les instances ayant 800 tâches que nous considérons semblent être particulièrement difficiles pour la résolution par génération de colonnes.

5.3 Stabilisation

L'approche de stabilisation sera appliquée à la relaxation linéaire de la formulation compacte (5.6)-(5.9) qui est résolue par génération de colonnes. Dans cette approche, la pénalisation est appliquée seulement aux variables duales correspondant aux contraintes de couverture. Deux raisons ont motivé ce choix : d'une part, la difficulté de la résolution de ces problèmes est due au nombre élevé de contraintes de couverture étant donné que le nombre de dépôt est généralement petit et varie très peu ; d'autre part, le nombre de véhicules étant largement suffisant pour couvrir toutes les tâches, les multiplicateurs de ces contraintes sont nuls à l'optimalité et s'ajustent facilement une fois que les autres multiplicateurs sont stabilisés. L'absence

des termes de stabilisation pour une contrainte se traduit dans la fonction de pénalité par des bornes infinie et des pénalités nulles. Les formulations qu'on obtient pour le problème primal stabilisé (*SMP*) et son dual (*SMD*) à une itération majeure l sont :

$$\begin{aligned}
 & (SMP)_l \\
 & \text{Min } \sum_{p \in \Omega} c_p \Theta_p + \sum_{i=1}^m (-\gamma_{-,i}^l z_i^- - \delta_{-,i}^l y_i^- + \delta_{+,i}^l y_i^+ + \gamma_{+,i}^l z_i^+) \\
 & \text{sc } \left\{ \begin{array}{l} \sum_{p \in \Omega} a_{ip} \Theta_p - z_i^- - y_i^- + y_i^+ + z_i^+ = 1, \quad i = 1, \dots, m \\ \sum_{p \in \Omega} b_p^k \Theta_p \leq n_k, \quad k = 1, 2, \dots, |K| \\ z^- \leq \zeta_-, y^- \leq \varepsilon_- \\ y^+ \leq \varepsilon_+, z^+ \leq \zeta_+ \\ \Theta, z^-, y^-, y^+, z^+ \geq 0 \end{array} \right.
 \end{aligned}$$

$$\begin{aligned}
 & (SMD)_l \\
 & \text{Max } \sum_{i=1}^m (b_i \pi_i - \zeta_{-,i} v_i^- - \varepsilon_{-,i} u_i^- - \varepsilon_{+,i} u_i^+ - \zeta_{+,i} v_i^+) - \sum_{k \in K} n_k \lambda^k \\
 & \text{sc } \left\{ \begin{array}{l} \sum_{i=1}^m a_{ip} \pi_i - \sum_{k \in K} b_p^k \lambda^k \leq c_p, \quad p \in \Omega \\ \delta_{-,i} - u_i^- \leq \pi_i \leq \delta_{+,i} + u_i^+, \quad i = 1, \dots, m \\ \gamma_{-,i} - v_i^- \leq \pi_i \leq \gamma_{+,i} + v_i^+, \quad i = 1, \dots, m \\ \lambda, v^-, u^-, u^+, v^+ \geq 0. \end{array} \right.
 \end{aligned}$$

Les problèmes (*MP*) et (*MD*) seront résolus à l'aide de l'algorithme donné au chapitre 3. Chaque itération majeure consiste à résoudre par génération de colonnes la paire de problèmes stabilisés $((SMP)_l, (SMD)_l)$.

Notons d'abord qu'une particularité importante de *MDVSP* est que tous les b_i ($i = 1, \dots, m$) sont égaux à 1. En fait, c'est une caractéristique de tous les problèmes de partitionnement. Ce qui veut dire que les résultats que nous obtiendrons ici donneront une bonne idée sur ce qu'on obtiendrait sur un problème semblable.

La fonction de pénalité utilisée g^l vérifie que pour au moins un certain $i \in \{1, \dots, m\}$ ³ :

$$\begin{aligned} \varepsilon_{-,i} + \zeta_{-,i} &> 1 \\ \varepsilon_{+,i} + \zeta_{+,i} &> 1. \end{aligned} \tag{5.11}$$

Dans le cas contraire, la pénalité n'aura pas d'effet de restriction. L'objectif sera toujours croissant dans la direction de

$$b = \begin{bmatrix} 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix}.$$

Les intervalles de confiance vérifient

$$\begin{aligned} \delta_{+,i} - \hat{\pi}_i &\geq 10, \\ \hat{\pi}_i - \delta_{-,i} &\geq 10. \end{aligned}$$

Ceci assurera la convergence de l'algorithme vers une paire de solutions optimales primale-duale de $((MP), (MD))$.

Nous nous penchons maintenant sur les caractéristiques de l'implantation de la stabilisation.

5.3.1 Initialisation

Le choix du centre de stabilité initial $\hat{\pi}^0$ peut avoir un impact significatif sur l'efficacité de la résolution. En effet, plus $\hat{\pi}^0$ est proche de l'espace optimal dual, plus la convergence se fera rapidement. De plus, une solution primale initiale permet d'enrichir le problème maître par des colonnes réalisables et de fournir une borne

³La même condition devrait être utilisée dans le cas d'un problème de partitionnement et peut être généralisé au cas où les membres de droite ne sont pas forcément égaux à 1. Il suffit de remplacer 1 par b_i .

supérieure sur la valeur optimale. Nous discutons de ces deux aspects dans ce qui suit.

Choix de $\hat{\pi}^0$: Un choix trivial de $\hat{\pi}^0$ est le vecteur nul. Ce choix ne fournit pas d'information particulière au problème et suppose implicitement la symétrie entre les variables duales. Il est toutefois réalisable pour (MD) .

Nous proposons dans ce qui suit une estimation de $\hat{\pi}^0$ à partir du problème *SDVSP*.

Supposons que les dépôts k ($k = 1, \dots, |K|$) sont agrégés en un seul dépôt D_0 auquel on associera le nœud 0. Ce dépôt a une capacité suffisante pour couvrir toutes les tâches. On définit le coût de l'arc $(0, i)$ ($i = 1, \dots, m$) comme étant le minimum des coûts $c_{n+k, i}$ ($k = 1, \dots, |K|$). De même, on définit le coût de l'arc $(0, i)$ par $c_{0, i} = \min \{c_{i, n+k}, k = 1, \dots, |K|\}$. On obtient alors un problème de tournées de véhicules à un seul dépôt qu'on note *SDVSP*. Ce problème, qui est une relaxation de *MDVSP*, est en fait un problème de flot à coût minimum sur le graphe modifié tel qu'expliqué ci-dessus (l'ensemble V devient $\{0, 1, \dots, m\}$ et l'ensemble A est formé par des arcs du type (i, j) ($0 \leq i, j \leq m$)). En modifiant la formulation de *MDVSP*, on obtient la formulation suivante pour *SDVSP* :

$$\text{Min} \quad \sum_{(i,j) \in A} c_{ij} X_{ij} \quad (5.12)$$

sc

$$\sum_{j \in \delta^+(i)} X_{ij} = 1 \quad i = 1, 2, \dots, n \quad \text{---} > \tilde{\pi} \quad (5.13)$$

$$\sum_{j \in \delta^-(i)} X_{ji} - \sum_{j \in \delta^+(i)} X_{ij} = 0 \quad i = 0, 1, \dots, n \quad \text{---} > \tilde{\alpha} \quad (5.14)$$

$$X_{ij} \in \{0, 1\} \quad \forall (i, j) \in A \quad (5.15)$$

Le dual de ce problème (duquel on enlève les contraintes d'intégrité (5.15)) est une

restriction du dual de la relaxation linéaire de $MDVSP$. La résolution de $SDVSP$ produit donc un vecteur de multiplicateurs réalisable pour (MD) . Par conséquent, un multiplicateur optimal de $SDVSP$ fournit une estimation pour une solution optimale de (MD) et une borne inférieure sur sa valeur optimale.

Notons par $\tilde{\pi}$ le vecteur des multiplicateurs optimaux associés à l'ensemble des contraintes (5.13). On peut alors énoncer le résultat suivant :

Proposition 5.1 $\tilde{\pi}$ vérifie les deux propriétés suivantes :

- i) $\tilde{\pi}$ est réalisable pour (MD)
- ii) $b^T \tilde{\pi}$ est une borne inférieure sur la valeur optimale de (MP) . ■

Une formulation alternative consisterait à remplacer les contraintes (5.13)-(5.14) par

$$\sum_{j \in \delta^+(i)} X_{ij} = 1 \quad i = 1, 2, \dots, n \quad \text{---} > \tilde{\pi} \quad (5.16)$$

$$\sum_{j \in \delta^-(i)} X_{ji} = 1, \quad i = 1, 2, \dots, n \quad \text{---} > \tilde{\sigma} \quad (5.17)$$

$$\sum_{i=1}^m X_{0i} - \sum_{i=1}^m X_{i0} = 0 \quad \text{---} > \bar{\alpha} \quad (5.18)$$

Si on note par $\tilde{\pi}$ et $\tilde{\sigma}$ les vecteurs de multiplicateurs optimaux associés aux contraintes (5.16) et (5.17) respectivement. La proposition suivante donne une estimation pour des multiplicateurs optimaux de (MP) .

Proposition 5.2 $\tilde{\pi}$ et $\tilde{\sigma}$ vérifient :

- i) $\tilde{\pi} + \tilde{\sigma}$ est réalisable pour (MD)

ii) $b^T(\bar{\pi} + \bar{\sigma})$ est une borne inférieure sur la valeur optimale de (MP) .■

En résolvant les deux problèmes par la procédure *netopt* de Cplex6.6 [7], nous avons remarqué que la deuxième formulation se résout plus rapidement que la première. Cependant, les temps de résolution restent très petits dans les deux cas.

Solution primale initiale : La solution primale de *SDVSP* peut être transformée en une solution primale réalisable pour *MDVSP*. Il suffit de construire, pour chaque colonne de la solution de *SDVSP*, $|K|$ colonnes partant chacune d'un dépôt tout en recalculant correctement le coût. En choisissant le meilleur dépôt pour chaque colonne tout en tenant compte des capacités des dépôts,⁴ on obtient une solution réalisable entière ainsi qu'une borne supérieure pour *MDVSP*.

5.3.2 Calcul d'une borne inférieure après une itération mineure

Si le coût d'une tournée comporte un coût fixe suffisamment grand, le nombre de colonnes à l'optimalité de (MP) est minimum [34]. De plus, toutes les tournées unitaires sont réalisables. On peut donc affirmer qu'à la fin d'une itération mineure,

$$\sum_j x_j + \sum_i (z_i^+ + y_i^+ - y_i^- - z_i^-)$$

⁴La solution fournie dépend de l'ordre dans lequel les colonnes sont traitées. Un critère *glouton* est nécessaire pour le choix d'un dépôt pour une colonne.

est une borne supérieure sur le nombre de véhicules à l'optimalité de (MP) . Par conséquent, si \bar{c}_{min} est le coût réduit minimum calculé à fin de cette itération mineure,

$$v(SMD_{l,i}) + \bar{c}_{min}(\sum_j x_j + \sum_i (z_i^+ + y_i^+ - y_i^- - z_i^-))$$

est une borne inférieure sur la valeur optimale de (MP) .

Remarque : En pratique, il n'est pas très utile de calculer la borne inférieure après chaque itération mineure. Pour les tests, la borne est calculée toutes les dix itérations. De plus, cette borne permet très rarement d'arrêter une itération majeure.

5.3.3 Choix de la fonction de pénalité

Le choix des paramètres de la fonction de pénalité est déterminant pour l'efficacité de l'algorithme. D'une part il faut que la résolution des problèmes stabilisés soit assez facile, et d'autre part le nombre d'itérations majeures doit être assez petit de façon à ce que la résolution des problèmes originaux par l'approche de stabilisation soit plus efficace que la résolution par génération de colonnes non-stabilisée. Nous discutons dans ce qui suit des choix des intervalles de confiance (γ_{\pm} et δ_{\pm}), des pénalités ε_{\pm} et ζ_{\pm} , du prochain centre de stabilité ($\hat{\pi}^{l+1}$) et des stratégies de mise à jour des paramètres.

Largeurs des intervalles : L'utilisation d'intervalles trop petits résulterait en un très grand nombre d'itérations majeures. Ces itérations permettraient des déplacements minimes du centre de stabilité. La convergence serait alors très lente et la résolution serait moins efficace que la résolution par génération de colonnes. Par contre, si l'intervalle de confiance est trop grand, une seule itération majeure sera

nécessaire et le problème est résolu au complet. La stabilisation est alors équivalente à la résolution par génération de colonnes. Un choix raisonnable qui tient compte des valeurs optimales des variables duales optimales s'impose. Dans notre cas, le choix des largeurs des intervalles dépend des estimations initiales des variables duales ainsi que de la taille des problèmes (i.e. le nombre de tâches m). Nous avons testé plusieurs intervalles possibles qui ont donné des résultats assez réguliers. Pour l'intervalle $[\delta_-, \delta_+]$, la largeur minimale est 10 et la largeur maximale est 1000. Pour l'intervalle $[\gamma_-, \gamma_+]$, la largeur minimale est 50 et la largeur maximale est 10000.

De plus, dépendamment du type des contraintes stabilisées et du signe du second membre associé, on peut utiliser un seul ou deux côtés de la fonction de pénalité. L'élimination du côté gauche, par exemple, revient soit à imposer des bornes infinies sur les variables duales soit à utiliser des pénalités nulles.

Pénalités : Des pénalités trop élevées limitent le déplacement du centre de stabilité à l'extérieur de la boîte $[\delta_-, \delta_+]$ alors que des pénalités faibles facilitent ce déplacement. Par exemple, si ε_- et ε_+ sont faibles, la fonction de pénalité permet d'élargir l'intervalle de confiance mais avec une légère perturbation. Le choix des pénalités dépend des largeurs des intervalles et de l'étape où l'optimisation est rendue. Pour les tests, nous avons utilisé les valeurs 0.3, 0.5 et 0.8 pour ε_{\pm} et la valeur 1 pour ζ_{\pm} . Les résultats présentent une variance très faible.

Mise à jour du centre de stabilité : La version standard de l'approche de stabilisation proposée consiste à centrer la fonction de pénalité g^l autour de la solution du problème dual stabilisé. Cependant tel que nous l'avons montré dans le chapitre 3, il est possible de déplacer ce centre dans n'importe quelle direction sans compromettre la convergence de l'algorithme grâce à un choix adéquat des pas de déplacement.

Nous utilisons cinq directions dans le but de voir leur impact sur l'efficacité de la résolution.

Stratégies de mise à jour : La stratégie la plus simple à implanter consiste à garder une fonction de pénalité inchangée autour du centre de stabilité courant. Ainsi, le problème maître subit peu de changement ; seuls les coûts des variables de stabilisation sont modifiés. Le comportement de ce genre de stratégie dépend fortement des largeurs des intervalles et des pénalités choisies.

Nous testons aussi une stratégie qui consiste à modifier les paramètres de la fonction de pénalité en fonction de la position de la solution du problème stabilisé par rapport aux intervalles $[\delta_-, \delta_+]$ et $[\gamma_-, \gamma_+]$. Ceci résulte en des changements importants dans le problème maître qui pourrait devenir coûteux à réoptimiser.

Nous testons également une stratégie qui consiste en des modifications limitées des paramètres de stabilisation et qui garde la fonction de pénalité inchangée à la fin du processus.

5.3.4 Quelques astuces d'implantation

Nous décrivons quelques astuces d'implantation que nous avons utilisées pour nos tests.

Arrêt prématuré des premières itérations majeures : L'obtention d'une solution optimale des problèmes stabilisés au début de la résolution n'est pas indispensable pour la convergence de l'algorithme. Sachant que ceci peut être particulièrement

coûteux, il serait utile d'éviter de gaspiller du temps de calcul à chercher l'optimalité des problèmes stabilisés pendant les premières itérations majeures.

Limitation du nombre d'itérations mineures d'une itération majeure :

Les tests effectués ont montré que le nombre d'itérations mineures à la fin du processus est très petit. On peut donc imposer une limite sur le nombre d'itérations mineures pour chaque itération majeure sans compromettre la convergence de l'algorithme en pratique. Cette idée est en fait une généralisation de la précédente.

Élimination des colonnes de stabilisation à la fin d'une itération majeure :

Les tests nous ont permis d'observer que le nombre de variables de stabilisation utilisées dans la solution du problème stabilisé décroît avec le nombre d'itérations majeures. L'élimination des variables de stabilisation inutilisées (de valeur nulle et hors-base) permettrait d'alléger le problème maître dont la résolution serait plus efficace. Nous avons aussi remarqué qu'une fois la solution du problème primal stabilisé est réalisable pour (P) , aucune variable de stabilisation n'est utilisée par la suite. Ces variables peuvent donc être éliminées une fois que la réalisabilité primale (de (P)) est atteinte.

5.4 Résultats numériques

Nous présentons dans cette section les résultats obtenus en appliquant l'approche de stabilisation présentée au chapitre 3 et expliquée plus haut à la relaxation linéaire de *MDVSP*. Nous commençons par tester trois stratégies de mise à jour. Puis, nous présentons les résultats de l'utilisation de quelques astuces d'implantation. Nous nous pencherons ensuite sur l'effet de l'utilisation de certaines directions particulières pour

un déplacement supplémentaire du centre de stabilité. Enfin, nous présentons les résultats obtenus en utilisant simultanément plusieurs techniques d'accélération.

5.4.1 Stratégies de mise à jour

La première stratégie, dite statique, consiste à conserver la fonction de pénalité inchangée par rapport au centre de stabilisation. La deuxième, dite dynamique, consiste à modifier les paramètres de la fonction de pénalité en fonction de la position de π^l par rapport aux intervalles de confiance. Enfin, la dernière stratégie, dite hybride, consiste à aplatiser la fonction entre δ_{\pm} et γ_{\pm} à chaque itération tout en gardant ε_{\pm} plus grand que 10^{-4} . Les intervalles $[\delta_{\pm}, \gamma_{\pm}]$ sont élargis si π^l est à l'extérieur de $[\gamma_-, \gamma_+]$. La largeur de cet intervalle ne doit pas dépasser 2×10^4 .

Le tableau 5.3 contient les résultats de l'utilisation de ces différentes stratégies. Nous y reportons les résultats de la résolution par génération de colonnes sous les colonnes STD. STAT désigne la stratégie de mise à jour statique, DYN désigne la stratégie de mise à jour dynamique et HYB désigne la stratégie de mise à jour hybride. $lr(s)$, $mp(s)$ et $sp(s)$ sont les temps de calcul de la relaxation linéaire, du problème maître et du sous-problème, respectivement. itr est le nombre total d'itérations de génération de colonnes (itérations mineures) et l est le nombre d'itérations majeures.

Les trois stratégies permettent d'améliorer les performances de la génération de colonnes d'une manière significative. Le nombre d'itérations de génération de colonnes et le temps de résolution du problème maître présentent les améliorations les plus nettes. Le temps de résolution du sous-problème est quasiment proportionnel au nombre d'itérations pour les mêmes raisons évoquées dans la section précédente. L'apport de la stabilisation est plus grand sur les problèmes ayant 800 tâches. Ces

Tableau 5.3 – *MDVSP* : stabilisation par une fonction de pénalité linéaire par morceaux avec 3 stratégies de mise à jour différentes.

Pb	<i>p400A4.0</i>				<i>p400B4.0</i>			
Stratégie	STD	STAT	DYN	HYB	STD	STAT	DYN	HYB
lr(s)	203.5	43.6	40.5	42.2	285.2	46.4	49.5	44.9
mp(s)	125.9	7.8	7.3	13.7	180.5	10.6	12.4	20.0
sp(s)	77.6	35.8	33.2	28.5	104.7	35.8	37.1	24.9
itr	149	74	69	52	196	72	75	53
<i>l</i>	-	11	10	3	-	13	13	4

Pb	<i>p800B4.0</i>				<i>p800A4.0</i>			
Stratégie	STD	STAT	DYN	HYB	STD	STAT	DYN	HYB
lr(s)	4178.4	670.0	951.4	595.5	3561.9	361.0	399.4	306.4
mp(s)	3149.2	203.2	187.4	216.2	2676.2	165.0	179.8	141.9
sp(s)	1029.2	466.8	764.0	379.3	885.7	196.0	219.6	164.5
itr	509	223	359	196	422	90	102	80
<i>l</i>	-	30	95	29	-	3	6	5

Pb	<i>p1000A5.0</i>				<i>p1000B5.0</i>			
Stratégie	STD	STAT	DYN	HYB	STD	STAT	DYN	HYB
lr(s)	2883.4	1213.2	1529.4	1223.5	1428.9	969.1	1107.7	837.1
mp(s)	1641.3	639.2	872.1	610.9	779.4	388.8	585.3	479.6
sp(s)	1242.1	574.0	657.3	612.6	649.2	580.3	522.4	357.5
itr	380	178	203	178	259	236	212	145
<i>l</i>	-	10	6	12	-	25	23	8

problèmes sont les plus difficiles et présentent probablement plus d'instabilité que les autres. Enfin, la stratégie hybride semble être la plus robuste et offre le meilleur comportement pour l'algorithme. Pour les problèmes de 400 tâches, le temps de calcul est divisé par un facteur de 4.8 pour le problème de type *A* et un facteur de 6.4 pour le problème de type *B*. Pour les problèmes de 800 tâches, ces facteurs sont 7.0 et 11.6. Enfin, pour les problèmes de 1000 tâches, ces facteurs sont 2.4 et 1.7. Le facteur de réduction du nombre d'itérations de génération de colonnes pour les problèmes à 400 tâches est de 2.9 pour le problème de type *A* et de 3.7 pour le problème de type *B*. Ces facteurs sont 2.6 et 5.3 pour les problèmes à 800 tâches et 2.1 et 1.8 pour les problèmes à 1000 tâches.

5.4.2 Effet de l'utilisation des astuces

Nous avons testé l'effet de l'utilisation de quelques astuces sur le comportement de la stabilisation. Les deux premières astuces consistent à fixer un nombre maximum d'itérations mineures à l'intérieur d'une itération majeure. Pour la première astuce, notée Stop1, ce nombre maximum est imposé seulement aux deux premières itérations majeures alors que pour la deuxième astuce, notée Stop2, ce nombre maximum est imposé à toutes les itérations majeures. La troisième astuce, notée Elim, consiste à éliminer les colonnes de stabilisation inutilisées par une solution optimale de (SP_t) . Les résultats sont présentés dans le tableau 5.4. Ces astuces sont utilisées avec la stratégie de mise à jour hybride qui s'est avérée la plus robuste (cf. tableau 5.3). Les notations utilisées sont les mêmes que les tableaux précédents.

Les résultats montrent que l'arrêt prématuré d'une itération mineure peut causer une augmentation du temps de calcul total par rapport à la stratégie de mise à jour hybride. Cependant dans les cas où ceci est vrai (problèmes à 800 tâches), le nombre d'itérations de génération de colonnes diminue. Le fait que l'arrêt prématuré

ne garantit pas une direction de montée pour le dual est probablement la cause de la détérioration possible du temps de calcul. De plus, la fonction peut ne pas être bien définie autour de ce point et alors l'itération majeure arrêtée est alors inutile.

Tableau 5.4 – *MDVSP* : utilisation des astuces dans l'implantation de la stabilisation par une fonction de pénalité linéaire par morceaux.

Pb	<i>p400A4_0</i>					<i>p400B4_0</i>				
Astuce	STD	HYB	Stop1	Stop2	Elim	STD	HYB	Stop1	Stop2	Elim
lr(s)	203.5	42.5	33.9	35.5	37.0	285.2	44.9	43.1	43.5	48.8
mp(s)	125.9	13.7	12.3	12.3	12.1	180.5	20.0	20.7	19.9	18.1
sp(s)	77.6	28.5	21.6	23.2	24.9	104.7	24.9	22.4	23.6	30.7
itr	149	52	44	44	48	196	53	45	45	58
<i>l</i>	-	3	3	3	3	-	4	3	3	3

Pb	<i>p800B4_0</i>					<i>p800A4_0</i>				
Astuce	STD	HYB	Stop1	Stop2	Elim	STD	HYB	Stop1	Stop2	Elim
lr(s)	4178.4	595.5	968.6	730.7	501.2	3561.9	306.4	353.1	335.1	256.0
mp(s)	3149.2	216.2	575.1	375.9	166.2	2676.2	141.9	210.5	201.1	113.9
sp(s)	1029.2	379.3	393.5	354.8	335.0	885.7	164.5	142.6	134.0	142.1
itr	509	196	197	166	160	422	80	68	68	72
<i>l</i>	-	29	17	17	9	-	5	4	4	3

Pb	<i>p1000A5_0</i>					<i>p1000B5_0</i>				
Astuce	STD	HYB	Stop1	Stop2	Elim	STD	HYB	Stop1	Stop2	Elim
lr(s)	2883.4	1223.5	1129.7	1155.6	1068.3	1428.9	837.1	815.4	837.7	757.2
mp(s)	1641.3	610.9	663.2	656.4	593.4	779.4	479.6	358.3	349.5	272.0
sp(s)	1242.1	612.6	466.5	499.2	474.9	649.2	357.5	457.1	488.2	485.2
itr	380	178	152	152	145	259	145	194	194	194
<i>l</i>	-	12	13	13	9	-	8	23	23	22

Contrairement aux deux premières, la troisième astuce s'est avérée bénéfique pour tous les problèmes. Généralement, le temps de calcul est amélioré et le nombre d'itérations de génération de colonnes (itérations mineures) reste raisonnable. Pour les problèmes ayant plus que 1000 tâches, le nombre d'itérations majeures augmente. Cependant, ces dernières deviennent plus faciles et le temps de calcul total n'est pas

détérioré.

5.4.3 Enrichissement des directions de déplacement

Nous testons cinq directions de déplacement supplémentaires dans le but d'obtenir un "meilleur" centre de stabilité à l'itération majeure suivante. L'implantation de cette modification obéit aux conditions du théorème 2.4. Ceci permettra entre autres de réduire le nombre d'itérations majeures requis pour atteindre l'optimalité de (MP) . Les cinq directions utilisées sont décrites dans ce qui suit :

- $d^1 = b$: Au lieu de prendre π^l comme prochain centre de stabilité, on fait un déplacement supplémentaire dans la direction du gradient de la fonction objectif en espérant que l'optimisation de (SP_l) et (SD_l) fournira une meilleure borne inférieure.
- $d^2 = \pi^l - \hat{\pi}^l$: Sachant que $\pi^* - \hat{\pi}^l$ est une direction de montée pour le dual, on effectue un déplacement supplémentaire dans cette même direction. Ceci revient à utiliser un pas de déplacement $1 + \alpha_l$ dans cette direction à partir de $\hat{\pi}^l$.
- $d^3 = (\pi^l - \hat{\pi}^l)^+$: Parmi les composantes de la direction précédente, on garde seulement celles ayant le même signe que b_i . Dans notre cas $b_i = 1 (i = 1, \dots, m)$. Donc, on se déplace seulement suivant les composantes positives de d^2 .
- $d^4 = b - Ax^l$: Sachant que $b - Ax^l$ est un sous-gradient de la fonction duale au point π^l , cette direction est une direction de descente pour la fonction qui calcule la distance entre un multiplicateur optimal et un vecteur de multiplicateur quelconque (cf. chapitre 2).
- $d^5 = (b - Ax^l)^+$: Cette direction est calculée à partir de la direction d^4 de la même manière que la direction d^3 est calculée à partir de la direction d^2 .

Les résultats sont donnés dans le tableau 5.5. Les notations sont les mêmes que celles utilisées dans les tableaux précédents. Les performances de la stabilisation avec déplacement supplémentaire pour les différentes directions d^p ($p = 1 \dots, 5$) sont comparées à la performance de la stabilisation utilisant une stratégie de mise à jour hybride ; cette dernière étant la stratégie ayant produit les meilleurs résultats du tableau 5.3. Les résultats montrent l'effet bénéfique de l'enrichissement de la direction de déplacement par les directions proposées ci-dessus. Pour le problème $p1000A5.0$, cet effet n'est pas très sensible. Ceci est dû probablement à la difficulté de résolution du problème maître après chaque déplacement.

Tableau 5.5 – *MDVSP* : déplacement supplémentaire des centres de stabilité.

Pb	<i>p400A4.0</i>						<i>p400B4.0</i>					
d	HYB	d^1	d^2	d^3	d^4	d^5	HYB	d^1	d^2	d^3	d^4	d^5
lr(s)	42.2	36.0	35.7	37.1	37.4	40.0	44.9	45.4	45.5	46.0	48.5	48.2
mp(s)	13.7	11.9	12.6	12.5	12.1	14.2	20.0	17.6	19.4	18.0	17.7	18.1
sp(s)	28.5	24.1	23.1	24.6	25.3	25.8	24.9	27.8	26.1	28.0	30.8	30.1
itr	52	47	47	50	49	49	53	56	53	56	50	57
l	3	3	3	3	3	3	4	3	3	3	3	3

Pb	<i>p800B4.0</i>						<i>p800A4.0</i>					
d	HYB	d^1	d^2	d^3	d^4	d^5	HYB	d^1	d^2	d^3	d^4	d^5
lr(s)	595.5	465.3	458.7	452.5	480.2	477.5	306.4	268.4	260.4	253.2	251.4	252.9
mp(s)	216.2	148.9	157.2	167.4	153.9	157.1	141.9	111.2	114.7	111.0	114.9	114.5
sp(s)	379.3	316.4	301.5	285.1	326.3	320.4	164.5	157.2	145.7	142.2	136.5	138.4
itr	196	158	157	147	156	152	80	73	71	69	69	70
l	29	13	9	9	13	12	5	3	3	3	3	3

Pb	<i>p1000A5.0</i>						<i>p1000B5.0</i>					
d	HYB	d^1	d^2	d^3	d^4	d^5	HYB	d^1	d^2	d^3	d^4	d^5
lr(s)	1223.5	1110.3	1164.1	1058.9	1139.2	1111.5	837.1	726.2	740.5	730.7	679.3	740.5
mp(s)	610.9	619.5	634.0	616.7	598.4	633.0	479.6	254.8	282.6	272.0	331.6	270.8
sp(s)	612.6	490.8	530.1	442.2	540.8	478.5	357.5	471.4	457.9	458.7	347.7	469.7
itr	178	195	171	143	165	146	145	191	192	190	133	189
l	12	22	12	9	11	9	8	22	22	22	11	22

5.4.4 Utilisation simultanée de plusieurs techniques d'accélération

Le but de ces derniers tests est d'illustrer l'effet de la stabilisation sur l'efficacité de la résolution de *MDVSP* par génération de colonnes. Nous présentons maintenant les résultats obtenus en utilisant simultanément plusieurs techniques d'accélération parmi celles présentées.

- Stratégie : Nous utilisons une stratégie hybride vue qu'elle a le comportement le plus stable.
- Astuces : Nous utilisons seulement la troisième astuce qui consiste à éliminer les colonnes de stabilisation inutilisées à l'optimalité d'un problème stabilisé. Les autres astuces n'apportant pas d'amélioration des performances de la stabilisation ne sont pas utilisées.
- Choix de la direction : Nous combinons deux directions parmi celles utilisées plus haut. Si l'itération majeure courante est arrêtée à l'optimalité du problème stabilisé courant, la direction utilisée est d^3 . Par contre si l'arrêt est dû à la borne inférieure, le déplacement se fait dans la direction du gradient de l'objectif dual b .

Les résultats sont présentés au tableau 5.6. Les colonnes de ce tableau sont les mêmes que pour les tableaux précédents. Les lignes portant les noms des problèmes donnent les résultats de la résolution de ces problèmes par génération de colonnes. La ligne Stab donne les résultats de la résolution par l'implantation de stabilisation décrite ci-dessus et la ligne Rapport donne les facteurs de réduction dû à la stabilisation.

L'apport de la stabilisation est impressionnant. Les facteurs de réduction pour le temps de calcul total, le nombre d'itérations de génération de colonnes et le temps du problème maître illustrent une plus grande stabilité du problème. Notons que l'utili-

Tableau 5.6 – *MDVSP* : utilisation simultanée de plusieurs techniques d'accélération dans l'implantation de la stabilisation.

Problème	lr(s)	mp(s)	sp(s)	itr	<i>l</i>
<i>p400A4.0</i>	203.5	125.9	77.6	149	-
Stab	32.7	9.4	23.3	43	3
Rapport	6.2	13.4	3.3	3.5	-
<i>p400B4.0</i>	285.2	180.5	104.7	196	-
Stab	38.6	17.0	21.6	46	2
Rapport	7.4	10.6	4.8	4.3	-
<i>p800A4.0</i>	4178.4	3149.2	1029.2	509	-
Stab	490.0	218.8	271.2	135	5
Rapport	8.5	14.4	3.8	3.8	-
<i>p800B4.0</i>	3561.9	2676.2	885.7	422	-
Stab	240.9	110.3	130.6	63	2
Rapport	14.8	24.3	6.8	6.7	-
<i>p1000A5.0</i>	2883.4	1641.3	1242.1	380	-
Stab	996.7	630.3	366.4	107	13
Rapport	2.9	2.6	3.4	3.6	-
<i>p1000B5.0</i>	1428.9	779.4	649.2	259	-
Stab	627.1	250.2	376.9	143	13
Rapport	2.3	3.1	1.7	1.8	-

sation de la borne inférieure n'est pas très utile pour l'optimisation. Elle arrête très rarement une itération majeure. Par contre, le choix du point initial, et à un degré moindre une solution primale initiale, peuvent avoir un effet sensible sur l'efficacité de la procédure de stabilisation.

Une stratégie de mise à jour qui modifie fréquemment le problème maître n'est pas le meilleur choix. Une stratégie qui tient compte du résultat de l'optimisation du problème stabilisé sans trop modifier le problème à résoudre et qui garde une fonction de pénalité constante autour du centre de stabilité est à conseiller. Des intervalles plus larges permettent de réduire le nombre d'itérations majeures. Cependant, il y a un compromis à faire avec le coût de la résolution des problèmes stabilisés et le coût de la réoptimisation.

L'élimination des colonnes de stabilisation inutilisées à l'optimalité d'un problème stabilisé permet de nettoyer le problème maître et force en quelque sorte l'utilisation des autres variables de stabilisation en cas de besoin. Par contre, l'arrêt des itérations majeures après un certain nombre d'itérations n'est pas très bénéfique. Un critère d'arrêt qui tient compte de la qualité de la solution pourrait donner de meilleurs résultats. Enfin, l'enrichissement de la direction de déplacement par des directions qui tendent à augmenter l'objectif ou à se rapprocher d'une solution optimale se sont avérés d'un apport appréciable à l'efficacité de la stabilisation.

5.5 Conclusion

Nous avons consacré ce chapitre à l'application de l'approche de stabilisation proposée au chapitre 3 au problème de tournées de véhicules multi-dépôts (*MDVSP*) [75, 34]. Nous avons d'abord donné les deux formulations de problème : problème

de multifactes et problème de partitionnement. Nous avons démontré l'équivalence entre les deux formulations en nombres entiers et entre leur relaxations linéaires en utilisant le principe de décomposition de Dantzig-Wolfe généralisé (*cf. chapitre 2*). Nous avons également montré que dépendamment du type des contraintes d'intégrité prises en compte dans la formulation du sous-problème, la formulation obtenue peut nécessiter des transformations supplémentaires afin d'aboutir à la formulation compacte connue. La relaxation linéaire de la formulation compacte, notée (MP) , et son dual (MD) sont résolus par génération de colonnes. Les résultats de la résolution de problèmes générés aléatoirement ont été présentés pour servir de référence pour l'appréciation de la qualité de l'algorithme de stabilisation utilisé.

Ensuite, nous avons appliqué l'algorithme de stabilisation du chapitre 3 à la paire de problèmes $((MP), (MD))$. Nous avons commencé par donner les formulations des problèmes stabilisés et une condition sur les pénalités qui seront utilisés dans l'implantation de l'algorithme. Puis, nous avons proposé une procédure d'initialisation basée sur un problème de tournées de véhicules à un seul dépôt ($SDVSP$) obtenu à partir de $MDVSP$. Cette procédure fournit un multiplicateur initial réalisable pour le dual (MD) , une borne inférieure sur la valeur optimale de (MP) et (MD) , une solution primale initiale en nombres entiers ainsi qu'une borne supérieure sur la valeur optimale du problème en nombres entiers de $MDVSP$. Nous avons également proposé une borne inférieure sur la valeur optimale de (MP) calculée à chaque itération de génération de colonnes lors de la résolution d'un problème stabilisé. Plusieurs stratégies de mise à jour ont été testées afin d'en choisir une pour les tests qui ont suivi. Des astuces basées sur l'arrêt prématuré de la résolution des problèmes stabilisés par génération de colonnes et sur l'élimination des colonnes de stabilisation inutilisées à l'optimalité des problèmes stabilisés ont été testées dans le but de voir leur effet sur l'efficacité de l'algorithme. Comme dernière technique d'accélération, nous avons enrichi la direction de déplacement du centre de stabilité en utilisant cinq

directions calculées à partir de directions de montée de l'objectif dual ou à partir d'un sous-gradient de la fonction duale au point optimum du problème stabilisé.

Par la suite, nous avons testé l'algorithme sur les six problèmes utilisés pour la résolution par génération de colonnes. Les résultats ont montré qu'une stratégie de mise à jour qui modifie légèrement la fonction de pénalité au début du processus et la garde constante autour du centre de stabilité à la fin du processus est la plus robuste et procure un comportement plus stable à l'algorithme. La stratégie qui garde la fonction de pénalité constante autour du centre de stabilité a donné de meilleurs résultats que la stratégie qui consiste à modifier les paramètres de la fonction de pénalité à chaque itération majeure dépendamment de la position de la solution optimale duale des problèmes stabilisés par rapport aux intervalles de confiance. Parmi les astuces proposées, l'élimination des colonnes inutilisées à l'optimalité des problèmes stabilisés s'est avérée bénéfique pour l'efficacité de l'algorithme. L'arrêt prématuré d'une itération majeure basé sur le nombre d'itérations mineures n'a pas permis d'accélérer l'algorithme de façon régulière. Un critère d'arrêt basé sur la qualité de la solution produirait probablement de meilleurs résultats. L'enrichissement de la direction de déplacement du centre de stabilité par les cinq directions proposées a été d'un apport appréciable à l'algorithme : le temps de calcul est réduit et le nombre d'itérations de génération de colonnes est diminué.

Enfin, les tests de l'algorithme où nous avons intégré plusieurs techniques de stabilisation ont montré l'apport significatif de la stabilisation à la stabilité de la résolution par génération de colonnes.

Chapitre 6

Stabilisation du problème de découpe binaire

Le problème de découpe binaire (*BCSP*) se définit comme suit. Étant donné un ensemble de rouleaux de longueur L , il s'agit de les découper en items de longueurs l_i ($i = 1, \dots, m$). La demande pour chacun de ces items est égale à 1. Une façon de découper un rouleau en un sous-ensemble d'items tout en respectant sa capacité est appelé patron réalisable. L'ensemble de tous les patrons réalisables est noté Ω . Il s'agit de satisfaire les demandes de tous les items i ($i = 1, \dots, m$) à l'aide des patrons réalisables. Afin d'écrire la formulation de *BCSP*, nous définissons les variables binaires x_p ($p \in \Omega$) : x_p compte le nombre de rouleaux coupés suivant le patron p . Les constantes binaires a_{ip} ($i = 1, \dots, m; p \in \Omega$) comptent le nombre de fois que l'item i est découpé dans le patron p . L'objectif est de minimiser le nombre de rouleaux découpés. Dans le but de réduire la symétrie du problème, les items sont numérotés suivant l'ordre décroissant de leur longueur.

Nous donnons la formulation continue du primal ainsi que celle de son dual afin de pouvoir observer l'effet de la stabilisation.

$(BCSP)$		$(BCSD)$
$\begin{aligned} & \text{Min } \sum_{p \in \Omega} x_p \\ & \text{sc } \left\{ \begin{array}{l} \sum_{p \in \Omega} a_{ip} x_p = 1, \quad i = 1, \dots, m \\ 0 \leq x_p \leq 1, \quad p \in \Omega \end{array} \right. \end{aligned}$		$\begin{aligned} & \text{Max } \sum_{i=1}^m \pi_i \\ & \text{sc } \left\{ \begin{array}{l} \sum_{i=1}^m a_{ip} \pi_i \leq 1, \quad p \in \Omega \end{array} \right. \end{aligned}$

Les approches les plus intéressantes pour la résolution de *BCSP* ont été proposées par Vance *et al.* 95 [1], Vanderbeck 96 [90], Valério de Carvalhò 97 [87] et Ben Amor 97 [4]. Ces auteurs ont tous remarqué que l'agrégation des items est très bénéfique pour la résolution de la relaxation linéaire. Les temps de calculs ainsi que le nombre d'itérations de génération de colonnes sont réduits d'une manière très significative.

La première contribution de ce chapitre est de situer l'idée de l'agrégation dans le cadre de la stabilisation par les inégalités partiellement valides duales-optimales introduites dans le chapitre 4. Nous montrerons aussi que l'agrégation est en fait une implantation implicite de certaines contraintes sur les variables duales. Cependant, l'agrégation des items de longueur identique réduit grandement la taille des problèmes résolus et rend leur résolution extrêmement facile. La deuxième contribution principale de ce chapitre consiste à tester l'approche de stabilisation en imposant explicitement les contraintes de stabilisation dans le but de déterminer l'apport intrinsèque de la stabilisation. Nous décrivons les modifications subies par les problèmes primal et dual avant de détailler l'implantation de la stabilisation. Les résultats des deux façons d'imposer la stabilisation sur des problèmes connus dans la littérature sont comparés aux résultats de la résolution par génération de colonnes. Ces résultats permettent de conclure à la grande utilité des inégalités valides utilisés dans la stabilisation de la résolution par génération de colonnes.

6.1 Résolution par génération de colonnes

Une solution entière au problème est obtenue par une procédure de branchement. À chaque nœud de l'arbre de branchement, le problème maître est résolu par génération de colonnes. À chaque itération de génération de colonnes, un sous-problème est résolu dans le but de générer des colonnes de coût réduit négatif.

Le sous-problème peut être formulé comme un problème de sac-à-dos unidimensionnel de capacité L [40, 1, 90] ou comme un problème de plus court chemin avec contraintes de ressource [4]. D'autres formulations sont également possibles. La formulation du sous-problème dépend de la formulation originale qu'on décompose pour obtenir la formulation donnée plus haut. Pour une revue exhaustive des principales formulations du problème de découpe unidimensionnelle, voir Ben Amor [4].

Nous considérons la formulation du sous-problème en plus court chemin avec contraintes de capacité (*SPC*). Les résultats obtenus avec cette formulation [4] et ceux par la formulation en problème de sac-à-dos [1] montrent que la formulation du sous-problème n'influence pas la rapidité de résolution du problème maître. De plus, la formulation du sous-problème en *SPC* permet de garder les mêmes formulations pour le problème maître et le sous-problème dans tous les nœuds de branchement.

Les résultats que nous présentons ici ont été obtenus sur des problèmes générés aléatoirement. Certains de ces problèmes proviennent de la librairie OR [3]. Cinq classes de problèmes sont considérées. Les quatre premières classes sont composées de problèmes générés suivant des distributions uniformes. Pour la dernière classe, les triplets, les problèmes sont générés suivant une procédure particulière. Nous décrivons dans ce qui suit ces cinq classes.

- u50020100 : La longueur des rouleaux $L = 150$. Les longueurs l_i ($i = 1, \dots, 500$) des items sont générés uniformément dans $[20, 100]$.
- u100020100 : La longueur des rouleaux $L = 150$. Les longueurs l_i ($i = 1, \dots, 1000$) sont générées uniformément dans $[20, 100]$.
- u5001100 : La longueur des rouleaux $L = 150$. Les longueurs l_i ($i = 1, \dots, 500$) des items sont générés uniformément dans $[1, 100]$.
- u10001100 : La longueur des rouleaux $L = 150$. Les longueurs l_i ($i = 1, \dots, 1000$) sont générées uniformément dans $[1, 100]$.

– t501 : Ces problèmes sont construits de façon que la solution optimale soit formée par des patrons ayant exactement trois items et que la perte soit nulle. Pour chaque classe, dix instances sont générées. Les instances des classes u50020100, u100020100 et t501 sont prises dans la librairie de problèmes d'optimisation OR [3].

Une procédure de pré-traitement due à Martello et Toth [63] permet de réduire le nombre d'items en moyenne de 49.7% pour u50020100, de 53.5% pour u100020100, de 37.0% pour u5001100 et de 41.1% pour u10001100. Pour les triplets, cette procédure n'élimine aucun item. Le tableau (6.1) présente les caractéristiques des problèmes résolus. L, IMIN et IMAX désignent respectivement la longueur des rouleaux, les longueurs minimale et maximale d'un item. nBP est le nombre d'items dans la version initiale des problèmes et nPrepBP et le nombre d'items restants après l'application de la procédure de pré-traitement. Les données sont calculées en prenant la moyenne sur 10 problèmes.

Tableau 6.1 – *BCSP* : Caractéristiques des problèmes résolus.

Problème	L	IMIN	IMAX	nBP	nPrepBP
u50020100	150	1	100	500	251.2
u100020100	150	1	100	1000	465.4
u5001100	150	1	100	500	314.8
u10001100	150	1	100	1000	589.0
t501	1000	250	500	501	501.0

Nous présentons les résultats obtenus sur ces instances dans le tableau (6.2). lr, mp et sp désignent le temps de calcul moyen (en secondes) de la relaxation linéaire (*BCSP*), du problème maître et du sous-problème, respectivement. itr est le nombre d'itérations de génération de colonnes nécessaires pour l'obtention de la solution optimale continue de (*BCSP*) et col est le nombre total de colonnes générées. Ces résultats ont été obtenus sur une machine SUN Ultra-10/440 (int95=18.1, fp95=22.7,

640M).

Tableau 6.2 – *BCSP* : résolution par génération de colonnes

Problème	lr(s)	mp(s)	sp(s)	itr	col
u50020100	27.2	7.0	20.2	63.9	46923.0
u100020100	242.9	63.2	179.6	94.6	152521.5
u5001100	105.2	28.8	76.4	66.4	3556.9
u10001100	1040.3	409.9	630.4	92.3	7380.7
t501	522.4	57.2	465.2	239.3	194944.9

L'analyse de ces résultats a permis de noter le nombre élevé d'items de longueur très petite dans les instances des problèmes générés uniformément. De plus, parmi ces items, plusieurs sont de longueur identique. Ceci entraîne beaucoup de symétrie entre les solutions primales et surtout duales. Nous montrons dans la suite comment nous pouvons briser la symétrie entre les items identiques afin d'accélérer la résolution de la relaxation linéaire de (*BCSP*) en utilisant des propriétés de certaines variables duales à l'optimalité.

6.2 Stabilisation

Nous allons montrer que le dual (*BCSD*) possède des solutions optimales ayant une propriété particulière qui permettra de briser la symétrie entre les items identiques. Le théorème suivant montre que les variables duales des items identiques peuvent être "identiques" dans une solution optimale de (*BCSD*).

Théorème 6.1 : *Il existe une solution optimale π^* de (*BCSD*) qui vérifie :*

$$l_i = l_j \Rightarrow \pi_i^* = \pi_j^*. \quad (6.1)$$

Preuve: Soit i et j deux items de longueur identique, c'est-à-dire que $l_i = l_j$ et soit π une solution réalisable de $(BCSD)$. Remarquons d'abord que les contraintes qui définissent le domaine de $(BCSD)$ correspondent aux patrons réalisables. Comme les longueurs des items sont identiques, i et j sont présents dans les mêmes patrons réalisables. Donc π_i et π_j figurent avec les mêmes variables dans les contraintes du dual.

Considérons maintenant la solution duale $\bar{\pi}$ définie par :

$$\begin{aligned}\bar{\pi}_l &= \pi_l, \text{ si } l \notin \{i, j\} \\ \bar{\pi}_i &= \bar{\pi}_j = \max\{\pi_i, \pi_j\}.\end{aligned}$$

D'après les remarques précédentes, cette solution est forcément réalisable. De plus, les coûts étant positifs ($= 1$), $\bar{\pi}$ est de meilleur coût que π .

Donc, pour toute solution réalisable de $(BCSD)$, il existe une solution de meilleur coût vérifiant (6.1). D'où on peut déduire que $(BCSD)$ admet au moins une solution optimale $\bar{\pi}$ vérifiant : $l_i = l_j \Rightarrow \bar{\pi}_i = \bar{\pi}_j$. ■

Ce théorème nous autorise à résoudre les problèmes $(BCSP)$ et $(BCSD)$ tout en visant les solutions duales vérifiant la propriété (6.1). Nous proposons deux manières d'imposer ces contraintes dans le problème. Nous présentons d'abord l'imposition explicite des contraintes.

6.2.1 Implantation explicite de la stabilisation

Il s'agit d'ajouter au dual l'ensemble de contraintes suivant :

$$-\pi_i + \pi_j = 0, \quad \forall (i, j), \quad l_i = l_j \tag{6.2}$$

Pour chacune de ces contraintes, une colonne ayant un coefficient -1 à la ligne i , 1 à la ligne j et 0 ailleurs est ajoutée au problème maître. Cette colonne est de la forme

suivante

$$[0, \dots, 0, -1, 0, \dots, 0, +1, 0, \dots, 0]^T.$$

La variable y_{ij} correspondante peut être positive ou négative et a un coût nul. Afin d'éviter les répétitions, les contraintes (et colonnes) sont incluses seulement pour $i > j$.

Les formulations résultantes pour les problèmes stabilisés sont :

$$\begin{array}{c}
 (SBCSP) \\
 \begin{array}{l}
 \text{Min } \sum_{p \in \Omega} x_p \\
 \text{sc } \left\{ \begin{array}{l}
 \sum_{p \in \Omega} a_{ip} x_p + \sum_{j > i: l_i = l_j} y_{ij} - \sum_{j < i: l_i = l_j} y_{ji} = 1, \quad i = 1, \dots, m \\
 0 \leq x_p \leq 1, \quad p \in \Omega
 \end{array} \right.
 \end{array} \\
 (SBCSD) \\
 \begin{array}{l}
 \text{Max } \sum_{i=1}^m \pi_i \\
 \text{sc } \left\{ \begin{array}{l}
 \sum_{i=1}^m a_{ip} \pi_i \leq 1, \quad p \in \Omega \\
 \pi_i - \pi_j = 0, \quad (i, j) : l_i = l_j, \quad j > i
 \end{array} \right.
 \end{array}
 \end{array}$$

Notons que le nombre de contraintes possibles dans le dual peut être très élevé. Le nombre de colonnes ajoutées au primal serait alors très élevé et pourrait affecter l'efficacité de la résolution. À l'image de l'idée qui consiste à utiliser l'ordre sur les items afin d'éviter d'inclure une contrainte (ou une colonne) deux fois, plusieurs techniques permettraient de réduire le nombre de colonnes ajoutées au problème maître. Nous en énumérons quelques-unes ci-après.

- Au lieu d'inclure les contraintes pour chaque item, on trouve un ensemble minimum de contraintes pour un ensemble d'items identiques. Supposons que

les items de longueur l_i sont les items $i, i + 1, \dots, i + k$. Alors, k contraintes sont suffisantes pour imposer les conditions sur les variables duales. On ajoute les contraintes

$$\pi_i = \pi_{i+l}, l = 1, \dots, k$$

ou les contraintes

$$\pi_l = \pi_{l+1}, l = i, \dots, i + k - 1.$$

Le nombre de colonnes (contraintes) ajoutées est alors $O(m)$.

- Vu que ce sont les petits items qui présentent le plus grand nombre d'items identiques, on pourrait juste s'intéresser aux items de grande taille. Ainsi, le nombre de colonnes ajoutées ne serait pas très élevé. Cependant, l'instabilité des variables duales correspondant aux petits items peut causer de la difficulté à la résolution vu leur grand nombre.
- Généralement, ce sont les petits items qui présentent un grand nombre d'items identiques. Pour les variables duales correspondantes, on peut ajouter seulement un sous-ensemble des contraintes. Après la résolution du problème maître restreint, si deux variables duales correspondant à deux items identiques ne sont pas égales (ou ont des valeurs très différentes), les contraintes correspondantes sont ajoutées au dual.

On peut ainsi commencer avec un nombre limité de contraintes ajoutées pour chaque groupe d'items identiques, par exemple 2, et ajouter les contraintes seulement quand elles ne sont pas vérifiées par la solution du problème maître restreint.

- Afin de rendre l'idée précédente plus efficace, on peut éliminer certaines colonnes "inutiles". Par exemple, si une variable est nulle pour un certain nombre d'itérations de génération de colonnes (tout en vérifiant la contrainte associée), on peut juger qu'elle est inutile et éliminer la colonne correspondante du problème maître.

Obtention d'une solution primale réalisable : Les colonnes ajoutées au primal permettent la surcouverture de certains items et la sous-couverture d'autres. Or, les colonnes de stabilisation lient les items de longueur identique entre eux. Il est donc possible de substituer les items couverts plus d'une fois par des items de même longueur non-couverts. Et on obtient une solution primale réalisable pour (P) .

6.2.2 Imposition implicite de la stabilisation

Les contraintes (6.2) peuvent être imposées implicitement dans les problèmes $(BCSP)$ et $(BCSD)$. Si on définit I l'ensemble des items de longueurs différentes notées l_e ($e \in I$), b_e ($e \in I$) la demande des items de longueur l_e , et a_{ep} ($e \in I, p \in \Omega$) le nombre d'items de longueur l_e dans le patron p , les formulations du primal et dual stabilisés deviennent :

$$\begin{array}{c|c} (ACSP) & (ACSD) \\ \hline \begin{array}{l} \text{Min } \sum_{p \in \Omega} x_p \\ \text{sc } \left\{ \begin{array}{l} \sum_{p \in \Omega} a_{ep} x_p = b_e, \quad e \in I \\ x_p \geq 0, \quad p \in \Omega \end{array} \right. \end{array} & \begin{array}{l} \text{Max } \sum_{e \in I} b_e \pi_e \\ \text{sc } \left\{ \begin{array}{l} \sum_{e \in I} a_{ep} \pi_e \leq 1, \quad p \in \Omega. \end{array} \right. \end{array} \end{array}$$

Une solution réalisable de (CSD) dans laquelle toutes les variables duales correspondant à des items de longueur identique ont la même valeur fournit une solution réalisable de $(ACSD)$. Il suffit d'affecter à π_e ($e \in I$) la valeur des variables duales correspondants aux items de longueur l_e . La valeur de l'objectif est la même vu que b_e est le nombre d'items de longueur l_e .

Réciproquement, étant donnée une solution réalisable π de $(ACSD)$, il suffit d'affecter la valeur π_e à toutes les variables duales de (CSD) correspondant aux items de longueur l_e . Le coût est le même pour la même raison évoquée ci-dessus.

De plus, la correspondance est injective. D'où l'équivalence des deux façons d'imposer les contraintes de stabilisation.

Cette façon d'imposer les contraintes sur les variables optimales duales permet de réduire le nombre de contraintes dans le primal sans ajouter des colonnes supplémentaires. La taille du problème est alors réduite.

Enfin, étant donné une solution réalisable de $(ACSP)$, on obtient une solution réalisable de (CSP) en distribuant les b_e ($e \in I$) items de longueur l_e sur les patrons qui couvrent l'item e .

6.3 Résultats numériques

Nous présentons les résultats obtenus sur les instances de problèmes présentés précédemment. D'abord, nous donnons les résultats obtenus en imposant les contraintes implicitement vu que nous nous attendons que cette approche fournira les meilleurs résultats numériques. Nous présenterons ensuite les résultats dans le cas où les contraintes sont imposées explicitement. Ces résultats permettront de conclure sur l'apport intrinsèque des contraintes ajoutées. Les résultats sont obtenus en calculant la moyenne des résultats de 10 problèmes de chaque classe.

6.3.1 Imposition implicite de la stabilisation

Le tableau (6.3) donne les nouvelles tailles des problèmes suite à l'agrégation des items de même longueur. Il est clair que les tailles des problèmes sont réduites de manière très significative ce qui devrait faciliter énormément leur résolution. Les

notations utilisées sont les mêmes qu'au tableau (6.1). En plus, n_{CS} et n_{PrepCS} désignent le nombre d'items résultant de l'agrégation des items de même longueur et le nombre restant d'items suite à l'application de la procédure de pré-traitement.

Tableau 6.3 – *BCSP* : caractéristiques des problèmes avec l'utilisation implicite des inégalités partiellement valides duales-optimales.

Problème	L	IMIN	IMAX	nBP	nCS	nPrepBP	nPrepCS
u50020100	150	1	100	500	80.7	251.2	51.7
u100020100	150	1	100	1000	81.0	465.4	53.7
u5001100	150	1	100	500	98.9	314.8	71.5
u10001100	150	1	100	1000	99.0	589.0	71.7
t501	1000	250	500	501.0	194.3	501.0	194.3

Le tableau (6.4) contient les résultats pour la génération de colonnes sans stabilisation (Std) et avec stabilisation imposée implicitement (Stab Imp). Pour chaque classe de problèmes, nous donnons le temps de calcul moyen (en secondes) pour la résolution du problème (lr), du problème maître (mp) et du sous-problème (sp). Le nombre moyen d'itérations de génération de colonnes est désigné par itr et le nombre moyen de colonnes générées est désigné par col. Ces résultats ont été obtenus sur le même type de machine (SUN Ultra-10/440, int95=18.1, fp95=22.7, 640M).

L'effet de la stabilisation est clairement bénéfique pour la résolution des problèmes sur tous les aspects. Le temps de calcul et le nombre d'itérations de génération de colonnes sont réduits considérablement. La réduction du temps de calcul du sous-problème est quasiment proportionnelle à la réduction du nombre d'itérations. Cependant, le temps de calcul du problème maître est réduit de manière extraordinaire. Ce temps est moins d'une seconde pour les problèmes uniformes et moins de 3 secondes pour les triplets. Ces résultats viennent confirmer les résultats obtenus par

Tableau 6.4 – *BCSP* : utilisation implicite des inégalités partiellement valides duales-optimales.

Problème	Version	lr(s)	mp(s)	sp(s)	itr	col
u50020100	Std	27.2	7.0	20.2	63.9	46923.0
u50020100	Stab Imp	0.3	0.1	0.2	12.8	2053.5
u100020100	Std	242.8	63.2	179.6	94.6	152521.5
u100020100	Stab Imp	0.3	0.1	0.2	13.1	2364.1
u5001100	Std	105.2	28.8	76.4	66.4	3556.9
u5001100	Stab Imp	1.3	0.1	1.2	15.3	6643.9
u10001100	Std	1040.3	409.9	630.4	92.3	7380.7
u10001100	Stab Imp	1.4	0.1	1.3	10.2	4246.6
t501	Std	522.4	57.2	465.2	239.3	194944.9
t501	Stab Imp	20.6	2.4	18.2	124.2	20812.8

Valério de Carvalhò 97 [87] et Ben Amor 97 [4] par l'agrégation des contraintes dans *BCSP*.

La différence entre les performances de la stabilisation sur les deux types de problèmes s'explique par le fait que le nombre d'items de longueur égale dans les problèmes uniformes est beaucoup plus élevé que dans le cas des triplets. Ceci montre un lien direct entre le type de contraintes ajoutées et l'efficacité de la résolution ainsi que la stabilité du processus de convergence des variables duales. Ceci se voit surtout dans la réduction du nombre d'itérations de génération de colonnes ainsi que l'efficacité de résolution du problème maître.

Dans le cas des classes u50020100 et u100020100, le nombre d'items de longueur différentes est plus petit que 80. Dans le cas des classes u5001100 et u10001100, ce nombre est plus petit que 100. Dans le cas des triplets, il est plus petit que 200. Ceci explique la rapidité de la résolution des sous-problèmes et peut avoir une certaine influence sur l'efficacité de la résolution du problème maître. L'analyse des résultats obtenus avec l'imposition explicite des contraintes dans ce qui suit montrera l'apport

intrinsèque de la stabilisation.

6.3.2 Imposition explicite de la stabilisation

Nous testons cette façon d'imposer la stabilisation en ajoutant toutes les colonnes de stabilisation possibles au problème maître (primal). Nous considérerons les mêmes instances testées plus haut afin de pouvoir comparer les deux façons d'imposer la stabilisation et de mesurer l'efficacité des contraintes ajoutées. Les problèmes résultants ont le même nombre d'items mais ont plus de colonnes.

Le tableau (6.5) contient les mêmes informations que la tableau (6.4) pour la résolution sans stabilisation (Std), la version implicite de la stabilisation (Stab Imp) et la présente version explicite (Stab Exp). La colonne m contient le nombre moyen de contraintes dans le problème maître pour chaque version. Notons également que le problème maître de la version Stab Exp contient en plus $O(m)$ colonnes de stabilisation.

L'effet bénéfique de la stabilisation dans sa version explicite est très clair à tous les niveaux. Le pourcentage de réduction du temps de calcul total est de 80.9% pour u50020100, 88.4% pour u100020100, 85.3% pour u5001100, 94.2% pour u10001100 et de 47.0% pour les triplets.

Les mêmes remarques faites pour l'imposition implicite de la stabilisation sont valables pour l'imposition explicite des contraintes de stabilisation. Nous notons cependant que la stabilisation implicite est plus performante que la stabilisation explicite. Mais, l'apport intrinsèque de la stabilisation se mesure sur la réduction du nombre d'itérations de génération de colonnes et l'efficacité de résolution du problème maître. Ces apports sont semblables pour les deux façons d'imposer la stabilisation

Tableau 6.5 – *BCSP* : utilisation explicite des inégalités partiellement valides duales-optimales.

Problème	Version	m	lr(s)	mp(s)	sp(s)	itr	col
u50020100	Std	251.2	27.2	7.0	20.2	63.9	46923.0
u50020100	Stab Imp	51.7	0.3	0.1	0.2	12.8	2053.5
u50020100	Stab Exp	251.2	5.2	0.2	5.0	14.5	7126.9
u100020100	Std	465.4	242.9	63.2	179.6	94.6	152521.5
u100020100	Stab Imp	53.7	0.3	0.1	0.2	13.1	2364.1
u100020100	Stab Exp	465.4	28.8	0.4	28.4	14.4	15352.9
u5001100	Std	314.8	105.2	28.8	76.4	66.4	3556.9
u5001100	Stab Imp	71.5	1.4	0.1	1.2	15.3	6643.9
u5001100	Stab Exp	314.8	15.5	0.5	15.0	10.9	15189.2
u10001100	Std	589.0	1040.3	409.9	630.4	92.3	7380.7
u10001100	Stab Imp	71.7	1.4	0.1	1.3	10.2	4246.6
u10001100	Stab Exp	589.0	60.7	0.7	60.0	7.8	16459.9
t501	Std	501.0	522.4	57.2	465.2	239.3	194944.9
t501	Stab Imp	194.3	20.6	2.5	18.2	124.2	20812.8
t501	Stab Exp	501.0	276.7	6.1	270.6	133.1	88651.8

avec une légère différence due à la différence des tailles des problèmes. La grande différence dans la résolution du sous-problème est due à la grande différence entre les tailles des réseaux. Les réseaux résultant de l'imposition implicite (par agrégation) des contraintes des items sont considérablement plus petits que ceux utilisés quand la stabilisation est imposée de manière explicite (par l'ajout de colonnes).

6.4 Conclusion

Nous avons montré que l'idée de l'agrégation des items identiques, utilisée par plusieurs auteurs pour accélérer la résolution de la relaxation linéaire de *BCSP*, revient en fait à l'imposition dans le dual de contraintes qui stipulent que des variables duales correspondant à des items de longueur identique doivent être égales. En prouvant l'existence d'une solution optimale duale vérifiant cette condition, nous avons en effet montré que l'agrégation est en fait une façon implicite d'implanter des inégalités partiellement valides duales-optimales (*cf.* chapitre 4). Cependant, l'agrégation réduit grandement la taille des problèmes résolus et ne permet pas de déduire clairement le rôle joué par la stabilisation dans l'efficacité de la résolution. Dans le but de montrer l'apport intrinsèque des inégalités valides établies, nous avons résolu les mêmes problèmes en implantant explicitement ces inégalités valides.

Les résultats obtenus par l'agrégation sont supérieurs à ceux obtenus par l'implantation explicite de la stabilisation. Cependant, les améliorations du nombre d'itérations de génération de colonnes et du temps de calcul du problème maître sont semblables dans les deux cas. Ceci montre que les contraintes utilisées apportent une grande stabilité au processus de résolution par génération de colonnes. On peut également affirmer que les inégalités valides utilisées apportent la plus grande part d'efficacité dans le cas de l'agrégation : la différence des temps de résolution du sous-

problème est due à la différence des tailles des réseaux inhérents. On peut conclure ainsi à l'effet bénéfique de l'utilisation de contraintes polyédrale dans le but de viser des multiplicateurs duaux particuliers pour lesquels l'information est disponible.

Chapitre 7

Stabilisation du problème de découpe unidimensionnelle

Le problème de découpe binaire (*BCSP*) est un cas particulier du problème de découpe unidimensionnelle (*CSP*). Dans ce dernier cas, les demandes b_i ($i = 1, \dots, m$) peuvent être plus grandes que 1 et toutes les longueurs l_i ($i = 1, \dots, m$) sont supposées distinctes. Un patron réalisable est défini de la même manière que dans le cas de *BCSP* avec la possibilité qu'un item de longueur l_i puisse figurer plusieurs fois dans un même patron.

Nous utilisons les mêmes notations que dans la section précédente. L'ensemble de tous les patrons réalisables est noté Ω et les variables qui comptent la fréquence d'un patron p ($p \in \Omega$) sont notées x_p . Il faut noter que ces variables ne sont plus contraintes à être binaires vu qu'un patron peut être utilisé plus d'une fois dans une solution réalisable. a_{ip} ($i = 1, \dots, m; p \in \Omega$) compte la fréquence de l'item i dans le patron p . Bien évidemment, a_{ip} prend une valeur entière qui peut être plus grande que 1. L'objectif du problème est bien sûr la minimisation du nombre de rouleaux découpés tout en satisfaisant les demandes pour chaque item.

Avant de donner la formulation de *CSP*, remarquons qu'un problème de découpe unidimensionnelle peut être vu comme un problème de découpe binaire dans lequel on a agrégé les demandes des items de même longueur. De même, si dans un problème de découpe unidimensionnelle on considère chaque unité de demande comme un item

séparé, on obtient un problème de découpe binaire équivalent.

Étant donné les notations introduites précédemment, le problème primal (*CSP*) et son dual (*CSD*) peuvent être formulés comme suit :

<i>(CSP)</i>	<i>(CSD)</i>
$\text{Min } \sum_{p \in \Omega} x_p$	$\text{Max } \sum_{i=1}^m b_i \pi_i$
$\text{sc } \left\{ \begin{array}{l} \sum_{p \in \Omega} a_{ip} x_p \geq b_i, \quad i = 1, \dots, m \\ x_p \geq 0, \quad p \in \Omega \end{array} \right.$	$\text{sc } \left\{ \begin{array}{l} \sum_{i=1}^m a_{ip} \pi_i \leq 1, \quad p \in \Omega \\ \pi_i \geq 0, \quad i = 1, \dots, m. \end{array} \right.$

Les contraintes d'intégrité ont été omises vu que la procédure de résolution consiste à résoudre un problème continu à chaque nœud de l'arbre de branchement.

Dans la formulation précédente, les contraintes de demande sont des inégalités qui exigent que le nombre d'items i découpés soit au moins égal à b_i . Ceci n'affecte pas la valeur optimale du problème. En effet, Gilmore&Gomory 61 [28] ont montré que pour la formulation présentée ci-dessus, il existe toujours une solution où les contraintes de demande sont toutes satisfaites à égalité. L'utilisation des contraintes d'inégalité revient à exiger que les variables duales correspondantes soient non-négatives. Cette modification du problème a été utilisée par la majorité des approches de résolution de *CSP* [90, 87, 4]. Les résultats montrent une certaine amélioration par rapport à l'utilisation des contraintes d'égalité. Ces contraintes sont en effet des inégalités partiellement valides duales-optimales présentées dans le chapitre 4.

Valério de Carvalhò 2000 [88] a utilisé d'autres coupes duales afin de stabiliser la résolution de *CSP* par génération de colonnes. L'auteur justifie la validité de ces coupes d'un point de vue primal et aucun lien n'est fait avec le polyèdre optimal dual.

Nous considérons les mêmes contraintes utilisées par Valério de Carvalhò 2000 [88]. Notre contribution théorique principale est de prouver que ces contraintes sont des inégalités valides sur le polyèdre optimal dual de (*CSP*). Nous aurons ainsi un cas particulier des inégalités valides duales-optimales présentées dans le chapitre 4. Nous montrons par un exemple que ces inégalités valides coupe effectivement le domaine réalisable dual. Ceci a pour conséquence de réduire l'ensemble des multiplicateurs admissibles durant la résolution et rend plus efficace la convergence vers un multiplicateur optimal. L'efficacité de l'obtention d'une solution optimale primale de base est justifiée par des arguments mathématiques. Les résultats de la résolution de problèmes de petite, moyenne et grande taille comparés aux résultats de la résolution par génération de colonnes standard viennent confirmer ces déductions.

Nous testons également une approche de stabilisation par une fonction de pénalité linéaire par morceaux définie autour d'estimations initiales des variables duales calculées à partir des poids des items. Ces estimations sont en fait très proches d'une solution optimale duale. En particulier, nous montrons que pour les problèmes ayant une perte nulle, ces estimations fournissent des variables optimales duales. Les problèmes les plus difficiles étant ceux ayant une perte très faible, ces estimations devraient permettre de stabiliser fortement leur résolution. Les résultats numériques viennent confirmer nos prédictions.

Dans la section qui suit, nous donnons les résultats obtenus pour la résolution de *CSP* par génération de colonnes sans stabilisation. Ensuite, nous donnerons les détails de l'approche de stabilisation que nous appliquerons à ce problème. Par la suite, nous présenterons les résultats numériques obtenus.

7.1 Résolution par génération de colonnes

L'obtention d'une solution entière optimale de *CSP* est assurée via une procédure de branchement. À chaque nœud de l'arbre de branchement, un programme linéaire est résolu par génération de colonnes. Le sous-problème de génération de colonnes est formulé soit comme un problème de sac-à-dos [40, 89, 90], soit comme un problème de plus court chemin sur un graphe en grille [87], soit comme un problème de plus court chemin avec contraintes de capacité [4]. Une revue exhaustive des formulations de *CSP* qui donnent la formulation donnée plus haut par décomposition est présentée dans [4].

Nous avons résolu les relaxations linéaires des problèmes (*CSP*, *CSD*) par génération de colonnes en formulant le sous-problème comme un problème de plus court chemin avec contraintes de capacité (*SPC*). Le réseau sur lequel les colonnes (chemins) sont définies est décrit dans [4].

Pour les tests, nous avons d'abord utilisé les problèmes obtenus suite à l'agrégation de problèmes utilisés pour *BCSP* (cf. chapitre 5). Ces problèmes étant devenus très faciles suite à l'agrégation des items de même longueur, nous avons considéré les trois types les plus difficiles : u5001100, u10001100 et t501. Vu que les temps de calcul sont très petits, le gain principal qu'on peut espérer est au niveau du nombre d'itérations. Nous avons ensuite généré des problèmes de plus grande taille et de difficulté supérieure. Nous avons généré ces problèmes à partir des problèmes u5001100 et u10001100 de la manière suivante. Pour chaque longueur d'item l , nous regardons la demande totale d . Si $d \leq 10$, nous construisons d items de longueurs $10l + i, i = 0, \dots, d - 1$. Si $d > 10$, nous construisons 10 items de longueur $10l + i, i = 0, \dots, 9$ et un item de longueur $10l$ avec une demande égale à $d - 10$. Ces problèmes sont notés hba5001100 et hba10001100. Le nombre moyen d'items

différents par problème est élevé, respectivement de 492.7 et 861.3. La procédure de pré-traitement réduit ce nombre à 396.4 (80.5%) et 479.1 (55.6%) respectivement.

Le tableau 7.1 donne les caractéristiques des problèmes résolus. L , l_{MIN} et l_{MAX} désignent, respectivement, la longueur des rouleaux, la longueur minimale et la longueur maximale d'un item. n_{BP} est le nombre d'items dans la version *BCSP* du problème, n_{CS} est le nombre d'items dans la version *CSP*, n_{PrepBP} est le nombre restant d'items après l'application de la procédure de pré-traitement [63] à la version *BCSP* et n_{PrepCS} est le nombre restant d'items après l'application de la procédure de pré-traitement à la version *CSP*.

Tableau 7.1 – *CSP* : caractéristiques des problèmes.

Problème	L	l_{MIN}	l_{MAX}	n_{BP}	n_{CS}	n_{PrepBP}	n_{PrepCS}
u5001100	150	1	100	500	98.9	314.8	71.5
u10001100	150	1	100	1000	99.0	589.0	71.7
t501	1000	250	500	501	194.3	501	194.3
hba5001100	1500	10	999	500	492.7	314.8	396.4
hba10001100	1500	10	999	1000	861.3	589.0	479.1

Les résultats de la résolution de ces problèmes par génération de colonnes sont présentés dans le tableau (7.2). Les colonnes du tableaux sont les mêmes que celles utilisées pour *BCSP*.

Les problèmes hba5001100 et hba10001100 sont nettement plus difficiles que les autres. Leur nombre de contraintes ainsi que la taille de leurs sous-problèmes sont la cause principale de ce comportement. Étant donné le nombre d'itérations de génération de colonnes et la taille des sous-problèmes (nombre de nœuds et nombre d'arcs), le temps moyen d'une résolution du sous-problème (*SPC*) est de autour de 10s pour hba5001100 et de 30s pour hba10001100. La résolution du sous-problème est

Tableau 7.2 – CSP : Résolution par génération de colonnes.

Problème	lr(s)	mp(s)	sp(s)	itr	col
u5001100	1.4	0.1	1.3	15.3	6643.9
u10001100	1.4	0.1	1.3	10.2	4246.6
t501	20.6	2.4	18.2	124.2	20812.8
hba5001100	1153.7	64.8	1088.9	129.6	325601.2
hba10001100	4301.4	190.1	4111.3	138.3	452500.1

particulièrement coûteuse pour les dernières itérations de génération de colonnes. La moyenne sur les 20 dernières itérations est supérieure à 40s pour hba5001100 avec un maximum autour de 50s et supérieure à 180s pour hba10001100 avec un maximum près de 190s. Ceci est dû à la corrélation entre les valeurs optimales des variables duales et les longueurs des items qui servent dans le calcul des valeurs cumulées de la ressource (capacité) [4].

7.2 Stabilisation

Nous utilisons des inégalités valides duales-optimales introduites dans le chapitre 4. Le théorème suivant montre qu'à l'optimalité, il existe un lien entre les poids des sous-ensembles d'items et les sous-ensembles de variables duales associées. En particulier, l'ordre sur la longueur des items doit être respecté par les valeurs des variables duales correspondantes à l'optimalité.

Théorème 7.1 : Soit $S \subset \{1, \dots, m\}$ et $k \in \{1, \dots, m\}$ tels que $\sum_{i \in S} l_i \leq l_k$.

Alors, toute solution optimale π^* de (CSD) vérifie :

$$\sum_{i \in S} \pi_i^* \leq \pi_k^*. \quad (7.1)$$

Preuve: Soit π^* une solution optimale de *CSD*. π^* est réalisable pour *CSD*, c'est-à-dire qu'elle vérifie :

$$\begin{aligned} \pi_i^* &\geq 0, \quad d = 1, \dots, m \\ \sum_{i=1}^m a_{ip} \pi_i^* &\leq 1, \quad p \in \Omega. \end{aligned}$$

Soit S et k tels que $\sum_{i \in S} l_i \leq l_k$. Tout patron réalisable p contenant l'item k peut être représenté par

$$(l_k; l_j, j \in R_p),$$

où R_p est l'ensemble des items restant qui constituent le patron réalisable p . La contrainte correspondante dans le dual est

$$\pi_k + \sum_{j \in R_p} \pi_j \leq 1.$$

Puisque $\sum_{i \in S} l_i \leq l_k$, il existe des patrons réalisables du type

$$(l_i, i \in S; l_j, j \in R_p).$$

Les contraintes correspondant à ces patrons s'écrivent

$$\sum_{i \in S} \pi_i + \sum_{j \in R_p} \pi_j \leq 1.$$

Supposons maintenant que

$$\sum_{i \in S} \pi_i^* > \pi_k^*.$$

Sachant que π^* est réalisable, on a

$$\pi_k^* + \sum_{j \in R_p} \pi_j^* < \sum_{i \in S} \pi_i^* + \sum_{j \in R_p} \pi_j^* \leq 1.$$

Ainsi pour tout patron réalisable p contenant l'item k ,

$$\pi_k^* + \sum_{j \in R_p} \pi_j^* < 1.$$

Par complémentarité linéaire, toutes les variables primales correspondant aux patrons réalisables p contenant l'item k seront nulles (i.e. $x_p = 0$). Par conséquent, aucune solution primale complémentaire à π^* ne peut être réalisable. Donc, π^* ne peut être optimale (et réalisable). On en déduit alors que toute solution optimale duale π^* doit vérifier

$$\sum_{i \in S} \pi_i^* \leq \pi_k^*.$$

■

Remarque : Le résultat du théorème permet d'affirmer que toute solution optimale vérifie que deux variables duales correspondant à des items de même longueur doivent avoir leur valeurs égales. Ainsi, les contraintes d'égalités utilisées pour *BCSP* au chapitre précédent sont vérifiées par toutes les solutions optimales. Cependant, l'approche utilisée pour *BCSP* utilise l'information disponible et peut être appliquée pour les deux types d'inégalités valides (cf. chapitre 4).

Valério de Carvalhò [88] a utilisé ces mêmes contraintes pour l'accélération de la résolution de *CSP*. La notion de validité utilisée est différente de celle exprimée dans le théorème 7.1. L'auteur montre qu'à partir d'une solution du problème primal obtenu en ajoutant ces contraintes au dual (*CSD*) il est possible de construire une solution réalisable pour (*CSP*) de même coût. Le fait que l'ajout des contraintes au dual entraîne une relaxation du primal et que le second membre des contraintes est nul permet de conclure que la solution ainsi construite est optimale pour *CSP*.

Les deux corollaires suivants donnent des résultats qui sont des cas particuliers du théorème 7.1. Le premier montre qu'à l'optimalité les variables duales doivent être ordonnées suivant les tailles des items correspondants.

Corollaire 7.1 : *Supposons que les items sont ordonnés dans l'ordre décroissant*

suivant leur taille, i.e.

$$l_1 \geq l_2 \geq \dots l_i \geq l_{i+1} \dots \geq l_m.$$

Alors toute solution optimale π^ de (CSD) doit vérifier*

$$\pi_1^* \geq \pi_2^* \geq \dots \pi_i^* \geq \pi_{i+1}^* \dots \geq \pi_m^*.$$

Preuve: Il suffit de considérer les ensembles S de cardinalité 1. Sachant que $l_i \geq l_{i+1}$ pour tout $i = 1, \dots, m - 1$, toute solution optimale π^* doit vérifier $\pi_i^* \geq \pi_{i+1}^*$. L'ordre sur les valeurs des variables duales en découle directement. ■

Le deuxième corollaire présente le cas particulier où $|S| = 2$.

Corollaire 7.2 *Si les items i, j et k sont tels que*

$$l_i + l_j \leq l_k,$$

alors toute solution optimale π^ de CSD vérifie*

$$\pi_i^* + \pi_j^* \leq \pi_k^*.$$

■

Le choix des contraintes à utiliser sera discuté ultérieurement. Il faut cependant noter que plus $|S|$ est grand, plus le nombre de contraintes possibles est élevé.

7.2.1 Effet sur la formulation du dual

L'ajout de contraintes à la formulation du dual (CSD) sera utile seulement si elles permettent de restreindre effectivement le domaine dual tout en conservant des solutions optimales du problème. Le deuxième point est assuré par le

théorème 7.1 ci-dessus. Concernant le premier point, il suffit de montrer que certains points réalisables pour (CSD) ne vérifient pas ces contraintes. Ceci est illustré dans l'exemple suivant.

Exemple 7.1 *Considérons l'inégalité valide définie par k et $S \subset \{1, \dots, m\}$ tels que*

$$\sum_{i \in S} l_i \leq l_k,$$

et soit j un élément de S .

Définissons le multiplicateur π par $\pi_i = 0$ si $i \neq j$ et $\pi_j = 1$. Ce multiplicateur est réalisable pour CSD, mais ne vérifie pas l'inégalité considérée. En effet

$$\sum_{i \in S} \pi_i = 1$$

et

$$\pi_k = 0.$$

Ceci est vrai aussi pour tout multiplicateur réalisable défini de façon à avoir $\sum_{i \in S} \pi_i = 1$ et $\pi_j = 0$ pour $j \notin S$. ■

Ainsi, l'utilisation des contraintes (7.1) permet de restreindre le domaine réalisable dual de manière assez significative. L'ensemble des multiplicateurs admissibles durant la génération de colonnes sera également restreint. Par conséquent, il y aura moins d'oscillations et d'instabilité dans le processus de convergence des variables duales.

Il est également possible de déduire une manière de choisir les contraintes pour un k donné. Tout d'abord, pour un ensemble $S_1 \subset S$, la contrainte définie par S_1 est redondante (dominée) par celle définie par S .

De plus, sachant qu'il s'agit d'un problème de maximisation dont les coûts sont positifs et que les variables doivent être non-négatives, π_k devrait prendre la valeur la plus élevée possible. Il est avantageux de considérer des contraintes (7.1) qui fourniraient la meilleure borne inférieure pour π_k ou encore l'ensemble S qui assurerait la plus grande valeur de $\sum_{i \in S} \pi_i$.

Ainsi, si S_1 est un ensemble d'items qui peut être partitionné en sous-ensembles $S_1^i, i \in S$ qui vérifient $\sum_{j \in S_1^i} l_j \leq l_i$, on aura nécessairement à l'optimalité

$$\sum_{j \in S_1^i} \pi_j \leq \pi_i.$$

Et on peut en déduire qu'il est préférable d'utiliser la contrainte définie par S . Ceci est illustré par l'exemple suivant.

Exemple 7.2 Supposons que $k = 2$, $S = \{4, 6, 7\}$, $S_1 = \{5, 8, 9, 10\}$ et que $l_8 + l_9 \leq l_6$ (les items sont ordonnés dans l'ordre décroissant suivant leur taille).

On peut affirmer qu'à l'optimalité $\pi_5 \leq \pi_4$, $\pi_8 + \pi_9 \leq \pi_6$ et $\pi_{10} \leq \pi_7$. Donc, on aura nécessairement

$$\pi_3 + \pi_6 + \pi_7 + \pi_8 \leq \pi_2 + \pi_4 + \pi_5 \leq \pi_2.$$

Et donc, la contrainte définie par S_1 est dominée par celle définie par S . ■

7.2.2 Effet sur la formulation du primal

L'ajout d'une contrainte (7.1) à la formulation du dual se traduit par l'ajout d'une colonne C dans le primal. Cette colonne est définie par

$$C_i = \begin{cases} 1, & i \in S \\ -1, & i = k \\ 0, & \text{sinon.} \end{cases}$$

La variable correspondant à cette colonne aura un coût nul et sera astreinte à être non-négative.

L'effet bénéfique de la présence d'une telle colonne peut être vu comme suit. Sachant que $\sum_{i \in S} l_i \leq l_k$, chaque fois qu'on remplace l'item k par l'ensemble S des items dans un patron réalisable, on obtient un autre patron réalisable. Supposons qu'à une itération de génération de colonnes, l'ensemble des colonnes présentes dans le problème maître et contenant k est E_k . La présence d'une colonne de stabilisation définie par l'ensemble S et k dans le problème maître permet de considérer implicitement toutes les colonnes correspondant aux patrons réalisables obtenus en remplaçant l'item k par l'ensemble d'items S dans tous les patrons de E_k . Afin d'illustrer ceci, reprenons la représentation d'un patron réalisable contenant k par

$$(l_k; l_j, j \in R_p).$$

La colonne correspondante est définie par

$$C_i^p = \begin{cases} 1, & i \in R_p \\ 1, & i = k \\ 0, & \text{sinon.} \end{cases}$$

La colonne de stabilisation définie par S et k est définie par

$$C_i = \begin{cases} 1, & i \in S \\ -1, & i = k \\ 0, & \text{sinon.} \end{cases}$$

En calculant $C^p + C$, on obtient la colonne C^S définie par

$$C_i^S = \begin{cases} 1, & i \in R_p \\ 1, & i \in S \\ 0, & \text{sinon.} \end{cases}$$

Il est clair que cette colonne correspond au patron réalisable par

$$(l_i, i \in S; l_j, j \in R_p).$$

Remarque : Ce développement nous fournit une façon d'obtenir une solution primale réalisable pour (P) à partir d'une solution réalisable de (SP) . Le coût de la solution de (P) obtenue est le même que le coût de la solution correspondante de (SP) , cette procédure permet d'obtenir une solution optimale de base pour (P) à partir de toute solution optimale (de base) de (SP) .

7.2.3 Remarques sur l'implantation

Le nombre de contraintes du type (7.1) peut être très grand. Il serait raisonnable de considérer seulement un sous-ensemble de ces contraintes. Nous formulons quelques remarques sur les types de contraintes possibles dépendamment de la cardinalité de l'ensemble S .

1. Quand $|S| = 1$, il y a potentiellement $\frac{m(m-1)}{2}$ contraintes. En effet pour chaque k , on a les contraintes

$$-\pi_k + \pi_j \leq 0$$

pour $j = k + 1, \dots, m$. Cependant, il est possible dans ce cas de réduire le nombre de contraintes à considérer sachant que plusieurs sont redondantes. Si on garde seulement les $m - 1$ contraintes

$$-\pi_k + \pi_{k+1} \leq 0$$

pour $k = 1, \dots, m - 1$, toutes les contraintes pour lesquelles $|S| = 1$ seront vérifiées par transitivité.

La vérification implicite des contraintes redondantes se traduit dans le primal par la prise en compte implicite des colonnes correspondantes. Prenons par exemple la contrainte

$$-\pi_k + \pi_{k+2} \leq 0$$

qui est obtenue en sommant les équations

$$-\pi_k + \pi_{k+1} \leq 0$$

et

$$-\pi_{k+1} + \pi_{k+2} \leq 0.$$

Les colonnes correspondant à ces trois contraintes sont notées $C^{k,k+2}$, $C^{k,k+1}$ et $C^{k+1,k+2}$ respectivement. Elles sont définies par

$$C_i^{k,k+2} = \begin{cases} -1, & i = k \\ 1, & i = k + 2 \\ 0, & \text{sinon,} \end{cases}$$

$$C_i^{k,k+1} = \begin{cases} -1, & i = k \\ 1, & i = k + 1 \\ 0, & \text{sinon,} \end{cases}$$

et

$$C_i^{k+1,k+2} = \begin{cases} -1, & i = k + 1 \\ 1, & i = k + 2 \\ 0, & \text{sinon.} \end{cases}$$

On peut voir alors que

$$C^{k,k+1} + C^{k+1,k+2} = C^{k,k+2}.$$

Il est donc possible d'affecter une valeur de α à la colonne $C^{k,k+2}$ en affectant la valeur $\frac{\alpha}{2}$ à chacune des colonnes $C^{k,k+1}$ et $C^{k+1,k+2}$.

2. Quand $|S| = 2$, le nombre de contraintes (colonnes) possibles est encore plus élevé, surtout si les demandes b_i ($i = 1, \dots, m$) sont élevées. Il est important de faire un choix parmi les ensembles $|S|$ possibles pour un item k ($k \leq m-2$). Par exemple, Valério de Carvalhò 2000 [88] choisit, pour un item k , la contrainte

$$-\pi_k + \pi_i + \pi_j \leq 0$$

pour laquelle i est le plus grand item pouvant définir une telle contrainte et j le plus grand item vérifiant $l_k \geq l_i + l_j$. Ceci revient à choisir les premiers i et j (dans l'ordre lexicographique) qui satisfont une telle contrainte. Cette façon est la plus simple vu l'ordre utilisé sur les items. Comme une seule contrainte est définie pour un k donné, le nombre de contraintes possibles ainsi définies est $O(m)$.

Le même raisonnement de la remarque précédente justifie la prise en compte implicite des contraintes (colonnes) redondantes. Cependant, il faut noter que le nombre de contraintes prises en compte implicitement est beaucoup plus élevé dans le cas de la remarque 1.

3. L'utilisation simultanée des colonnes (contraintes) définies pour $|S| \leq 2$ permet de prendre en considération implicitement plusieurs autres colonnes définies pour $|S| = 3$ ou $|S| = 4$. La combinaison de 2 colonnes définies pour $|S| = 2$ ou de 2 colonnes définies pour $|S| = 1$ avec une colonne définie pour $|S| = 2$ permet de prendre en compte implicitement une colonne définie avec $|S| = 3$. La combinaison de 3 colonnes définies pour $|S| = 2$ permet de prendre en compte implicitement une colonne définie pour $|S| = 4$.
4. Une autre idée consisterait à choisir, pour un item k , un ensemble S de cardinalité maximum. Il est possible de considérer un seul ensemble S de la cardinalité trouvée. Bien sûr, il faut s'assurer que les contraintes considérées ne sont pas dominées ou redondantes. Parmi ces ensembles, on peut choisir en priorité les ensembles S dont le poids total est le plus proche de l_i ou choisir les plus gros items possibles dans l'ordre de leur taille. Ce dernier cas revient à remplir un

rouleau fictif de longueur l_k avec les items suivants dans l'ordre $k + 1, \dots, m$.

5. L'utilisation d'un ensemble S dont le poids total est le plus proche de l_i permet de considérer implicitement les patrons qui seraient proches des patrons maximaux. Ce qui est potentiellement bénéfique pour la résolution de la relaxation linéaire vu que la solution optimale est une combinaison convexe des points extrêmes du polyèdre du sous-problème de génération de colonnes (patrons maximaux).

Notons enfin que l'ajout des colonnes de stabilisation permet au problème maître de prendre en considération des colonnes correspondant à des patrons non-extrémaux et qui ne seraient pas générés par le sous-problème de génération de colonnes [4, 93].

7.3 Résultats numériques

Nous considérons les mêmes problèmes qui ont été décrits dans le tableau 7.1. Deux types de contraintes sont utilisées. Nous les distinguons, dans le tableau 7.3, par la cardinalité de l'ensemble S .¹

- $|S| \leq 1$ correspond à l'ajout des contraintes

$$-\pi_k + \pi_{k+1} \leq 0, \quad k = 1, \dots, m - 1.$$

Il y a $(m - 1)$ contraintes de ce type.

- $|S| \leq 2$ correspond à l'utilisation des contraintes définies pour $|S| = 1$ et $|S| = 2$ conjointement. En plus des contraintes ci-dessus, on utilise les contraintes

$$-\pi_k + \pi_i + \pi_j \leq 0, \quad l_k \geq l_i + l_j, \quad k = 1, \dots, m$$

telles que définies dans la remarque 1. Le nombre total de contraintes utilisées est $\leq 2m - 3$.

¹Valério de Carvalho [88] utilise les mêmes contraintes pour les tests.

Les résultats sont présentés dans le tableau 7.3. Les notations sont les même que pour les tableaux précédents.

Tableau 7.3 – *CSP* : stabilisation par des inégalités valides duales-optimales.

Problème	$ S $	lr(s)	mp(s)	sp(s)	itr
u5001100	0	1.4	0.1	1.3	15.3
u5001100	1	1.1	0.1	1.0	9.6
u5001100	2	1.0	0.1	0.9	8.7
u10001100	0	1.4	0.1	1.3	10.2
u10001100	1	1.2	0.1	1.1	8.1
u10001100	2	1.1	0.1	1.0	8.3
t501	0	20.6	2.4	18.2	124.2
t501	1	19.3	2.2	17.3	113.3
t501	2	19.2	2.2	17.0	113.3
hba5001100	0	1153.7	64.8	1088.9	129.6
hba5001100	1	813.5	26.9	786.6	30.5
hba5001100	2	747.9	26.1	721.8	25.0
hba10001100	0	4301.4	190.1	4111.3	138.3
hba10001100	1	2539.3	43.1	2496.2	25.1
hba10001100	2	2324.8	41.9	2282.9	20.4

Le tableau montre une amélioration des performances de la génération de colonnes sur tous les niveaux. Pour les problèmes faciles u5001100 et u10001100, les temps de calculs restent du même ordre (1s) et le nombre d'itérations est réduit de manière assez significative (37.3% et 43.1%). Pour les triplets (t501), l'amélioration des temps de calcul est de l'ordre de 6%. Le nombre d'itération de génération de colonnes est réduit de 8.8% dans les deux cas. L'apport relativement faible de la stabilisation dans le cas de ces problèmes s'explique par leur structure particulière qui entraîne que le nombre de colonnes de stabilisation ajoutées du deuxième type est très petit. De plus la structure de leurs solutions optimales fait que les substitutions fictives des items dans les colonnes présentes dans les problèmes maîtres restreints n'est pas très fructueuse.

Dans le cas des problèmes les plus difficiles, hba5001100 et hba10001100, l'amélioration du temps de calcul total est de l'ordre de 41.0% pour $|S| \leq 1$ et de 46.0% pour $|S| \leq 2$. Le temps des problèmes maîtres est réduit de 77.3% et 78.0% respectivement. Le temps des sous-problèmes est réduit de 39.3% et 44.5% respectivement. Enfin, le nombre d'itérations de génération de colonnes est réduit de plus de 80% dans les deux cas. Ceci, ainsi que l'amélioration au niveau du problème maître, témoignent de la robustesse de l'approche de stabilisation utilisée. Les temps de calcul des sous-problèmes restent néanmoins très élevés. Ceci s'explique par les mêmes arguments utilisés pour la génération de colonnes sans stabilisation. La corrélation forte entre les valeurs des variables duales vers lesquels converge la génération de colonnes $(l_i/L, i = 1, \dots, m)$ et les consommations de la capacité sur les arcs $(l_i, i = 1, \dots, m)$ empêche la dominance des étiquettes et le nombre d'étiquettes gardées durant la résolution de *SPC* est alors énorme. Une méthode d'accélération de la résolution de *SPC* s'impose dans de tels problèmes. Des idées qui permettraient d'accélérer la résolution du sous-problème de génération de colonnes dans le cas de *CSP* sont suggérées dans Ben Amor 97 [4].

En conclusion, l'approche de stabilisation de *CSP* par des inégalités valides sur le polyèdre optimal s'est avérée fructueuse. Les performances de la génération de colonnes ont été améliorées dans tous les problèmes traités. Ces performances varient d'un type de problèmes à l'autre. Ceci est tout à fait normal vu les structures particulières des contraintes ajoutées.

Notons enfin que les performances des méthodes proposées par Vanderbeck 96 [90] et Valério de Carvalhò 2000 [88] sont en grande partie dues à l'utilisation d'heuristiques pour l'initialisation du problème. Ces heuristiques résolvent une suite de problèmes de sac-à-dos pour générer successivement les colonnes les plus remplies. Ces colonnes générées, les problèmes sont alors très proches de l'optimalité dès le

départ et la majorité des colonnes d'une base optimale sont déjà présentes dans le problème maître restreint. Pour les triplets, vu que la perte est nulle, ces heuristiques génèrent très souvent une solution optimale dès le départ. En plus, pour Valério de Carvalhò [88], le temps de calcul pour les triplets sans l'utilisation de la stabilisation est de l'ordre de 3s, ce qui est très petit pour voir l'apport de la stabilisation. Pour d'autres problèmes plus difficiles, l'amélioration des temps de calcul est claire. Il est intéressant de remarquer que la plus grande part de temps de calcul est passée sur la résolution du problème maître restreint, ce qui est contraire à ce qu'on peut observer dans nos résultats.

Stabilisation par une fonction de pénalité linéaire par morceaux : Il est répandu que les problèmes de découpe unidimensionnelle les plus difficiles sont ceux pour lesquels la perte est très petite, voire nulle. Ceci est dû principalement au fait que la résolution des sous-problèmes ne prend pas en compte le remplissage des patrons correspondants aux colonnes générées et que l'écart entre les solutions initiales est assez grand. Cependant, on peut montrer un résultat intéressant pour ce genre de problème. La proposition suivante montre que si la perte d'un problème de *CSP* est nulle, alors on en connaît une solution optimale duale.

Proposition 7.1 *Si (CSP) est telque sa perte est nulle à l'optimalité, alors le multiplicateur défini par*

$$\pi_i^* = \frac{l_i}{L}, i = 1, \dots, m \quad (7.2)$$

est une solution optimale pour le dual (CSD) .

Preuve: Pour un patron réalisable P , on

$$\sum_{i \in P} \pi_i^* = \frac{\sum_{i \in P} l_i}{L} \leq 1.$$

Donc π^* est réalisable pour (CSD) .

De plus,

$$\sum_{i=1}^m b_i \pi_i^* = \frac{\sum_{i=1}^m b_i l_i}{L}.$$

Le deuxième terme est le rapport de l'espace l'espace minimal nécessaire pour découper tous les items sur la longueur d'un rouleau. Il constitue donc une borne inférieure sur la valeur optimale de (CSP) et (CSD) . On conclut alors que π^* défini par (7.2) est une solution optimale pour (CSD) . ■

Les problèmes de triplets ont une perte nulle à l'optimalité. Nous nous attendons donc à ce que l'application de la stabilisation par une fonction de pénalité linéaire par morceaux réduise très significativement le nombre d'itérations nécessaires pour atteindre l'optimalité (cf. chapitre 3). De plus, pour la plupart des problèmes de découpe que nous avons expérimentés, les valeurs des variables duales optimales obtenues sont très proches de $\frac{l_i}{L}, i = 1, \dots, m$. Nous allons tester une approche de stabilisation par une pénalité linéaire par morceaux définie initialement autour de π^* défini par (7.2) sur les problèmes que nous avons considérés dans ce chapitre. Pour les problèmes les plus difficiles (hba5001100 et hba 10001100), la résolution des sous-problèmes risque d'être coûteuse même pour les premières itération à cause de la corrélation entre les valeurs des variables duales et les consommations des ressources sur les arcs. Le tableau 7.4 contient les résultats obtenus. Les notations sont les mêmes que celles des tableaux précédents. Pour chaque classe de problèmes, la première ligne donne les résultats de la résolution par génération de colonnes (cf.

tableau 7.2) et la deuxième ligne donne les résultats de l'utilisation d'une fonction de pénalité linéaire par morceaux pour la stabilisation.

Tableau 7.4 – *CSP* : stabilisation par une fonction de pénalité linéaire par morceaux.

Problème	lr(s)	mp(s)	sp(s)	itr
u5001100	1.4	0.1	1.3	15.3
	1.0	0.1	0.9	6.5
t501	20.6	2.4	18.2	124.2
	7.9	0.5	7.4	12.2
hba5001100	1153.7	64.8	1088.9	129.6
	1087.3	31.1	1056.2	27.0
hba10001100	4301.4	190.1	4111.3	138.3
	2378.5	23.4	2355.1	15.5

Les résultats montrent une amélioration nette dans le nombre d'itérations de génération de colonnes nécessaires pour la résolution des problème à l'optimalité. Les temps de calculs sont aussi améliorés de manière significative. Nous remarquons toutefois que les sous-problèmes des problèmes hba sont très coûteux (voir l'explication plus haut). De plus, sachant que les intervalles utilisés sont petits et centrés autour des multiplicateurs $(\frac{l_i}{L})_{i=1,\dots,m}$, la résolution du sous-problème est très coûteuse dès la première itération.

Pour les triplets, une seule itération majeure est nécessaire vu que $\pi = (\frac{l_i}{L})_{i=1,\dots,m}$ est optimale pour (*CSD*). Pour les autres types de problèmes, au plus deux itérations majeures sont nécessaires pour l'obtention d'une solution optimale pour (*CSP*). Le vecteur π défini ci-dessus est très proche de l'optimalité duale.

Nous avons également testé la stabilisation en utilisant simultanément les inégalités valides sur le polyèdre optimal dual et une pénalité linéaire par morceaux autour de $\pi = (\frac{l_i}{L})_{i=1,\dots,m}$. Les résultats sont donnés dans le tableau 7.5. Pour chaque classe

de problèmes, la première ligne donne les résultats de la résolution par génération de colonnes (*cf.* tableau 7.2) et la deuxième ligne donne les résultats de l'utilisation simultanée d'une fonction de pénalité linéaire par morceaux et des inégalités valides ($|S| \leq 2$).

Tableau 7.5 – *CSP* : stabilisation simultanée par des inégalités valides duales-optimales et une pénalité linéaire par morceaux.

Problème	lr(s)	mp(s)	sp(s)	itr
u5001100	1.4	0.1	1.3	15.3
	1.0	0.1	0.9	6.3
t501	20.6	2.4	18.2	124.2
	6.8	0.4	6.4	11.2
hba5001100	1153.7	64.8	1088.9	129.6
	928.8	32.3	896.5	23.7
hba10001100	4301.4	190.1	4111.3	138.3
	2189.6	27.1	2162.5	14.4

Les remarques sont les mêmes que pour la stabilisation par une fonction de pénalité linéaire par morceaux. On note toutefois la légère amélioration sur tous les aspects due au surplus d'information disponible grâce à l'utilisation des inégalités valides.

Le tableau (7.6) résume tous les résultats de la stabilisation de *CSP* par les différentes techniques proposées. L'effet bénéfique de la stabilisation y est très clair. Les inégalités valides éprouvent cependant certaines difficultés dans la réduction du nombre d'itérations pour les triplets dont la structure particulière réduit l'efficacité de ces contraintes. Par contre, la pénalisation par une fonction linéaire par morceaux réduit de manière significative le nombre d'itérations de génération de colonnes. Toutefois, elle peut causer certaines difficultés liées à la méthode de résolution du sous-problème.

Tableau 7.6 – CSP : résumé des résultats de l'utilisation de la stabilisation.

Problème	$ S $	lr(s)	mp(s)	sp(s)	itr
u5001100	Std	1.4	0.1	1.3	15.3
	$ S \leq 1$	1.1	0.1	1.0	9.6
	$ S \leq 2$	1.0	0.1	0.9	8.7
	<i>pwlpf</i>	1.0	0.1	0.9	6.5
	$ S \leq 2$ et <i>pwlpf</i>	1.0	0.1	0.9	6.3
t501	Std	20.6	2.4	18.2	124.2
	$ S \leq 1$	19.3	2.2	17.3	113.3
	$ S \leq 2$	19.2	2.2	17.0	113.3
	<i>pwlpf</i>	7.9	0.5	7.4	12.2
	$ S \leq 2$ et <i>pwlpf</i>	6.8	0.4	6.4	11.2
hba5001100	Std	1153.7	64.8	1088.9	129.6
	$ S \leq 1$	813.5	26.9	786.6	30.5
	$ S \leq 2$	747.9	26.1	721.8	25.0
	<i>pwlpf</i>	1087.3	31.1	1056.2	27.0
	$ S \leq 2$ et <i>pwlpf</i>	928.8	32.3	896.5	23.7
hba10001100	Std	4301.4	190.1	4111.3	138.3
	$ S \leq 1$	2539.3	43.1	2496.2	25.1
	$ S \leq 2$	2324.8	41.9	2282.9	20.4
	<i>pwlpf</i>	2378.5	23.4	2355.1	15.5
	$ S \leq 2$ et <i>pwlpf</i>	2189.6	27.1	2162.5	14.4

7.4 Conclusion

Dans ce chapitre, nous avons appliqué la stabilisation par des inégalités valides duales-optimales (chapitre 4) au problème de découpe unidimensionnelle (*CSP*). Les contraintes ajoutées ont déjà été utilisés dans la littérature. Cependant, nous avons montré que ces contraintes, qui traduisent un certain lien entre les poids des items et les valeurs des variables duales correspondantes, étaient en effet des inégalités valides sur le polyèdre optimal dual de (*CSP*). Nous avons également expliqué par une discussion mathématique pourquoi la résolution des problèmes stabilisés est plus facile que celle des problèmes originaux. L'utilisation de ces inégalités valides a permis d'améliorer de manière significative les performances de la génération de colonnes. Les tests ont montré que cette approche de stabilisation permet une réduction sensible des temps de calcul et surtout le nombre d'itérations de génération de colonnes pour tous les problèmes résolus.

Nous avons également testé une approche de stabilisation par une fonction de pénalité linéaire définie autour d'une approximation initiale des variables duales calculée à partir des poids des items correspondants (*cf.* chapitre 3). La stabilisation permet de réduire le nombre d'itérations de génération de colonnes d'une manière extraordinaire, même pour les problèmes pour lesquels la stabilisation par des inégalités valides ne réduit que légèrement le nombre d'itérations. Pour ces problèmes (les triplets), les approximations initiales des variables duales sont, en effet, optimales.

L'utilisation simultanée de ces deux techniques de stabilisation a permis de réduire encore plus le nombre d'itérations de génération de colonnes. Les améliorations sont cependant légères ce qui témoigne de la puissance de chacune des approches utilisée séparément. Néanmoins, une technique qui rendrait plus efficace la résolution de *SPC* [14, 4] s'impose afin d'éviter des temps de calculs énormes lorsqu'il existe une

corrélation forte entre les valeurs de variables duales à l'optimalité et les consommations des ressources utilisées par l'optimiseur du sous-problème.

Chapitre 8

Conclusion

L'élaboration de ce travail nous a permis d'approfondir les connaissances dans les domaines de la résolution de problèmes de grande taille, des méthodes de décomposition et des méthodes de résolution des problèmes convexes non-différentiables. Ce travail représente une contribution à la fois à la programmation linéaire, à la résolution des problèmes de grande taille, aux méthodes de stabilisation, à l'optimisation non-différentiable ainsi qu'aux méthodes de descente en général.

L'une des conclusions principales de notre travail est que dans le cadre de la génération de colonnes, l'information duale peut être d'une grande utilité pour résoudre efficacement le primal. Nous avons montré ceci tant sur le plan théorique que sur le plan pratique en appliquant les méthodes développées sur des problèmes connus de la littérature. En particulier, nous avons prouvé que la pénalisation du lagrangien d'un programme linéaire autour d'un multiplicateur optimal permet d'obtenir une solution optimale primale de base. Une autre conclusion importante est que la stabilisation de l'évolution des multiplicateurs duaux a un effet bénéfique sur la stabilité de la résolution du problème primal. Ces développements viennent entre autre illustrer les liens très forts entre un problème et son dual. Nous avons également établi des résultats intéressants pour les problèmes étudiés, c'est-à-dire *MDVSP*, *BCSP* et *CSP*. Nous détaillons dans la prochaine section les contributions de notre travail et consacrons la section suivante aux perspectives et aux travaux futurs basés sur cette thèse.

8.1 Contributions

Le premier chapitre a été consacré à une synthèse des résultats connus sur la résolution des problèmes de grande taille en utilisant le principe de décomposition et la génération de colonnes ou en utilisant la relaxation lagrangienne et les méthodes d'optimisation non-différentiable. Nous avons fait une revue des principaux développements liés à la généralisation du principe de décomposition de Dantzig-Wolfe aux problèmes à variables entières ainsi qu'un survol des difficultés rencontrées par les méthodes classiques face à l'accroissement de la taille des problèmes résolus. Nous avons présenté les premières approches de stabilisation que nous avons regroupées en deux classes. Elles diffèrent par le schéma général de l'algorithme, soit les méthodes de type faisceaux et les méthodes de type proximal. Nous avons également présenté un survol des approches de stabilisation unifiées récentes.

Dans le chapitre 2, nous avons proposé une méthode proximale pour les problèmes d'optimisation convexe non-différentiable. Nous avons généralisé l'idée proposée par Kim *et al.* [43] qui est elle-même une généralisation du principe de l'algorithme du point proximal (Rockafellar 76 [76]). La méthode consiste à résoudre une suite de problèmes dits stabilisés jusqu'à la convergence vers une solution optimale. Les problèmes stabilisés sont définis en pénalisant le problème original à l'extérieur d'un domaine convexe fermé appelé région de confiance. La résolution d'un problème stabilisé fournit une direction de montée qui sert à calculer un point appelé centre de stabilité. Cette résolution définit une itération majeure. Les itérations de l'algorithme utilisé pour la résolution d'un problème stabilisé sont appelées itérations mineures. Dans les travaux de Kim *et al.* [43], cette région est réduite au centre de stabilité courant. Nous avons étudié la convergence de la méthode dans le cas où la région de confiance est de pleine dimension et contient le centre de stabilité courant dans son intérieur. Ainsi, la convergence finie est prouvée même dans le cas des fonctions

non-polyédrales si les problèmes stabilisés sont résolus exactement à chaque itération majeure. Cette condition peut être utile dans le cas non-linéaire où la résolution des problèmes stabilisés produit des solutions approchées. Même si ces solutions ne sont pas optimales, un choix approprié de la largeur des boîtes utilisées permettra de s'assurer que l'optimum du problème stabilisé est à l'intérieur de la région de confiance. Étant donnée la précision visée pour la solution du problème original, les problèmes stabilisés peuvent alors être résolus avec cette même précision. Elle peut être utile aussi pour assurer la convergence dans le primal comme dans le cas d'une fonction affine par morceaux.

Nous avons utilisé un cas particulier de cette approche pour la stabilisation des algorithmes de génération de colonnes et de Kelley dans le chapitre 3. Afin de préserver la linéarité des problèmes résolus, nous utilisons une fonction de pénalité linéaire par morceaux. La convergence duale découlant de la convergence de la méthode générale, la convergence primale est assurée grâce à l'utilisation d'un pavé de pleine dimension comme région de confiance. La fonction de pénalité à cinq morceaux utilisée peut également être vue comme une approximation linéaire externe d'une fonction de pénalité quadratique. Le type des régions de confiance utilisées permet également de montrer qu'il est possible de calculer une solution optimale primale de base à partir d'une solution optimale duale en résolvant un programme linéaire facile. Cette affirmation a été vérifiée sur un problème pratique de grande taille résolu par génération de colonnes. Nous avons enfin montré que la direction de déplacement calculée à chaque itération majeure peut être enrichie par n'importe quelle autre direction sans compromettre la convergence de l'algorithme sous certaines conditions. Bien sûr, ces directions doivent être choisies adéquatement pour accélérer l'évolution des centres de stabilité vers des points proches de l'optimum.

L'approche de stabilisation ainsi développée a été appliquée à la résolution de

la relaxation linéaire de *MDVSP* par génération de colonnes. Nous avons d'abord montré comment l'application du principe de décomposition généralisé à la formulation multiflots de *MDVSP* donne la formulation compacte en nombres entiers. L'implantation de la stabilisation est sujette à quelques conditions reliées à la formulation du problème. La procédure d'initialisation proposée permet de fournir des estimations initiales pour les variables duales, une borne inférieure sur la valeur optimale de la relaxation linéaire, une solution primale entière ainsi qu'une borne supérieure sur le problème en nombres entiers. Les résultats ont confirmé l'efficacité prédite pour la méthode. Des directions d'enrichissement du déplacement des centres de stabilité, basées sur l'augmentation de l'objectif dual ou sur un surgradient de la fonction duale, se sont avérées bénéfiques pour l'efficacité de la méthode.

Nous avons étudié un autre cas particulier de fonction de pénalité pour la stabilisation de la génération de colonnes. Les développements théoriques de cette approche sont faits au chapitre 4. Nous avons introduit deux types d'inégalités valides sur le polyèdre des solutions optimales duales. Les inégalités du premier type, appelées inégalités partiellement valides duales-optimales (*Dual-Optimal Partial Valid Inequalities*), sont vérifiées par au moins une solution optimale duale. Les inégalités du deuxième type, appelées inégalités valides duales-optimales (*Dual-Optimal Valid Inequalities*), sont vérifiées par toutes les solutions optimales duales. L'utilisation de ces inégalités restreint l'ensemble des multiplicateurs admissibles et relaxe le domaine primal, ce qui rend sa résolution plus facile. Cependant, bien que l'optimalité duale ne soit pas compromise, la réalisabilité primale n'est plus assurée. Nous proposons une méthode en deux phases qui permet d'obtenir une paire de solutions optimales primale et duale. La deuxième phase utilise une fonction de pénalité définie autour de l'optimum dual fourni par la première phase. Cette méthode est applicable aux deux types d'inégalités valides. Pour le deuxième type, nous profitons de la richesse de l'information duale pour obtenir une paire de solutions optimales primale et duale

en une seule phase : il suffit de modifier légèrement les contraintes utilisées.

L'approche de stabilisation développée au chapitre 4 a été appliquée à deux problèmes : *BCSP* et *CSP*. Dans le premier cas, nous avons montré que la technique connue d'agrégation des items de longueur identique revient à l'utilisation d'inégalités valides duales du premier type. Nous avons, en effet, montré que la relaxation linéaire de *BCSP* admet une solution optimale duale dans laquelle des variables duales associées à des items de longueur identique ont leur valeurs égales. L'agrégation étant une façon implicite d'implanter ces contraintes, elle réduit la taille des problèmes résolus à tel point qu'il n'est pas facile de décider si l'efficacité de résolution observée est due à la stabilisation ou à la petite taille des problèmes résolus. Nous avons alors effectué les tests en imposant explicitement les contraintes de stabilisation. Ceci nous a permis de conclure à l'apport considérable de l'utilisation de ces contraintes et de l'information duale en général dans le but de stabiliser l'algorithme de génération de colonnes. Dans le cas de *CSP*, nous avons d'abord montré que des coupes déjà utilisées dans la littérature sont en effet des inégalités valides du deuxième type. Nous avons également montré pourquoi la résolution des problèmes stabilisés est plus facile que celle des problèmes originaux. Les problèmes test connus étant très faciles, nous avons générés d'autres problèmes dont les relaxations linéaires se sont avérées très difficiles à résoudre. Notons que nos tests n'utilisent pas les heuristiques connues pour générer des solutions primales très proches de l'optimalité pour des problèmes difficiles ; ceci dans le but de montrer l'apport intrinsèque de la stabilisation. Les résultats viennent confirmer l'effet fortement bénéfique des contraintes de stabilisation pour le comportement de la génération de colonnes.

8.2 Perspectives et travaux futurs

8.2.1 Approche unifiée

Nous avons prouvé la convergence de l'algorithme proposé au chapitre 2 dans le cas où tous les problèmes sont résolus à l'optimalité. Ceci est possible dans le cas des programmes linéaires résolus par génération de colonnes que nous avons étudié au chapitre 3. Dans le cas des problèmes non-linéaires, la résolution approximative des problèmes stabilisés produit des solutions approchées. La précision des problèmes stabilisés est généralement beaucoup plus petite que la précision globale. Ainsi, si on vise une solution ϵ -optimale, on doit résoudre les problèmes stabilisés avec une précision $\eta \ll \epsilon$. Cependant, l'utilisation de largeurs adéquates des boîtes (diamètres des régions de confiance) permet de s'assurer que la nouvelle région de confiance contient une solution optimale du problème stabilisée. Il est alors possible d'utiliser la même précision $\eta = \epsilon$ pour la résolution des problèmes stabilisés.

Une autre question qui mérite d'être étudiée est la généralisation de l'étude de la convergence à d'autres type de région de confiance. L'utilisation de la méthode pour la résolution de programmes convexes non-linéaires permettrait de mieux apprécier son efficacité.

8.2.2 Utilisation d'un multiplicateur optimal dual pour le calcul d'une solution optimale primale de base

Le théorème 3.1 affirme qu'étant donné un multiplicateur dual optimal pour un programme linéaire, il est possible d'obtenir une solution optimale primale de base en résolvant un programme linéaire. Ce programme linéaire est le dual du problème

obtenu en restreignant le dual original à une boîte autour du multiplicateur en question (cela revient aussi à pénaliser le lagrangien défini à l'aide ce multiplicateur). La proposition 3.3 suggère que cette résolution est efficace, ce que nous avons vérifié sur une instance de *MDVSP* (cf. chapitre 3) et sur les problèmes de triplets (cf. chapitre 7). Cette technique peut être utilisée dans une procédure à deux phases pour remplacer la génération de colonnes. Il s'agit de calculer, assez rapidement, un multiplicateur dual optimal (ou ϵ -optimal) par une méthode d'optimisation non-différentiable ou par une méthode de points intérieurs. Ensuite, le programme linéaire défini à partir du multiplicateur trouvé est résolu par génération de colonnes dans le but de déterminer une solution optimale primale de base.

Ce résultat peut aussi être utilisé pour remplacer les procédures de *Crossover*. Dans le cas de programmes linéaires de très grande taille, il est souvent utile de résoudre le problème par une méthode de points intérieurs qui fournit une solution optimale duale et éventuellement une solution primale au centre de la face optimale. À partir de ces solutions, une procédure de *Crossover* est exécutée dans le but d'obtenir une solution optimale primale de base. Cette deuxième phase peut être remplacée par la technique simple et efficace que nous proposons. Afin d'avoir une idée préliminaire du comportement d'une telle méthode, nous avons effectué des tests avec 5 problèmes pour comparer les procédures de *Crossover* de Cplex6.6 [7] avec notre technique. Les tailles des problèmes sont données aux lignes *Contraintes* et *Variables* du tableau 8.1. La ligne *Primopt* indique les temps de calcul obtenus en résolvant les problèmes par l'algorithme primal du simplexe implanté dans cplex6.6. La ligne *Baropt* donne les résultats de la résolution par l'algorithme barrière implanté dans Cplex6.6 [7]. Les lignes *CO Primopt* et *CO Dualopt* donnent les temps de calcul du *Crossover* de Cplex6.6 à partir des solutions fournies par l'algorithme *Baropt* utilisant, respectivement, un algorithme primal et un algorithme dual. Les lignes *CO* 10^{-k} donnent quant à elles les résultats de l'utilisation de notre procédure avec une

boite de largeur 2×10^{-k} à partir de la solution optimale duale fournie par *Baropt*. Chacune des colonnes Pb_i donnent les temps de calcul obtenus pour le problème correspondant.

Tableau 8.1 – Résultats préliminaires du *Crossover*.

	Pb1	Pb2	Pb3	pb4	Pb5
<i>Contraintes</i>	12354	12313	13344	13453	13269
<i>Variables</i>	126329	129349	151665	156841	148040
<i>Primopt(s)</i>	> 10000	> 10000	> 10000	> 10000	> 10000
<i>Baropt(s)</i>	1055.6	1011.1	1341.9	1382.7	1171.3
<i>CO Primopt(s)</i>	13.3	3427.3	> 10000	96.6	6659.3
<i>CO Dualopt(s)</i>	49.6	697.4	1875.7	686.3	777.7
<i>CO $10^{-1}(s)$</i>	188.9	291.3	670.7	631.0	417.4
<i>CO $10^{-2}(s)$</i>	99.5	120.7	554.4	557.1	379.9
<i>CO $10^{-3}(s)$</i>	90.9	102.1	407.2	396.0	352.9
<i>CO $10^{-4}(s)$</i>	87.3	92.2	350.1	346.8	284.5

Les résultats obtenus sur ces problèmes indiquent que la procédure simple de *Crossover* que nous proposons est très prometteuse. Sur les problèmes résolus, elle affiche une certaine stabilité dans les performances par rapport aux deux procédures de Cplex6.6 [7]. De plus, on peut dire que plus la boîte est petite, plus notre méthode est efficace. Cependant, il faut tenir compte de la précision des valeurs des variables duales fournies par l'algorithme *Baropt* surtout qu'il s'agit de problèmes de très grande taille.

Les variables de stabilisation peuvent aussi servir de tolérance sur la réalisabilité du primal. On leur impose alors des bornes supérieures qui seront des coûts de pénalité dans le dual. On obtiendra une fonction de pénalité linéaire à trois morceaux.

Enfin, nous visons à rendre notre méthode encore plus efficace en utilisant la proposition 3.3. En effet, cette proposition nous autorise à éliminer toutes les colonnes

pour lesquelles on est sûr que le coût réduit ne sera pas négatif.

8.2.3 Stabilisation par une fonction de pénalité linéaire par morceaux : utilisation pour la résolution des problèmes en nombres entiers

Les problèmes résolus par génération de colonnes sont souvent les relaxations linéaires d'un problème en nombres entiers qui est résolu par séparation et évaluation progressives. Étant donné que l'algorithme fournit une solution optimale primale de base, les décisions de branchement classiques sont applicables et aucune procédure spéciale de décision n'est nécessaire. Cependant, pour passer d'un nœud à un de ses descendants, on peut profiter de l'information duale à l'optimalité du nœud parent pour initialiser la stabilisation du descendant. Ceci peut permettre une résolution rapide du descendant vu que la différence entre la valeur optimale du nœud parent et ses descendants directs n'est généralement pas grande.

Le problème *MDVSP* constitue un candidat intéressant pour appliquer la stabilisation à la résolution d'un problème en nombres entiers vu que la résolution de sa relaxation linéaire par l'approche proposée au chapitre 3 s'est avérée très efficace (*cf.* chapitre 5). D'autres procédures d'initialisation sont à envisager dans le but de stabiliser seulement certaines variables duales "importantes". De plus, la solution primale entière fournie par la procédure d'initialisation que nous avons utilisée peut être utilisée pour effectuer une élimination de plusieurs arcs du réseau en appliquant le résultat de Hadjar *et al.* 2001 [34]. Il serait également intéressant d'étendre l'étude au problème avec fenêtre de temps qui modélise plusieurs problèmes pratiques. Dans ce cas, il faut définir une procédure pour l'initialisation du problème. Notons enfin que la connaissance d'une solution primale réalisable peut être utilisée pour le calcul d'un solution duale réalisable.

8.2.4 Inégalités valides duales-optimales

Dans le but de rendre la résolution de *BCSP* et *CSP* la plus efficace possible, une technique d'accélération de la résolution du sous-problème de plus court chemin avec contraintes de capacité est nécessaire. Des idées ont été proposées dans [4]. Il serait intéressant de tester ces idées ou de développer des idées qui évitent les temps de calcul énormes observés (*cf.* chapitre 7).

Bibliographie

- [1] BARNHART, C., JOHNSON, E.L. et NEMHAUSER, G.L. et VANCE, P.H. (1994). Solving Binary Cutting Stock Problems by Column Generation and Branch-and-Bound. *Computational Optimization and Applications* Vol. 3, 111-130.

- [2] BAZARAA, M.S., SHERALI, H.D. et SHETTY, C.M. (1993). *Nonlinear Programming : theory and algorithms*. John Wiley & Sons, Inc.

- [3] BEASLEY, J.E. (1990). OR-Library : Distributing test problems by electronic mail. *Journal of the Operational Research Society* Vol. 41, 1060-1072. ([http ://www.ms.ic.ac.uk](http://www.ms.ic.ac.uk))

- [4] BEN AMOR, H. (1997). *Résolution du Problème de Découpe par une Méthode de Génération de Colonnes*. Mémoire de maîtrise, Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, Canada.

- [5] CARPANETO, D., DELL'AMICO, M., FISCHETTI, M. et TOTH, P. (1989). A Branch and Bound Algorithm for the Multiple Depot Vehicle Scheduling Problem. *Networks*, 19 531-548.

- [6] CHEN, G. et TEBoulLE, M. (1993). Convergence Analysis of a Proximal-Like Minimization Algorithm using Bregman Functions. *SIAM J. Optim.* 3, 538-543.
- [7] ILOG. 'ILOG CPLEX 6.6 : User's Manual.
- [8] DANTZIG, G.B. et VAN SLYKE, R.M.(1971). Generalized Linear Programming. D. A. Wismer (ed.), *Optimization Methods for Large-Scale Systems...with applications*, McGraw-Hill, 75-120.
- [9] DANTZIG, G.B. et WOLFE, P. (1960). Decomposition Principle for Linear Programs, *Operations Research* Vol. 8, No 1, 101-111.
- [10] DANTZIG, G.B. (1963). *Linear Programs and Extensions*. Princeton University Press.
- [11] DANTZIG, G.B. (1968). Large Scale Linear Programming. Dans *Mathematics in The Decision Sciences*, Vol. 11 of *Lectures in Applied Mathematics*, American Mathematical Society.
- [12] DESAULNIERS, G., DESROSIERS, J., DUMAS, Y., MARC, S., RIOUX, B., SOLOMON, M.M. et SOUMIS, F. (1997). Crew Pairing at Air France, *European Journal of Operational Research* 97, 245-259.

- [13] DESAULNIERS, G., DESROSIERS, J., IOACHIM I., SOLOMON, M.M., SOUMIS, F. et VILLENEUVE, D. (1997). A Unified Framework for Deterministic Time Constrained Vehicle Routing and Crew Scheduling Problems. In T.G. Crainig and G. Laporte (eds.), *Fleet Management and Logistics*, Kluwer, Norwell, MA, 57-93.

- [14] DESROSIERS, J., DUMAS, Y., SOLOMON, M.M., et SOUMIS, F. (1995). *Time Constrained Routing and Scheduling*. In M.O. Ball *et al.* (eds.), *Network Routing*, Handbooks in Operations Research and Management Science 8. Elsevier Science, Amsterdam, 35-139.

- [15] ECKSTEIN, J. (1993). Nonlinear Proximal Point Algorithms Using Bregman Functions with Applications to Convex Programming. *Math. Oper. Res.* 18, 202-226.

- [16] ELZINGA, J. et MOORE, T.G. (1973). A central cutting plane algorithm for the convex programming problem. *Math. Program.* 8, 134-145.

- [17] ERMOLIEV, Y.M. (1966), Methods of solution of nonlinear extremal problems. *Cybernetics* 2 4, 1-17.

- [18] EVERETT, H. (1963). Generalized Lagrange multiplier method for solving problems of optimal allocation of resources. *Operations Research* Vol. 11,

399-417.

- [19] FARLEY, A.A. (1990). A Note on Bounding a Class of Linear Programming Problems, Including Cutting Stock Problems. *Operations Research* Vol. 38, 922-923.
- [20] FRANGIONI, A. (2002). Genralized Bundle Methods. *SIAM J. On Opt.* (to appear).
- [21] FISHER, M.L. (1981). The Lagrangean Relaxation Method for Solving Integer Programming Problems. *Management Sci.* 27, 1-18.
- [22] GAMACHE, M., SOUMIS, F., MARQUIS, J. et DESROSIERS, J. (1997). A Column Generation Approach for Large Scale Aircrew Rostering Problem. *Operations Research* 47(2), 247-262.
- [23] GAUVIN, J. (1995). *Leçons de programmation mathématiques*. Éditions de l'École Polytechnique de Montréal.
- [24] GEOFFRION, A.M. (1968). Primal Resource-Directive Approaches for Optimizing Nonlinear Decomposable Systems. *Operations Research*, 18, 375-403.
- [25] GEOFFRION, A.M. (1971). Large-Scale Linear and Nonlinear Programming. Dans D. A. Wismer (ed.), *Optimization Methods for Large-Scale Systems...with*

applications, McGraw-Hill, 47–74.

- [26] GEOFFRION, A.M. (1972). Generalized Benders Decomposition. *J. Optim. Theory Appl.* 10, 237-260.
- [27] GEOFFRION, A.M. (1974). Lagrangean Relaxation for Integer Programming. *Math. Program. Study* 2, 82-114.
- [28] GILMORE, P.C. et GOMORY, R.E. (1961). A Linear Programming Approach to the Cutting Stock Problem. *Operations Research* Vol. 11, 849–859.
- [29] GOFFIN, J.-L. (1977). On the Convergence Rates of Subgradient Optimization Methods. *Math. Program.* 13, 329-347.
- [30] GOFFIN, J.-L. (1981). *Convergence Results in a Class of Variable Metric Subgradient Methods*, O.L. Mangasarian, R. R. Meyer et S.M. Robinson (ed.), Nonlinear Programming 4 (Academic Press, New York).
- [31] GOFFIN, J.-L., LUO, Z. et YE, Y. (1993). Complexity Analysis of an Interior Cutting Plane Method for Convex Feasibility Problem, manuscript, McGill University, Canada. *To appear in SIAM J. Optimization.*
- [32] GOFFIN, J.-L. et VIAL, J.,P. (1999). Convex Nondifferentiable Optimization : A survey Focused on the Analytic Center Cutting Plane Method. *Cahiers du*

GERAD, G99-17.

- [33] GULER, O. (1991). On the Convergence of the Proximal Point Algorithm for Convex Minimization. *SIAM J. Control. Optim.* 29, 403-419.

- [34] HADJAR, A., MARCOTTE, O. et SOUMIS, F. (2001). A Branch-and-Cut Algorithm for the Multiple Depot Vehicle Scheduling Problem. *Cahiers du GERAD* G-2001-25.

- [35] HELD, M. , WOLFE, P. et CROWDER, H.P. (1974). Validation of Subgradient Optimization. *Math. Program.* 6, 62-68.

- [36] HIRIART-URRUTY, J. et LEMARÉCHAL, C. (1991). *Convex Analysis and Minimization Algorithms I : fundamentals*. A Series of Comprehensive Studies in Mathematics, Springer, New York.

- [37] HIRIART-URRUTY, J. et LEMARÉCHAL, C. (1991). *Convex Analysis and Minimization Algorithms II : Advanced Theory of bundle methods*. A Series of Comprehensive Studies in Mathematics, Springer, New York.

- [38] IOACHIM, I. (1994). *Planification des Itinéraires d'une Flotte d'Avions avec Contraintes de Synchronisation*. Ph.D. Dissertation, Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal,

Montréal, Canada.

- [39] IUSEM, A. N. et TEBoulLE M. (1995). Convergence Rate Analysis on Non-Quadratic Proximal Methods for Convex and Linear Programming. *Math. Oper. Res.* 20, 657-677.
- [40] KANTOROVICH, L.V. (1960). Mathematical Methods of Organizing and Planning Production. *Management Science* Vol. 6, No 4, 366-393.
- [41] KHACHYIAN, L.G. (1979). A Polynomial Algorithm in Linear Programming. *Doklady Akademiia Nauk SSSR* 244, 1093-1096.
- [42] KELLEY, J.E., Jr. (1960). The Cutting-plane Method for Solving Convex Programs. *SIAM J.*, Vol. 8, No 4, 703-712.
- [43] KIM, S., CHANG, K.-N. et LEE, J.-Y. (1995). A Descent Method with Linear Programming Subproblems for Nondifferentiable Convex Optimization. *Math. Program.* 71, 17-28.
- [44] KIWIEL, K.C. (1985). *Methods of Descent for Nondifferentiable Optimization*. Lectures Notes in Economics and Mathematics, Springer-Verlag, Berlin.
- [45] KIWIEL, K.C. (1989). A Survey of Bundle Methods for Nondifferentiable Optimization. M. Iri et K. Tanabe (ed.) *Mathematical Programming : Recent*

Developments and Applications KTT/Kluwer, Tokyo, 263-282.

- [46] KIWIEL, K.C. (1989). Exact Penalty Functions in Proximal Bundle Methods for Constrained Convex Nondifferentiable Minimisation. Technical Report of Systems Research Institute, Newelska 6, 01-447 Warsaw, Poland.
- [47] KIWIEL, K.C. (1994). A Bundle Method for Minimizing a Sum of Convex Functions with Smooth Weights. Working Paper o IIASA', A-2361 Laxenburg, Austria.
- [48] KIWIEL, K.C. (1997). Approximations in Proximal Bundle Methods and Decomposition of Convex Programs. *JOTA* 84(3), 529-548.
- [49] KIWIEL, K.C. (1996). Complexity of Some Cutting Plane Methods that Use Analytic Centers. *Math. Program.* 74(1).
- [50] KIWIEL, K.C. (1997). Proximal Minimization Methods with Generalized Bregman Functions. *SIAM J. Control. Optim.* 35, 1142-1168.
- [51] KIWIEL, K.C. (1999). A Bundle Bregman Proximal Method for Convex Non-Differentiable Optimization. *Math. Prog.* 85, 241-258.
- [52] LASDON, L.S. et MACKEY, J.E. (1968). An Efficient Algorithm for the Multi-item Scheduling. Systems Research Center Report SRC 68-9, Case

Western University.

- [53] LASDON, L.S. (1972). *Optimization Theory for Large Systems*. Macmillan Series in Operations Research.
- [54] LEE, H.L. et NAHMIAS, S. (1993). Single-Product, Single-Location Models. Dans S.C. Graves, A.H.G. Rinnooy Kan, P.H. Zipkin *Handbooks in Operations Research and Management Science : Logistics of Production and Inventory*, Elsevier Science Publishers B.V. (North-Holland), 3-55.
- [55] LEMAIRE, B. (1989). The Proximal Algorithm. Dans *International Series of Numerical Mathematics*. J.P. Penot eds. 87, 73-87.
- [56] LEMARÉCHAL, C. (1974). An Algorithm for Minimizing Convex Functions. Proceedings IFIP '74 Congress, J.L. Rosenfeld (ed.), North-Holland, Amsterdam, 552-556.
- [57] LEMARÉCHAL, C. (1975). An Extension of Davidon Methods to Nondifferentiable Problems. *Math. Program.* 3, 95-109.
- [58] LEMARÉCHAL, C. (1989). Nondifferentiable Optimization. In G.L. Nemhauser, A.H.G. Rinnooy Kan et M.J. Todd (eds) *Handbooks in Operations Research and Management Science : Optimization*, 1 Elsevier Science Publishers B. V.

(North-Holland), 529-572.

- [59] LEMARÉCHAL, C. (2000). *Lagrangian Relaxation*. INRIA.
- [60] LEVIN, A.Y. (1965). On an algorithm for the minimization of convex functions over convex sets. *Soviet Mathematical Doklady* 6, 286-290.
- [61] MARSTEN, R.E., HOGAN, W.W. et BLANKENSHIP, J.W. (1975). The BOXSTEP Method for Large-scale Optimization. *Operations Research*, Vol. 23 N0 3, 389-405.
- [62] MARSTEN, R.E. (1975). The Use of Boxtep Method in Discrete Optimization. *Math. Program. Study* 3, 127-144.
- [63] MARTELLO, S. et TOTH, P. (1990). *Knapsack Problems : Algorithms and Computer Implementations*. Wiley-Interscience in Discrete Mathematics and Optimization.
- [64] du MERLE, O., VILLENEUVE, D., DESROSIERS, J. et HANSEN, P. (1998). Stabilized Column Generation. *Discrete Mathematics* 194, 229-237.
- [65] MINOUX, M. (1984). *Programmation mathématique : théorie et algorithmes*, 2 Dunod (Bordas), Paris.

- [66] MOREAU, J. J. (1965). Proximité et Dualité dans un Espace Hilbertien. *Bull. Soc. Math. France* 93, 273-299.
- [67] NEAME, P. (1999). *Nonsmooth Methods in Integer Programming*. Ph.D. Dissertation, University of Melbourne, Australia.
- [68] NEMHAUSER, G.L. et Wolsey, (1988). *Integer Programming and Combinatorial Optimization*, J. Wiley & Sons, New York.
- [69] NEMIROVSKII, A. (1994). *unpublished lecture notes*
- [70] NEMIROVSKII, A. et YUDIN, D.B. (1983). *Problem Complexity and Method Efficiency in Optimization*, Wiley Interscience.
- [71] NESTEROV, Y. and NEMIROVSKII, A. (1994) *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Studies in Applied Mathematics, Pennsylvania.
- [72] NESTEROV, Y. (1999) *Introductory Lectures on Convex Programming*.
- [73] POLYAK, B.T. (1966). A General Method of Solving Extremum Problems. *Soviet Mathematics* 8, 593-597.

- [74] POLYAK, B.T. Minimization of Unsmooth Functions. *USSR Computational Mathematics and Mathematical Physics* 9, 14-29.
- [75] RIBEIRO, C.C. et SOUMIS, F. (1994). A Column Generation Approach to the Multi-Depot Vehicle Scheduling Problem. Dans *Operations Research*. Vol. 42, No. 1, 41-52.
- [76] ROCKAFELLAR, R.T. (1976). Monotone Operators and The Proximal Point Algorithm. *SIAM J. Control and Optimization* Vol.14, No. 5, 877-898.
- [77] SHOR, N.Z. (1968). On the Rate of Convergence of the Generalized Gradient Descent Method. *Cybernetics* 4 3, 79-80.
- [78] SHOR, N.Z. (1970). Convergence Rate of the Gradient Descent Method with Dilatation of Space, *Cybernetics* 6 2 94-96.
- [79] SCHRAMM, H. et ZOWE, J. (1992). A Version of the Bundle Idea for Minimizing a Nonsmooth Function : Conceptual Idea, Convergence Analysis, Numerical Results. *SIAM J. on Optimization* 2, 121-152.
- [80] SHAPIRO, J.F. (1993). Mathematical Programming Models and Methods for Production Planning and Scheduling. Dans S.C. Graves, A.H.G. Rinnouy Kan, P.H. Zipkin *Handbooks in Operations Research and Management Science : Logistics of Production and Inventory*, Elsevier Science Publishers B.V.

(North-Holland), 333-370.

- [81] SLATER, M. (1950). Lagrange Multipliers Revisited : a Contribution to Nonlinear Programming. *Cowles commission discussion paper*, Mathematics, 403.
- [82] SOUMIS, F. (1997) Decomposition and Column Generation. Dans *Annotated Bibliographies in Combinatorial Optimization*, M. Dell'Amico, F. Maffioli and S. Martello (ed.), John Wiley & Sons, Ltd.
- [83] STOJKOVIC, M., Soumis, F. et Desrosiers, J. (1997). The Operational Airline Crew Scheduling Problem. *Transportation Science*, 32(3), 232-245.
- [84] TACHEFINE, B. (1997). *Méthode d'Optimisation pour la Planification dans une Mine à Ciel Ouvert*. Ph.D. Dissertation, Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, Montréal, Canada.
- [85] THOMAS, L.J. et McLAIN, J.O. (1993). Production Planning and Scheduling. Dans S.C. Graves, A.H.G. Rinnooy Kan, P.H. Zipkin *Handbooks in Operations Research and Management Science : Logistics of Production and Inventory*, Elsevier Science Publishers B.V. (North-Holland), 333-370.
- [86] VAIDYA, P. (). A New Algorithm for Minimizing Convex Functions over Convex Sets. AT&T Bell Laboratories, Murray Hill, NJ 07974.

- [87] VALÉRIO de CARVALHÒ, J.M. (1997). *Exact Solution of Bin-Packing Problems Using Column Generation and Branch-and-Bound*. Working paper. Dept. Produção e Sistemas, Universidade de Minho, 4709 Braga Codex, Portugal.

- [88] VALÉRIO de CARVALHÒ, J.M. (2000). *Using extra dual cuts to accelerate column generation*. Working paper, Jan 2000. Dept. Produção e Sistemas, Universidade de Minho, 4709 Braga Codex, Portugal.

- [89] VANCE, P.H. (1996). *Branch-and-Price Algorithms for the One-dimensional Cutting Stock Problem*. Rapport de recherche, Department of Industrial Engeneering, Auburn University, Auburn, Alabama 36849-5364.

- [90] VANDERBECK, F. (1996). *On Integer Programming Decomposition and Ways to Enforce Integrality in the Master*. Rapport de recherche, Judge Institute of Management Studies, Cambridge University, Trumpington St., Cambridge CB2 1AG, U.K.

- [91] VANDERBECK, F. (1997). *Computational Study of a Column Generation Algorithm for Bin Packing and Cutting Stock Problems*. Rapport de recherche, Judge Institute of Management Studies, Cambridge University, Trumpington St., Cambridge CB2 1AG, U.K.

- [92] VANDERBECK, F. (2000). On Dantzig-Wolfe Decomposition in Integer Programming and Ways to Perform Branching in a Branch-and-Price Algorithm. *Operations Research*, Vol. 48, No.1, 111-128.
- [93] VILLENEUVE, D. (1999). *Logiciel de Génération de Colonnes*. Ph.D. Dissertation, École Polytechnique de Montréal, 1999.
- [94] WOLFE, P. (1975). A method of Conjugate Gradients for Minimizing Nondifferentiable Convex Functions. *Math. Program.* 3, 145-173.
- [95] YE, Y. (1994). Complexity Analysis of the Analytic Center Cutting Plane Method that Uses Multiple Cuts. Technical Report.
- [96] YE, Y. (1997). *Interior Point Algorithms : Theory and Analysis*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, New York.
- [97] ZIARATI, K. (1996). *Affectation de Locomotives au Train*. Ph.D. Dissertation, Département de Mathématiques et de Génie Industriel, École Polytechnique de Montréal, Montréal, Canada.