



Titre: Résolution du problème de découpe unidimensionnelle par une
Title: méthode de génération de colonnes

Auteur: Hatem Ben Amor
Author:

Date: 1997

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Ben Amor, H. (1997). Résolution du problème de découpe unidimensionnelle par
Citation: une méthode de génération de colonnes [Mémoire de maîtrise, École
Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/6695/>

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

URL de PolyPublie: <https://publications.polymtl.ca/6695/>
PolyPublie URL:

**Directeurs de
recherche:** François Soumis, & Martine Bellaïche
Advisors:

Programme: Non spécifié
Program:

UNIVERSITÉ DE MONTRÉAL

RÉSOLUTION DU PROBLÈME
DE DÉCOUPE UNIDIMENSIONNELLE
PAR UNE MÉTHODE DE GÉNÉRATION DE COLONNES

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

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(MATHÉMATIQUES APPLIQUÉES)
DÉCEMBRE 1997



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.

0-612-33109-1

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

RÉSOLUTION DU PROBLÈME
DE DÉCOUPE UNIDIMENSIONNELLE
PAR UNE MÉTHODE DE GÉNÉRATION DE COLONNES

présenté par: BEN AMOR Hatem

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

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

M. DESROSIERS, Jacques, Ph.D., président

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

Mme BELLAICHE Martine, M. Sc., membre et codirectrice de recherche

M. DESAULNIERS Guy, Ph.D., membre

À mes parents

À mon petit frère

Remerciements

Tout d'abord, je tiens à remercier Jacques Desrosiers pour son aide considérable, ses conseils précieux et ses encouragements qui m'ont permis de mener à bien ce travail.

Mes remerciements s'adressent également à F. Chauny, G. Desaulniers, J. Desrosiers et F. Soumis dont les conseils et les directives m'ont été d'une grande utilité dans la rédaction du présent mémoire.

Je remercie le Ministère de l'enseignement supérieur de la Tunisie ainsi que François Soumis qui ont rendu possible la réalisation de cette recherche.

Merci enfin, à tous ceux qui m'ont aidé durant ces deux années par leurs conseils et leurs encouragements.

Résumé

Ce mémoire s'intéresse à l'étude du problème de découpe unidimensionnelle en le considérant comme un problème de tournées de véhicules. Nous utilisons une formulation multiflot dans laquelle les patrons de découpe sont représentés sur un réseau acyclique.

Nous appliquons ensuite la décomposition de Dantzig-Wolfe à cette formulation et nous résolvons le problème par une méthode de génération de colonnes où le problème maître est formulé en terme des variables de patrons et où le sous-problème est un problème de plus court chemin avec contrainte de capacité. Le branchement est effectué sur les variables de flot du réseau.

Pour les tests, nous considérons tout d'abord le problème de découpe binaire pour lequel toutes les demandes sont unitaires. Les décisions de branchement sont alors portées directement dans le réseau du sous-problème. Ensuite, nous étudions quelques techniques pour accélérer la résolution de la relaxation linéaire et le processus de branchement. Nous montrons enfin, que la résolution de la relaxation linéaire en agrégeant les items de longueur égale est beaucoup plus efficace.

Pour le problème plus général, les décisions de branchement ne peuvent cependant être portées dans le sous-problème sans altérer sa structure: elles seront donc traduites par des contraintes dans le problème maître. Ceci permet de conserver la structure du sous-problème tout le long du processus de résolution. Nous présentons quelques résultats préliminaires pour cette approche. Enfin, nous montrerons com-

ment le processus de branchement peut être accéléré par l'utilisation de diverses techniques.

Ce mémoire a permis de tirer plusieurs conclusions sur la résolution des problèmes de tournées et d'horaires. Bien que le problème de découpe soit facile et présente un saut d'intégrité très petit, sa structure présente une grande symétrie qui rend la résolution instable et difficile. Le présent travail montre que l'utilisation de bornes inférieures et de techniques heuristiques pour l'initialisation peuvent être d'un grand apport pour l'optimisation. Des branchements plus rapides peuvent être obtenus en tenant compte des caractéristiques des tâches à accomplir et par la prise de plusieurs décisions simultanément. La dégérescence du problème dual peut avoir un impact important sur la stabilité du processus de résolution par génération de colonnes, d'où l'intérêt d'avoir une procédure de stabilisation qui contrôle celle-ci au niveau des variables duales.

Abstract

The aim of this thesis is solving the cutting stock problem (CSP) as a constrained vehicle routing problem. The problem is formulated as multi-commodity flow problem where the cutting patterns are represented on an acyclic network.

We apply the Dantzig-Wolfe decomposition technique on this formulation to obtain a master problem formulated in terms of pattern variables and a shortest path problem with capacity constraint as a subproblem. Branching is performed on flow variables.

We first test our approach on the binary cutting stock problem. Branch-and-bound decisions are dealt with by eliminating arcs from the subproblem network. Then, we propose some acceleration techniques in order to improve the performance of the linear relaxation optimization and the branch-and-bound process. We show then that if the constraints corresponding to items of equal length are aggregated, the solution process of the linear relaxation is much faster.

For the general case, considering branching decisions within the subproblem would destroy its structure. Therefore, we had to add new constraints to the master problem so that the subproblem structure is preserved. Preliminary results for this approach are presented and some enhancement features are proposed.

Although the cutting stock problem is easy to solve and its integrality gap is very small, it presents a lot of symmetry which affects the stability of the solution process.

Several conclusions on routing problems raised from this work. Good lower bounds and heuristic initialization may be very beneficial to the problem optimization. More efficient branching can be obtained by taking in account item attributes or by taking several decisions simultaneously. Also, the dual problem degeneracy may affect the stability of a column generation solving process. Hence, stabilization techniques would be very beneficial for hard problems solution.

Table des matières

DÉDICACE	iv
REMERCIEMENTS	v
RÉSUMÉ	vi
ABSTRACT	viii
TABLES DES MATIÈRES	x
LISTE DES TABLEAUX	xiv
LISTES DES FIGURES	xv
CHAPITRE 1: INTRODUCTION	1
CHAPITRE 2: REVUE DE LA LITTÉRATURE	4
CHAPITRE 3: FORMULATIONS CLASSIQUES	8
3.1 Formulation de Kantorovich (1960)	8
3.2 Formulation de Gilmore et Gomory (1961)	10
3.3 Décomposition de la formulation de Kantorovich	13

CHAPITRE 4: FORMULATIONS DE TYPE RÉSEAU 17

4.1 La formulation de Valério de Carvalhò (1996) 17

4.1.1 Description du réseau 17

4.1.2 Formulation mathématique 18

4.1.3 Décomposition de la formulation de Valério de Carvalhò . . . 19

4.2 Formulation multiflot 22

4.2.1 Représentation des patrons par des chemins 23

4.2.2 Un réseau acyclique pour représenter les patrons 25

4.2.3 Formulation mathématique 26

4.2.4 Décomposition de la formulation multiflot 28

4.2.5 Procédure générale de résolution 36

CHAPITRE 5: RÉSULTATS EXPÉRIMENTAUX: CAS BINAIRE 38

5.1 Branchement 39

5.2 Caractéristiques de l'implémentation 42

5.3 Description des problèmes résolus 44

5.4	Résultats préliminaires	46
5.5	Améliorations	52
5.5.1	Branchement	52
5.5.2	Relaxation linéaire	57
CHAPITRE 6: GÉNÉRALISATION		65
6.1	Agrégation des items	65
6.2	Branchement	68
6.3	Autres techniques de branchement	76
6.4	Autres techniques d'accélération	78
CHAPITRE 7: CONCLUSION		81
BIBLIOGRAPHIE		83

Liste des tableaux

5.1	Description des problèmes de tests: uniformes (u) et triplets (t) . . .	47
5.2	Résultats préliminaires (version standard)	49
5.3	Comparaison avec les résultats de Vance <i>et al.</i> (1994)	51
5.4	Caractéristiques des instances choisies	52
5.5	Utilisation de différents critères pour le choix de la variable de bran- chement	54
5.6	Branchement sur plusieurs variables simultanément	57
5.7	Utilisation de la borne de Farley	59
5.8	Au nœud 0, utilisation de la borne LB^2 et initialisation par les meil- leurs paires et triplets	63
5.9	Résultats combinant l'utilisation de la borne LB^2 , l'initialisation par les meilleurs paires et triplets et le branchement simultané sur plu- sieurs variables de flot	64
6.1	Au nœud 0, agrégation des items de même longueur, utilisation de la borne LB^2 et initialisation par les meilleurs paires et triplets	67

6.2	Résultats préliminaires pour la formulation agrégée du problème maître	74
6.3	Utilisation d'une heuristique d'arrondissement à chaque nœud de l'arbre de branchement pour la résolution des deux instances du problème u9	75

Liste des figures

3.1	Exemple illustrant la possibilité de ne pouvoir obtenir la solution optimale avec la technique proposée par Gilmore et Gomory.	12
4.1	Réseau de Valério de Carvalhò (1996) avec $L = 5$, $l_1 = 2$ et $l_2 = 3$. . .	18
4.2	Exemple d'une représentation d'un patron par un chemin.	23
4.3	Représentation des patrons de l'exemple.	24
4.4	Exemple de réseau acyclique pour KP.	25
5.1	Exemple de réseau d'un problème de découpe binaire.	39
5.2	Effet des décisions de branchement sur le réseau de génération des patrons.	42
5.3	Branchement sur plusieurs variables	56
6.1	Exemple d'une solution fractionnaire en terme des chemins qui donne des flots entiers sur les arcs.	71

Chapitre 1

Introduction

Le problème de découpe (*Cutting Stock Problem*) apparaît dans une large variété de secteurs, tels ceux de l'industrie du papier, de l'aluminium et d'autres métaux, du verre, etc. L'intérêt économique de trouver des solutions plus efficaces a poussé plusieurs chercheurs à entreprendre des recherches portant sur ce problème. Sweeny et Paternoster (1991) ont identifié plus de 500 articles qui traitent des problèmes de découpe.

De part sa relation avec l'industrie, le problème de découpe présente plusieurs variantes. De façon générale, il s'agit d'affecter un ensemble de petits items à un ensemble de grands objets tout en satisfaisant certaines contraintes. Dyckhoff (1990) a proposé une classification pour les différents types de problèmes de découpe. Il a utilisé quatre caractéristiques: la dimension, le type d'affectation, les caractéristiques des grands objets, et celles de l'ensemble des petits items. Le premier paramètre compte le nombre de dimensions suivant lesquelles le découpage est effectué. Il y a principalement deux types: la découpe unidimensionnelle et la découpe bidimensionnelle. L'affectation peut être faite de deux manières: déterminer un sous-ensemble de petits items à découper sur tous les grands objets ou sélectionner un sous-ensemble de grands objets pour découper tous les petits items. Trois cas de figures sont possibles pour les types de grands objets: un seul grand objet, plusieurs grands objets identiques et plusieurs grands objets différents. Enfin, la dernière caractéristique

définit la distribution des différentes longueurs des petits items. D'autres paramètres, tels que les restrictions sur les couteaux et la réutilisation de certains items non utilisés peuvent fournir des contraintes additionnelles au problème.

Dans ce travail, nous nous intéressons au problème de découpe unidimensionnelle avec des tailles identiques pour les grands objets. Dans la suite du texte, nous utiliserons les termes de l'industrie du papier. Un rouleau désignera un grand objet et un item désignera un petit item qui devra être découpé sur un rouleau de taille standard. Le problème considéré consiste donc à découper sur de grands rouleaux de taille standard, des petits items dont les tailles sont déterminées par les commandes des clients. Ces dernières sont définies en regroupant les items de même taille dans un même lot. L'objectif de l'optimisation est de minimiser le nombre total de grands rouleaux utilisés, et par conséquent, la perte totale encourue. La perte sur un rouleau est la partie qui reste après avoir découpé tous les items qui serviront à satisfaire les demandes.

Un problème qui est très proche du problème de découpe est celui de la découpe binaire (*Bin Packing Problem* ou *Binary Cutting Stock Problem*). Ce problème peut être vu comme un cas particulier du problème de découpe unidimensionnelle où toutes les demandes sont unitaires. Parmi les applications les plus connues, on trouve le chargement des avions et le placement des circuits imprimés sur une carte mère. En fait, dans ce problème, chaque item est considéré comme une commande à part et pour chacun d'eux, il s'agit de déterminer le rouleau sur lequel il devra être découpé. L'objectif est le même, c'est-à-dire minimiser le nombre de grands rouleaux utilisés. Il est possible de transformer tout problème de découpe en celui de découpe binaire, en associant à chaque item autant de copies qu'il y a d'unités demandées. Toutefois,

l'utilisation de l'approche binaire peut augmenter considérablement la taille d'un problème, et en rendre la résolution plus difficile.

Dans le chapitre suivant, nous ferons une brève revue de la littérature sur les problèmes de découpe unidimensionnelle et binaire, ainsi que sur le problème de tournées de véhicules qui nous a été utile pour accomplir ce travail. Ensuite, nous présenterons les deux formulations classiques de Kantorovich (1960) et de Gilmore et Gomory (1961, 1963), une formulation récente de type réseau due à Valério de Carvalhò (1996), et enfin la formulation multiflot que nous proposons. Par la suite, nous proposons une méthode de résolution basée sur le principe de décomposition de Dantzig et Wolfe (1960). Les expériences numériques sont présentées pour les cas de la découpe binaire.

Chapitre 2

Revue de la littérature

Le problème de la découpe unidimensionnelle peut être formulé comme un programme en nombres entiers et il est particulièrement difficile à résoudre. En fait, il est NP-difficile (Martello et Toth, 1990). Marcotte (1985) a cependant montré que souvent la valeur optimale de la solution peut être obtenue en arrondissant à l'entier supérieur la valeur optimale de la relaxation linéaire obtenue en utilisant le polyèdre des patrons réalisables. Des résultats similaires se trouvent dans Berge et Johnson (1976).

La première formulation mathématique a été proposée en 1939 mais elle n'est parue en anglais que plus de 20 ans plus tard (Kantorovich, 1960). Il s'agit d'un programme linéaire en nombres entiers ayant un nombre polynomial de variables. Cette formulation fut rejetée à cause de la piètre qualité de la borne inférieure fournie par sa relaxation linéaire. Gilmore et Gomory (1961, 1963) formulèrent le problème en utilisant comme variables de décision, tous les patrons de découpe réalisables. Le nombre de ces derniers étant exponentiel en terme du nombre d'items différents, ils proposèrent une procédure de résolution par génération de colonnes afin de ne générer au besoin que les patrons nécessaires. Ceux-ci sont obtenus comme solutions d'un problème de sac-à-dos (*Knapsack Problem*). La méthode présente toutefois le défaut que la procédure de génération de colonnes utilisée ne garantit pas l'obtention de la solution optimale entière (voir section 3.2). Les contraintes d'intégralité sont

ainsi relaxées, et des techniques d'arrondissement ont été proposées par les auteurs pour obtenir une borne supérieure sur la valeur optimale de la solution.

Des approches heuristiques séquentielles ont également été proposées. Elles consistent à construire la solution en considérant un seul item à la fois. La première a été présentée par Haessler (1971). Parmi les heuristiques les plus connues, il y a le *Next Fit Decreasing* (NFD), le *First Fit Decreasing* (FFD), et le *Best Fit Decreasing* (BFD). Elles consistent à numéroter les items dans l'ordre décroissant des longueurs et ensuite à remplir les rouleaux suivant l'espace résiduel en utilisant pour chacune un critère différent. Les deux dernières sont meilleures que la première et présentent un rapport asymptotique $Sol.Heuristique/Sol.Exacte \leq 1.222$ (Martello et Toth 1990). D'autres procédures d'arrondissement basées sur la technique de résolution de Gilmore et Gomory furent proposées par Roodman (1985) et Stadtler (1990). Le premier arrondit la solution de la relaxation linéaire pour obtenir une solution entière éventuellement non réalisable. Il essaie ensuite d'en obtenir une solution qui satisfait toutes les demandes et l'améliore en réarrangeant la perte sur les rouleaux. Le deuxième applique une procédure d'arrondissement à la solution de la relaxation linéaire pour obtenir une solution entière réalisable. Ensuite, il combine cette solution avec la procédure FFD pour obtenir une solution meilleure que celle fournie par FFD. Enfin, Martello et Toth (1990) proposent une procédure simple pour calculer une borne inférieure dont le ratio du pire cas est $2/3$.

La plupart des approches optimales sont basées sur la technique de résolution par génération de colonnes de Gilmore et Gomory. Dyckhoff (1985) propose un modèle séquentiel à un seul couteau. Il génère tous les patrons possibles par l'intermédiaire de patrons à deux composantes appelés *one-cuts*. Cette approche est utile dans le

cas de plusieurs tailles standards pour les grands rouleaux, ou d'un grand nombre de longueurs différentes pour les petits items. Eilon et Christofides (1971) ainsi que Martello et Toth (1990) ont présenté des méthodes d'énumération en profondeur d'abord, les premiers utilisant une stratégie BFD et les deuxièmes une stratégie FFD. La méthode proposée par Goulimis (1990) consiste à générer préalablement tous les patrons réalisables pour résoudre la relaxation linéaire. La solution optimale entière est obtenue par une procédure qui combine le branchement et les plans de coupe. Bien que la génération des patrons profite des contraintes opérationnelles (nombre de couteaux, nombre d'items par rouleau, limites sur la perte par rouleau), la méthode présente l'inconvénient majeur que le nombre de patrons est souvent trop élevé et elle ne permet de résoudre que des problèmes de taille limitée.

Les approches optimales récentes utilisent toutes la méthode de génération de colonnes. Les plus intéressantes ont été proposées par Vance *et al.* (1994), Vance (1996), Vanderbeck (1996) et Valério de Carvalhò (1996). Dans Vance *et al.* (1994), les auteurs proposent la formulation de Gilmore et Gomory pour le problème de découpe binaire avec une procédure de branchement qui impose (ou interdit) l'utilisation simultanée de deux items sur un rouleau. Ce branchement réduit la taille du problème dans la branche de gauche (items utilisés simultanément sur un même rouleau), mais modifie la structure du sous-problème dans la branche de droite et le rend difficile à résoudre. Vance (1996) propose deux schémas branchements. Le premier présente le défaut de ne pas éliminer la symétrie entre les rouleaux. Dans le deuxième, la procédure de branchement utilise un ensemble d'items dont chaque élément l doit être découpé α_l fois sur un nombre minimum (ou maximum, suivant la branche de l'arbre) de patrons utilisés. Mais le sous-problème change de structure après le branchement et sa résolution devient plus difficile. Pour pallier à cette

difficulté, seul les patrons maximaux sont utilisés. Vanderbeck (1996) a proposé deux formulations, l'une basée sur la discrétisation du polyèdre du sous-problème et l'autre basée sur la convexification de ce polyèdre. Le branchement partitionne l'ensemble des colonnes, soit à l'aide d'un hyperplan, ou en imposant des bornes supérieures ou inférieures sur les composantes des colonnes. Enfin, Valério de Carvalho (1996) utilise une formulation réseau résolue par génération de colonnes. Le sous-problème est un problème de plus long chemin sur un réseau acyclique qui représente la taille d'un rouleau standard. Le problème maître ne conserve de la formulation réseau que les contraintes de couverture de la demande. Les contraintes de conservation de flot sont ajoutées au fur et à mesure que les variables qui y figurent sont générées.

Dans cette recherche, nous examinons le problème de découpe unidimensionnelle comme un problème de tournées des véhicules. Les items correspondent aux clients à visiter, les rouleaux représentent les véhicules alors que les patrons correspondent aux tournées de véhicules. Ce travail va dans la même direction que les travaux sur les tournées de véhicules effectués par Desrosiers *et al.* (1995) et Desaulniers *et al.* (1997) pour lesquels des résultats fort intéressants ont été obtenus dans des applications en transport urbain, aérien et ferroviaire. La formulation utilisée est du type multiflot et la résolution utilise la méthode de décomposition de Dantzig-Wolfe (1960) imbriquée dans une procédure de séparation et d'évaluation pour obtenir une solution optimale entière. Cette recherche fait enfin suite aux travaux de Paquette (1996) réalisés sur le problème de la découpe binaire et effectués avec la même approche de résolution.

Chapitre 3

Formulations classiques

Les formulations classiques pour le problème de découpe sont celles de Kantorovich (1960) et de Gilmore et Gomory (1961, 1963). Si la première fût rejetée pendant longtemps à cause de la piètre qualité de sa relaxation linéaire, la deuxième a été très souvent utilisée comme point de départ de nombreuses recherches. Dans la présente section, nous décrivons ces deux formulations et nous analysons les relations entre celles-ci.

Nous considérons la notation suivante: L est la longueur d'un rouleau standard, $N = \{1, \dots, n\}$ est l'ensemble des items de tailles l_1, \dots, l_n , et b_i est la demande de l'item i . L'ensemble des rouleaux de taille standard est noté par K .

3.1 Formulation de Kantorovich (1960)

On définit les variables entières X_i^k ($i \in N$, $k \in K$) qui comptent le nombre d'items i découpés sur le rouleau k , et les variables binaires X_0^k ($k \in K$) qui prennent la valeur 1 si le rouleau k est utilisé et la valeur 0 sinon. La formulation proposée par Kantorovich est la suivante:

$$\text{Minimiser } \sum_{k \in K} X_0^k \quad (3.1)$$

sujet à :

$$\sum_{k \in K} X_i^k \geq b_i, \quad \forall i \in N \quad (3.2)$$

$$\sum_{i \in N} l_i X_i^k \leq L X_0^k, \quad \forall k \in K \quad (3.3)$$

$$X_0^k \in \{0, 1\}, \quad \forall k \in K \quad (3.4)$$

$$X_i^k \in Z^+, \quad \forall k \in K, \forall i \in N. \quad (3.5)$$

Le premier ensemble de contraintes (3.2) impose la satisfaction de toutes les demandes par la solution, et le deuxième (3.3) stipule que l'ensemble des items découpés sur un rouleau ne dépasse pas la capacité du rouleau. La fonction objectif (3.1) représente la minimisation du nombre de rouleaux utilisés.

La formulation (3.1)-(3.5) comporte un nombre polynomial $|K| * (n + 1)$ de variables, et un nombre polynomial $n + |K|$ de contraintes, ce qui peut représenter un avantage pour la résolution par une méthode directe. Cependant, la relaxation linéaire fournit une borne de piètre qualité et la présence de nombreuses solutions symétriques introduit beaucoup de difficultés dans la résolution.

Proposition 1: La valeur optimale de la relaxation linéaire de (3.1)-(3.5) est $\frac{\sum_{i=1}^n l_i b_i}{L}$.

Preuve : Soit la solution

$$X_0^k = \frac{\sum_{i=1}^n l_i b_i}{L|K|}, \quad \forall k \in K$$

$$X_i^k = \frac{b_i}{|K|}, \quad \forall k \in K, i \in N.$$

Cette solution est réalisable et elle atteint la borne inférieure de l'objectif déduite des contraintes (3.2)-(3.3). Donc c'est une solution optimale et la valeur optimale de la relaxation linéaire est donnée par $\frac{\sum_{i=1}^n l_i b_i}{L}$. ■

Pour des instances avec des $l_i = \frac{L}{2} + 1$ ($\forall i \in N$), la borne fournie est $\frac{\sum_{i=1}^n b_i}{2} + \frac{\sum_{i=1}^n b_i}{L}$, alors que la valeur optimale est $\sum_{i=1}^n b_i$, i.e. chaque rouleau ne contient qu'un seul item. Le rapport est donc égal à $\frac{1}{2} + \frac{1}{L}$ qui est proche de $\frac{1}{2}$ et l'écart est donc important. Le même inconvénient persiste pour les instances ayant les demandes b_i proches de $L/2$ tout en étant plus grandes, et celles présentant une grande perte. Ceci explique le fait que cette formulation fût abandonnée, et elle n'a jamais été prise comme point de départ d'une méthode de résolution.

3.2 Formulation de Gilmore et Gomory (1961)

Afin de remédier aux inconvénients de la formulation précédente, Gilmore et Gomory (1961) ont proposé une formulation prenant en compte tous les patrons réalisables non vides. Notons par P cet ensemble, $P_0 = P \cup \{0\}$ l'ensemble de tous les patrons réalisables y compris le patron vide 0, a_{ip} ($i \in N, p \in P_0$) la fréquence de l'item i sur le patron p , et Y_p ($p \in P_0$) la fréquence de l'utilisation du patron p dans la solution. Le problème peut ainsi être formulé simplement comme suit:

$$\text{Minimiser } \sum_{p \in P} Y_p \quad (3.6)$$

sujet à :

$$\sum_{p \in P} a_{ip} Y_p \geq b_i, \quad \forall i \in N \quad (3.7)$$

$$Y_p \in Z^+, \quad \forall p \in P. \quad (3.8)$$

La relaxation linéaire de (3.6)-(3.8) donne souvent une borne très proche de la valeur optimale entière puisque les coefficients technologiques des colonnes sont déjà entiers. Pour générer les patrons, Gilmore et Gomory proposèrent la résolution du problème de sac-à-dos suivant, où les π_i ($i = 1, \dots, n$) sont les valeurs duales associées aux contraintes (3.7) et fournies par la solution de la relaxation linéaire du problème maître (3.6)-(3.8) et X_i ($i \in N$) compte le nombre de copies de l'objet i utilisées dans la solution:

$$\text{Maximiser } \sum_{i \in N} \pi_i X_i \quad (3.9)$$

sujet à :

$$\sum_{i \in N} l_i X_i \leq L \quad (3.10)$$

$$X_i \in Z^+, \quad \forall i \in N. \quad (3.11)$$

Le sous-problème correspond en fait à déterminer la colonne de coût marginal minimum, i.e.,

$$\text{Minimiser}_{p \in P} 1 - \sum_{i \in N} \pi_i a_{ip}. \quad (3.12)$$

Cependant, il faut souligner ici que ce sous-problème ne permet pas de générer tous les patrons réalisables et que par conséquent, il ne garantit pas l'obtention de la solution optimale. L'exemple suivant montre que même si on dispose de tous les

patrons susceptibles d'être générés par le sous-problème (3.9)-(3.11), on n'est pas assuré d'obtenir l'optimum entier. La figure 3.1 montre un exemple avec $L = 15$, $l_1 = 5$, $l_2 = 6$ et $b_1 = b_2 = 1$. Il est clair que le sous-problème ne générera que les patrons extrémaux: soit le patron vide $(0,0)$, soit les patrons non vides $(3,0)$ et $(0,2)$. Avec les contraintes d'intégralité sur les variables Y_0 , Y_1 et Y_2 correspondant à ces patrons, il est impossible d'obtenir le patron $(1,1)$, et par conséquent la solution optimale entière ne pourra pas être trouvée. La difficulté est que les contraintes d'intégralité imposées sur les variables de patrons (3.8) ne permettent pas de décrire tous les patrons réalisables à partir des patrons extrémaux. On notera toutefois que le patron optimal $(1,1)$ s'obtient comme combinaison convexe des patrons extrémaux, i.e. $(1,1) = \frac{1}{6}(0,0) + \frac{1}{3}(3,0) + \frac{1}{2}(0,2)$.

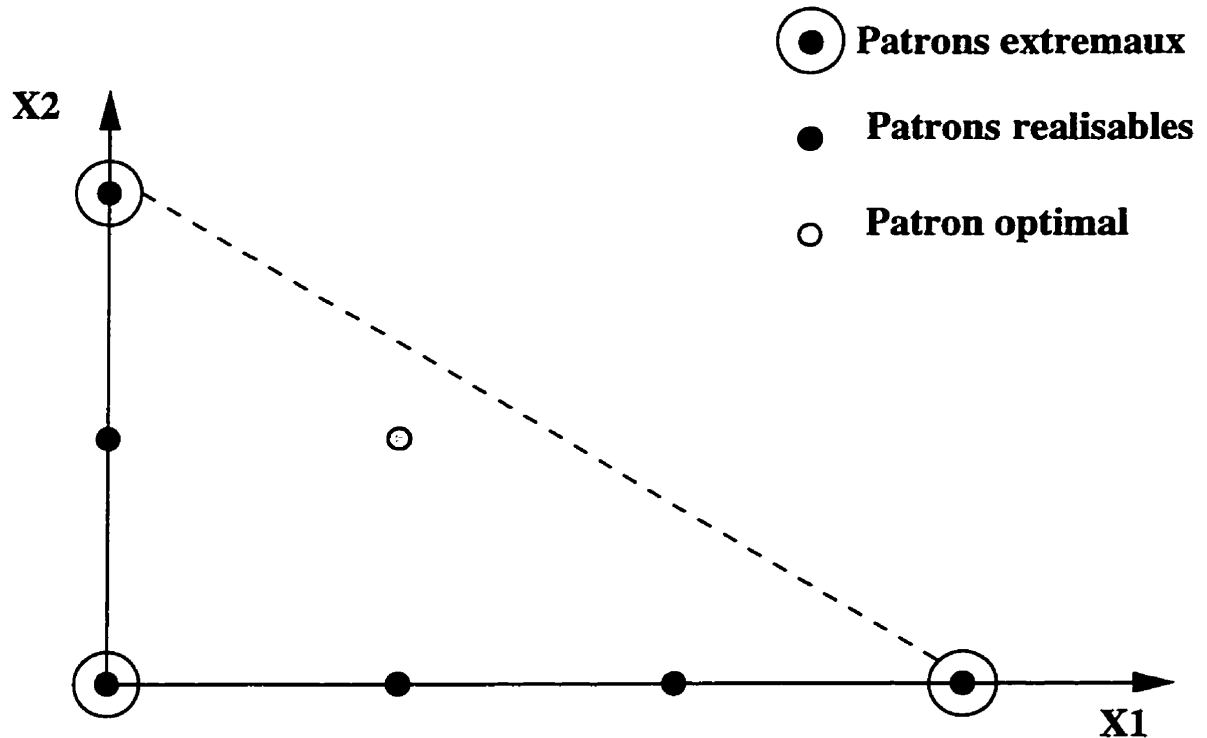


Figure 3.1: Exemple illustrant la possibilité de ne pouvoir obtenir la solution optimale avec la technique proposée par Gilmore et Gomory.

La difficulté de résolution de l'exemple précédent disparaît si on restreint le domaine du sous-problème à $0 \leq X_1, X_2 \leq 1$. Toutefois, cette modification n'est pas valide si on porte la demande à $b_1 = 4$ et $b_2 = 3$; dans ce cas, la solution optimale requiert les deux patrons extrémaux $(3, 0)$ et $(0, 2)$ ainsi qu'une fois le patron non extrémal $(1, 1)$. On peut toutefois contourner cette difficulté avec une formulation de Gilmore et Gomory modifiée obtenue en appliquant la décomposition de Dantzig-Wolfe à la formulation de Kantorovich. Cette formulation modifiée garde le même sous-problème (3.9)-(3.11), en nombres entiers, mais conserve les contraintes d'intégralité sur les variables originales X_i^k ($i \in N, k \in K$) et X_0^k ($k \in K$) qui vont s'écrire, après transformation, comme la combinaison convexe des points extrêmes du domaine du sous-problème.

3.3 Décomposition de la formulation de Kantorovich

La décomposition de Dantzig-Wolfe de la formulation (3.1)-(3.5) retient dans le problème maître l'objectif (3.1) et les contraintes (3.2). Par ailleurs, le domaine du sous-problème est composé des contraintes (3.3)-(3.5) (y compris les contraintes d'intégralité). Ce sous-problème est séparable par rouleau ($k \in K$), et qui plus est, chacun est identique. Dénотons par $\Omega_0^k = \{x_p^k = (x_{ip}^k), i \in N \cup \{0\}\}$ l'ensemble des points extrêmes de l'enveloppe convexe des contraintes (3.3)-(3.5); ces points décrivent en nombres entiers les patrons p admissibles. Par convention, on associe l'indice $p = 0$ au patron vide, i.e. pour tout $k \in K$, $(x_{00}^k, x_{10}^k, \dots, x_{n0}^k) = (0, \dots, 0)$, et pour tout patron non vide p , on a $x_{0p}^k = 1$. Tout point réalisable entier de (3.3)-(3.5)

s'écrit:

$$X_i^k = \sum_{p \in \Omega_0^k} x_{ip}^k Y_p^k, \quad \forall k \in K, \forall i \in N \cup \{0\} \quad (3.13)$$

$$\sum_{p \in \Omega_0^k} Y_p^k = 1, \quad \forall k \in K \quad (3.14)$$

$$Y_p^k \geq 0, \quad \forall k \in K, \forall p \in \Omega_0^k \quad (3.15)$$

$$X_0^k \in \{0, 1\}, \quad \forall k \in K \quad (3.16)$$

$$X_i^k \in Z^+, \quad \forall k \in K, \forall i \in N. \quad (3.17)$$

Par substitution de (3.13)-(3.17) dans (3.1)-(3.5), on obtient le problème maître:

$$\text{Minimiser } \sum_{k \in K} \sum_{p \in \Omega_0^k} x_{0p}^k Y_p^k$$

sujet à :

$$\sum_{k \in K} \sum_{p \in \Omega_0^k} x_{ip}^k Y_p^k \geq b_i, \quad \forall i \in N$$

et (3.13) – (3.17).

Puisque les ensembles Ω_0^k ($k \in K$) sont identiques, posons $\forall k \in K, \Omega_0 = \Omega_0^k$ l'ensemble des patrons réalisables, $\Omega = \Omega_0 - \{0\}$ l'ensemble des patrons non vides, et $a_{ip} = x_{ip}^k$. Définissons ensuite les variables

$$Y_p = \sum_{k \in K} Y_p^k, \quad \forall p \in \Omega_0 \quad (3.18)$$

qui sont non-négatives puisque les variables Y_p^k ($k \in K, p \in \Omega_0$) le sont également. Puisque $a_{i0} = 0$ ($i \in N$) et a_{ip} ($i \in N, p \in \Omega$) est le nombre d'items i découpés sur le patron non vide p , on obtient la formulation suivante pour le problème maître:

$$\text{Minimiser } \sum_{p \in \Omega} Y_p \quad (3.19)$$

sujet à :

$$\sum_{p \in \Omega} a_{ip} Y_p \geq b_i, \quad \forall i \in N \quad (3.20)$$

$$Y_p \geq 0, \quad \forall p \in \Omega \quad (3.21)$$

et (3.13) – (3.18).

On voit que les deux substitutions (3.13)-(3.17) et (3.18) opérées successivement sur la formulation compacte (3.1)-(3.5) gardent les contraintes d'intégralité sur les variables X_i^k ($k \in K, i \in N \cup \{0\}$), et non sur les nouvelles variables Y_p ($p \in \Omega$). En relaxant les contraintes (3.13)-(3.18) de la formulation précédente, on obtient la relaxation linéaire du problème maître de la formulation de Gilmore et Gomory. On note de plus que le sous-problème de génération des colonnes est le même, i.e. problème de sac-à-dos en nombres entiers. (Ce sous-problème admet en fait un patron supplémentaire, le patron vide.)

Une fois les variables Y_p ($p \in \Omega$) trouvées, les équations (3.14)-(3.15) et (3.18) forment un problème de transport qui admet une solution, puisque à l'optimum $\sum_{p \in \Omega} Y_p \leq |K|$, et en posant $Y_0 = |K| - \sum_{p \in \Omega} Y_p$, ce problème vérifie la condition de réalisabilité. De plus sa taille est réduite ($|\{p \in \Omega : Y_p > 0\}| * |K|$). On peut donc résoudre (3.19)-(3.21) pour déterminer Y_p ($p \in \Omega_0$), puis résoudre (3.14)-(3.15) et (3.18) pour trouver les Y_p^k ($k \in K, p \in \Omega_0$), et ensuite calculer X_i^k ($k \in K, i \in N \cup \{0\}$) par (3.13) et prendre les décisions de branchement sur ces dernières variables.

Notons que $\Omega \subset P$ et qu'on peut avoir $\Omega \neq P$. Ceci explique en partie pourquoi la résolution avec $Y_p \in Z^+$ ($p \in \Omega$) n'est pas correcte. Une manière de résoudre le problème de découpe avec la technique de génération de colonnes de Gilmore et

Gomory tout en prenant en compte les contraintes (3.13)-(3.18) est de brancher sur les variables X_i^k comme indiqué ci-dessus. Mais, la symétrie entre les rouleaux qui est difficile à éliminer dans cette formulation rend la résolution peu efficace.

On peut conclure que la formulation (3.1)-(3.5) de Kantorovich n'est pas une mauvaise formulation, mais que les deux formulations présentées sont équivalentes si on utilise pour la résolution la décomposition de Dantzig-Wolfe, avec les contraintes d'intégralité dans le sous-problème. Dans le présent travail, nous montrerons qu'à partir de cette formulation, on peut trouver une formulation équivalente qui permettra de résoudre efficacement le problème de découpe.

Chapitre 4

Formulations de type réseau

Pour un schéma de génération de colonnes donné, il existe généralement plusieurs formulations compactes qui donnent le même problème maître lorsqu'on applique la décomposition de Dantzig-Wolfe. Dans cette section, nous allons présenter deux formulations qui utilisent un réseau pour décrire les patrons. La première formulation a été proposée par Valério de Carvalhò (1996) alors que la seconde est une formulation multiflot inspirée de celles utilisées pour les problèmes de tournées de véhicules.

4.1 La formulation de Valério de Carvalhò (1996)

4.1.1 Description du réseau

Dans la formulation de Valério de Carvalhò (1996), le réseau représente la taille d'un rouleau standard. À chaque unité de longueur ($1 \leq u \leq L$) correspond un nœud, et pour le début du découpage, un nœud origine 0 est ajouté. L'ensemble des nœuds est noté $V = \{0, 1, \dots, L\}$ et l'ensemble des arcs est noté A , lequel est composé de deux sous-ensembles. Le premier sous-ensemble comporte les arcs qui permettent de découper les items, i.e., pour $u < v$, $(u, v) \in A$ s'il existe un item de longueur $v - u$. Le deuxième sous-ensemble contient les arcs de perte, i.e. les arcs

$(u, u + 1)$ pour $u = 0, 1, \dots, L - 1$. On peut donc écrire

$$A = \{(u, v) : 0 \leq u < v \leq L \text{ et } v - u = l_i, 1 \leq i \leq n\} \cup \{(u, u + 1), u = 0, 1, \dots, L - 1\}.$$

Ce réseau est acyclique et tout chemin de 0 à L décrit un patron réalisable pour le problème de découpe, et inversement tout patron réalisable peut être mis sous la forme d'un chemin (non nécessairement unique) de 0 à L . Un exemple de réseau est donné à la figure 4.1. La longueur du grand rouleau est égale à 5; deux types d'items doivent être découpés et correspondent à $l_1 = 2$ et $l_2 = 3$. Le chemin $\{(0, 2), (2, 4), (4, 5)\}$ correspond au patron $[2 * l_1]$. La perte encourue par l'utilisation de ce patron est égale à 1 et est localisée sur l'arc $(4, 5)$ du chemin correspondant.

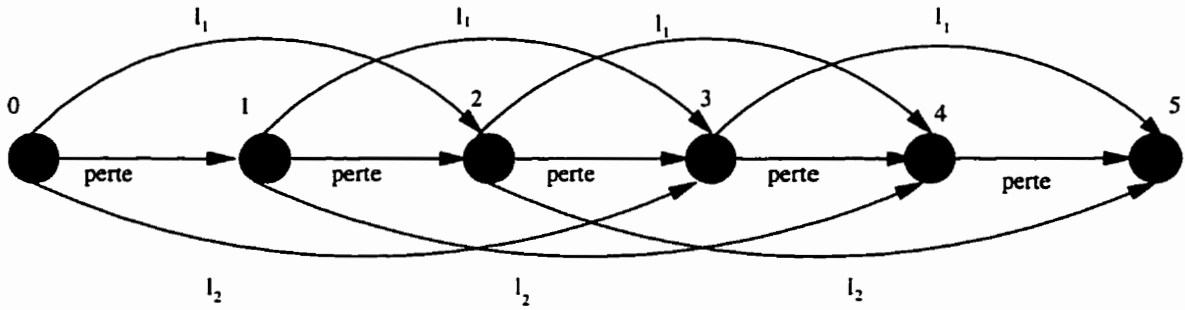


Figure 4.1: Réseau de Valério de Carvalhò (1996) avec $L = 5$, $l_1 = 2$ et $l_2 = 3$

4.1.2 Formulation mathématique

La formulation utilise les variables entières non négatives X_{uv} $((u, v) \in A)$ qui correspondent aux arcs du réseau décrit dans le paragraphe précédent. De plus une variable z est définie comme étant le nombre de rouleaux utilisés. La formulation est la suivante:

$$\text{Minimiser } z \quad (4.1)$$

sujet à :

$$\sum_{(u,u+l_i) \in A} X_{u,u+l_i} \geq b_i, \quad \forall i \in N \quad (4.2)$$

$$\sum_{(0,v) \in A} X_{0v} = z, \quad (4.3)$$

$$\sum_{(u,v) \in A} X_{uv} - \sum_{(v,u) \in A} X_{vu} = 0, \quad \forall v \in \{1, \dots, L-1\} \quad (4.4)$$

$$\sum_{(u,L) \in A} X_{uL} = z, \quad (4.5)$$

$$X_{uv} \in Z^+, \quad \forall (u,v) \in A. \quad (4.6)$$

L'objectif (4.1) minimise z qui compte le nombre de rouleaux utilisés. Les contraintes (4.2) traduisent la satisfaction de la demande pour chaque type d'item $i \in N$. Les contraintes (4.3)-(4.5) sont les contraintes de conservation de flot: elles définissent la structure d'un chemin (ou d'un patron) de 0 à L . Les contraintes d'intégralité des variables de flot sont données par (4.6). On remarque qu'il n'y a pas de contraintes sur la capacité d'un rouleau. En fait, celles-ci sont implicites et sont imposées par la structure du réseau qui représente toutes les unités de longueur d'un rouleau.

4.1.3 Décomposition de la formulation de Valério de Carvalho

En appliquant la décomposition de Dantzig-Wolfe à la formulation précédente (4.1)-(4.6), le problème maître retient l'objectif (4.1) et les contraintes (4.2) alors que le sous-problème est défini par les contraintes (4.3)-(4.6), en plus de l'objectif

visant à minimiser le coût réduit d'un chemin. Le domaine de ce sous-problème est non borné, et tout patron non vide correspond à un rayon extrémal de l'ensemble des solutions admissibles. Le seul point extrême du domaine est le patron vide, le vecteur 0 de dimension $|A|$. Ainsi, tout point admissible au sous-problème s'écrit comme combinaison linéaire non-négative des rayons extrémaux. Soit Ω l'ensemble de ces rayons extrémaux (cet ensemble ne contient pas le patron vide). Dans l'expression d'un patron non vide, la coordonnée qui correspond à la variable z est nécessairement égale à 1; par conséquent, on peut noter un rayon extrémal ($p \in \Omega$) du sous-problème par $(1; x_{uvp})$, pour $(u, v) \in A$. Dès lors, tout point $(z; X_{uv}, (u, v) \in A)$ s'écrit comme la combinaison linéaire non-négative de ces rayons extrémaux:

$$z = \sum_{p \in \Omega} Y_p \quad (4.7)$$

$$X_{uv} = \sum_{p \in \Omega} x_{uvp} Y_p \quad (4.8)$$

$$Y_p \geq 0, \quad \forall p \in \Omega \quad (4.9)$$

$$X_{uv} \in Z^+, \quad \forall (u, v) \in A. \quad (4.10)$$

En notant par a_{ip} ($i \in N, p \in \Omega$) l'expression $\sum_{(u, u+l_i) \in A} x_{u, u+l_i, p}$ qui compte la présence de l'item i sur le patron p , on obtient, après la substitution de (4.7)-(4.10) dans (4.1)-(4.4), la formulation suivante pour le problème maître :

$$\text{Minimiser } z = \sum_{p \in \Omega} Y_p \quad (4.11)$$

sujet à :

$$\sum_{p \in \Omega} a_{ip} Y_p \geq b_i, \quad \forall i \in N \quad (4.12)$$

$$Y_p \geq 0, \quad \forall p \in \Omega \quad (4.13)$$

$$X_{uv} = \sum_{p \in \Omega} x_{uvp} Y_p, \quad (4.14)$$

$$X_{uv} \in \mathbb{Z}^+, \quad \forall (u, v) \in A. \quad (4.15)$$

On voit ainsi que la relaxation linéaire (4.11)-(4.13) de cette dernière formulation est la même que celle donnée par Gilmore et Gomory (3.6)-(3.8) en utilisant comme sous-problème (4.3)-(4.6) et un objectif qui dépend des variables duales optimales du problème maître restreint.

Pour la résolution du problème de découpe, Valério de Carvalhò (1996) n'utilise pas le résultat de la décomposition. Il garde la formulation en terme des variables de flot, i.e. (4.1)-(4.6) pour le problème maître et utilise le même sous-problème pour générer ces variables de flot en utilisant le chemin le plus long de 0 à L . Les contraintes de conservation de flot (4.3)-(4.5) ne sont pas mises dans le problème maître au départ et sont générées au fur et à mesure que les variables qui y sont présentes entrent dans le problème maître restreint. Le branchement se fait sur les variables de flot une seule à la fois, ce qui revient à imposer une limite sur le nombre d'unités empruntant un arc du réseau. Les détails de l'implantation de cette approche ainsi que les résultats numériques peuvent être trouvés dans le rapport de recherche déjà cité.

4.2 Formulation multiflot

La formulation que nous allons présenter constitue un point de vue nouveau pour le problème de découpe. Elle a été inspirée de la formulation unifiée proposée par Desaulniers *et al.* (1997) pour les problèmes de tournées et d'horaires. Le problème de découpe est considéré ici comme un problème de tournées où les grands rouleaux correspondent aux véhicules dont la capacité est égale à la capacité L des rouleaux alors que les items différents sont les clients qui doivent recevoir un nombre minimum de visites. La distance entre les clients est considérée nulle. Un client peut être visité par un ou plusieurs véhicules et un véhicule peut desservir le même client plus d'une fois dans une tournée. Un patron de découpe correspond alors à la tournée d'un véhicule.

Le problème présente des particularités intéressantes qui en font un problème de tournées de véhicules plus facile et plus simple à traiter. En fait la symétrie du problème fait que l'ordre dans lequel les clients sont visités n'a pas d'impact sur la solution. Par conséquent, les items peuvent être ordonnés dans un ordre qui peut être qualifié d'ordre topologique, dans le sens où les clients seront visités dans cet ordre. Le réseau associé ne contiendra donc pas de cycles ayant plus d'un nœud (ou arc). Une autre propriété importante est que l'objectif consiste à minimiser le nombre total de véhicules utilisés et que la notion de dépôt n'a pas de signification concrète. Ainsi, la structure du coût sera plus simple et les dépôts utilisés seront fictifs, c'est-à-dire un point de début et un point de fin pour le découpage, sans attributs dans l'espace ou dans le temps.

4.2.1 Représentation des patrons par des chemins

Une tournée de véhicule correspond à un patron utilisé pour découper un rouleau. Si on dispose les items dans un ordre arbitraire, en ajoutant un nœud de départ o et un nœud d'arrivée d , un patron est un chemin de o à d . Pour découper un item plus d'une fois sur un rouleau, le nœud correspondant peut être revisité à l'aide d'une boucle. La figure 4.2 donne un exemple de patron $[3 * l_i, l_{i+1}, 2 * l_j]$ qui correspond à la tournée $[(o, i), (i, i), (i, i), (i, i+1), (i+1, j), (j, j), (j, d)]$.

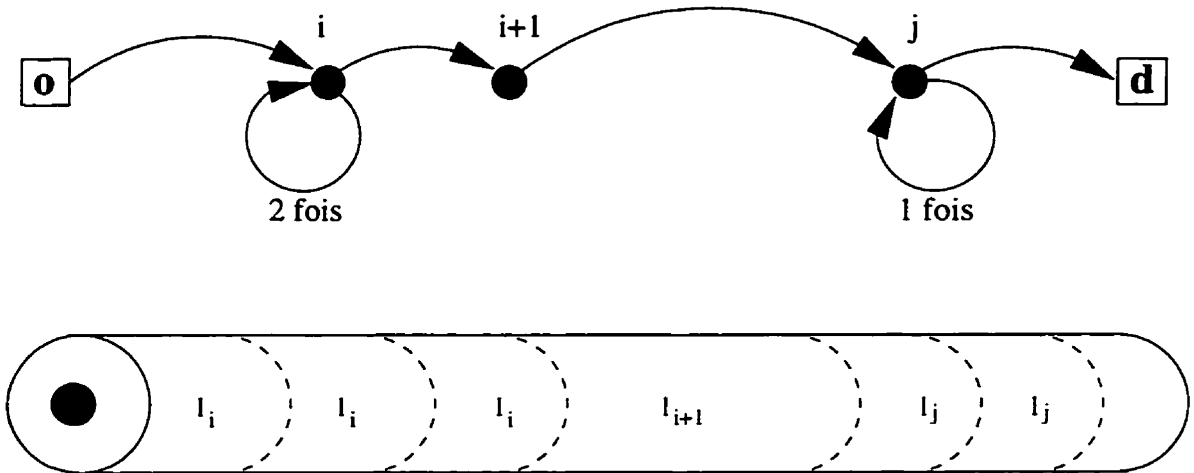


Figure 4.2: Exemple d'une représentation d'un patron par un chemin.

En reprenant l'exemple de la figure 3.1 de la section 3.2, si on représente les patrons extrémaux et le patron optimal $(1, 1)$ on obtient les chemins suivants :

- Patron Nul : $[o, d]$
- Patron $(3, 0)$: $[(o, 1), (1, 1), (1, 1), (1, d)]$
- Patron $(0, 2)$: $[(o, 2), (2, 2), (2, d)]$

- Patron* (1, 1): $[(o, 1), (1, 2), (2, d)]$.

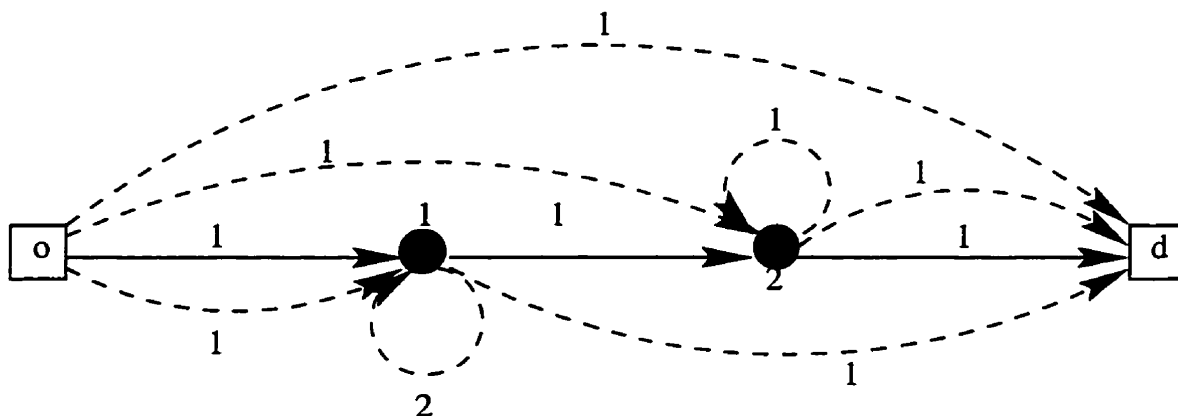


Figure 4.3: Représentation des patrons de l'exemple.

On voit bien sur la figure 4.3 qu'il n'est pas possible d'exprimer le dernier patron comme combinaison convexe des trois premiers, et par conséquent ces points extrêmes ne suffisent pas à eux seuls pour décrire tous les patrons réalisables. En effet, le patron (1, 1) est un point extrême du problème de plus court chemin avec contraintes de capacité (SPC). Ainsi, la représentation du problème du sac-à-dos (KP) par un problème de plus court chemin, quoiqu'équivalente grâce à la structure de coût particulière, fait paraître des points intérieurs du domaine de KP comme des points extrêmes du nouveau domaine SPC. Cependant, la présence des boucles dans le réseau rend la description des points extrêmes confuse et compliquée. Nous allons donc donner une représentation des patrons admissibles sur un réseau acyclique, ce qui nous permettra de caractériser les patrons qui sont des points extrêmes du sous-problème SPC.

4.2.2 Un réseau acyclique pour représenter les patrons

Le réseau que nous donnerons ci-après est obtenu à partir du précédent en remplaçant les boucles par des nœuds additionnels qui permettent de découper un item plusieurs fois. En fait, la boucle (i, i) ($i \in N$) est remplacée par un ensemble de sous-chemins dont le nombre d'arcs peut varier de 1 à p_i . La valeur p_i est le nombre maximum de duplications de l'item i sur un même rouleau; il dépend naturellement de l_i mais aussi de la demande b_i . On peut définir p_i par $p_i = \text{Min} \{ \lfloor \frac{L}{l_i} \rfloor, b_i \} - 1$. La figure 4.4 donne la représentation du nouveau réseau.

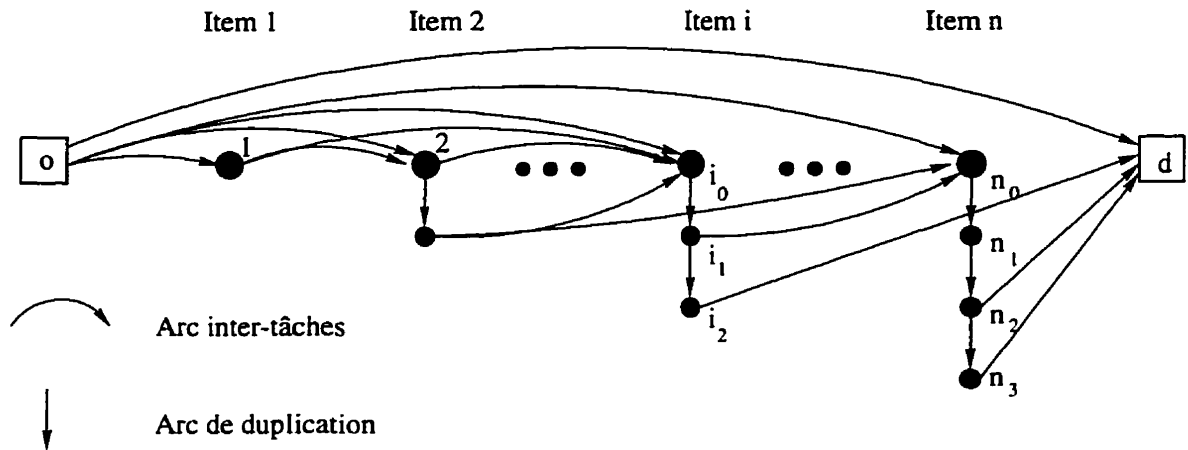


Figure 4.4: Exemple de réseau acyclique pour KP.

Les réseaux sont identiques pour tous les rouleaux et sont construits comme suit. Les items sont numérotés suivant l'ordre décroissant des tailles de 1 jusqu'à n . L'ensemble des nœuds est $N^k = \{i_v : v = 0, \dots, p_i, i \in N\} \cup \{o(k), d(k)\}$, où $o(k)$ et $d(k)$ sont les nœuds de début et de fin pour le rouleau k . Les éléments de N^k peuvent être classés suivant un ordre topologique tel que si $(i, j) \in A^k$ alors $i < j$. Les arcs de A^k vont dans un seul sens, celui de l'ordre topologique des nœuds. Il y a essentiellement deux types d'arcs: les arcs d'inter-tâches relient des items

différents et sont de la forme (i_v, j_0) ($v = 0, \dots, p_i$) et les arcs de duplication relient les nœuds i_{v-1} et i_v ($i \in N, v = 1, \dots, p_i$). Il y a un arc entre deux nœuds si la somme des longueurs correspondantes ne dépasse pas la capacité L d'un rouleau, où la longueur qui correspond au nœud i_v est $(v + 1) * l_i$. Il est important de noter que les arcs d'inter-tâches peuvent partir des nœuds i_0, \dots, i_{p_i} mais vont seulement vers les nœuds j_0 . Les arcs $(o(k), j_0)$ et $(i_0, d(k))$ correspondent respectivement au début et à la fin de la découpe sur un rouleau k . Pour tout rouleau $k \in K$, il existe un arc $(o(k), d(k))$ qui indique, s'il est utilisé, que ce rouleau k ne sera pas découpé. Tous les arcs portent des flots binaires et un patron correspond à un chemin qui permet de passer une seule unité de flot de $o(k)$ à $d(k)$.

4.2.3 Formulation mathématique

Pour $k \in K$, on définit les variables binaires X_{ij}^k ($(i, j) \in A^k$) qui prennent la valeur 1 si on utilise l'arc (i, j) et 0 sinon. La formulation du problème de découpe est alors la suivante:

$$\text{Minimiser } \sum_{k \in K} \sum_{(o(k), j) \in A^k} X_{o(k)j}^k \quad (4.16)$$

sujet à :

$$\sum_{k \in K} \left(\sum_{v=0}^{p_i} \sum_{(i_v, j) \in A^k} X_{i_v j}^k \right) \geq b_i, \quad \forall i \in N \quad (4.17)$$

$$\sum_{(o(k), j) \in A^k} X_{o(k)j}^k = 1, \quad \forall k \in K \quad (4.18)$$

$$\sum_{(i, j) \in A^k} X_{ij}^k - \sum_{(j, i) \in A^k} X_{ji}^k = 0, \quad \forall i \in N, \forall k \in K \quad (4.19)$$

$$\sum_{(i, d(k)) \in A^k} X_{id(k)}^k = 1, \quad \forall k \in K \quad (4.20)$$

$$\sum_{i \in N} l_i \left(\sum_{v=0}^{p_i} \sum_{(i_v, j) \in A^k} X_{i_v j}^k \right) \leq L \left(\sum_{(o(k), j) \in A^k} X_{o(k) j}^k \right), \quad \forall k \in K \quad (4.21)$$

$$X_{ij}^k \in \{0, 1\}, \quad \forall k \in K, \forall (i, j) \in A^k. \quad (4.22)$$

Le premier ensemble de contraintes (4.17) assure la satisfaction des demandes de tous les items: le nombre d'items de longueur l_i découpés sur tous les rouleaux de K doit être au moins égal à la demande b_i . Les contraintes (4.18)-(4.21) définissent la structure d'un chemin de $o(k)$ à $d(k)$ sur chaque rouleau k . Une des contraintes (4.18) ou (4.20) est redondante et peut être omise de la formulation. Les contraintes (4.21) portent sur la capacité de chaque rouleau. Nous remarquons que pour les rouleaux utilisés, ces contraintes sont identiques, et que pour les rouleaux non utilisés, elles imposent que toutes les variables qui y figurent soient nulles. La fonction objectif (4.16) compte le nombre de rouleaux utilisés en sommant le flot sur les arcs qui partent de la source de chaque sous-réseau vers l'ensemble des items.

Cette formulation peut être considérée comme une forme désagrégée de la formulation (3.1)-(3.5) de Kantorovich. En fait, si on définit les variables X_i^k par les équations suivantes:

$$X_i^k = \sum_{v=0}^{p_i} \sum_{(i_v, j) \in A^k} X_{i_v j}^k, \quad \forall i \in N, \forall k \in K$$

$$X_0^k = \sum_{(o(k), j) \in A^k} X_{o(k) j}^k, \quad \forall k \in K,$$

on obtient la formulation (3.1)-(3.5) avec les mêmes définitions pour les variables X_i^k ($k \in K, i \in N \cup \{0\}$).

La relaxation linéaire de la formulation (4.16)-(4.22) fournit la même borne inférieure que la formulation de Kantorovich, c'est-à-dire $\lceil \frac{\sum_{i \in N} l_i b_i}{L} \rceil$. La preuve est sensiblement la même que pour (3.1)-(3.5). Cette borne ne tient compte ni de l'intégralité

de la composition des patrons, ni de la capacité des rouleaux. En fait, elle donne la valeur de la solution si on juxtaposait tous les rouleaux de K et qu'on découpait les items comme si on le faisait sur un seul rouleau de taille $|K| \times L$. Ceci explique la médiocrité de cette borne inférieure. Par contre en vertu de la relation de cette formulation avec celle de Kantorovich, on obtient par application de la décomposition de Dantzig-Wolfe une borne inférieure égale à la borne fournie par la formulation de Gilmore-Gomory.

Ainsi, pour la résolution du problème de découpe, nous appliquerons la décomposition de Dantzig-Wolfe à la formulation multiflot pour obtenir un problème maître qui utilise des variables de patrons et un sous-problème de génération de colonnes ayant la structure d'un problème de plus court chemin avec contrainte de capacité. Ceci nous permettra de tirer profit des avantages que présente la formulation en problème de tournées pour déterminer des branchements non symétriques et efficaces.

4.2.4 Décomposition de la formulation multiflot

La formulation multiflot présente une structure diagonale par bloc (4.18)-(4.21), des contraintes de liaison (4.17) (les demandes) et une fonction objectif (4.16) séparable suivant ces blocs. Donc, elle se prête bien à la décomposition de Dantzig-Wolfe, d'autant plus qu'une procédure de génération de colonnes semble être le meilleur moyen pour résoudre le problème de découpe à cause du très grand nombre de patrons réalisables. Le problème maître aura pour variables les patrons mais ne sera résolu qu'avec un sous-ensemble de colonnes générées au besoin par le sous-problème

jusqu'à ce que les conditions d'optimalité soient vérifiées. Le problème défini par le sous-ensemble actuel des colonnes sera appelé problème maître restreint. C'est la relaxation linéaire de ce problème qui sera résolue par génération de colonnes et l'obtention d'une solution entière se fera à l'aide d'une procédure de séparation et d'évaluation progressive effectuée sur les variables de flot. Les sous-problèmes sont composés des contraintes (4.18)-(4.22), pour k fixé, et leur objectif dépend des variables duales optimales de la relaxation linéaire du problème maître restreint. Cet objectif cherche à minimiser le coût réduit des chemins allant de $o(k)$ à $d(k)$.

Afin que la section puisse être lue indépendamment, nous incluons les détails de la procédure de décomposition qui est presque identique à la décomposition Dantzig-Wolfe de la formulation de Kantorovich. Le lecteur averti peut consulter seulement les résultats (4.31)-(4.34) de la présente section.

Points extrêmes des sous-problèmes: Pour le sous-problème k ($k \in K$), on définit le réseau G^k correspondant; le domaine réalisable est donné par:

$$S^k = \{X_{ij}^k, (i, j) \in A^k : (4.18) - (4.22)\}.$$

Les contraintes (4.18)-(4.22) décrivent une structure de chemin qui sert à envoyer une unité de flot entre les nœuds $o(k)$ et $d(k)$. Alors, les points extrêmes correspondent à des chemins dans G^k et peuvent être représentés par les vecteurs suivants:

$$x_p^k = (x_{ijp}^k), k \in K, p \in \Omega_0^k, (i, j) \in A^k,$$

où Ω_0^k est l'ensemble des points extrêmes du sous-problème k , l'ensemble des rayons extrémaux étant vide car S^k est borné, et $x_{ijp}^k = 1$ si l'arc (i, j) est présent dans le

patron (chemin) p et 0 sinon. Ainsi, toute solution entière X_{ij}^k ($k \in K, (i, j) \in A^k$) peut s'écrire comme combinaison convexe des points extrêmes de Ω_0^k , i.e.,

$$X_{ij}^k = \sum_{p \in \Omega_0^k} x_{ijp}^k Y_p^k, \quad \forall k \in K, \forall (i, j) \in A^k \quad (4.23)$$

$$\sum_{p \in \Omega_0^k} Y_p^k = 1, \quad \forall k \in K \quad (4.24)$$

$$Y_p^k \geq 0, \quad \forall k \in K, \forall p \in \Omega_0^k \quad (4.25)$$

$$X_{ij}^k \in \{0, 1\}, \quad \forall k \in K, \forall (i, j) \in A^k. \quad (4.26)$$

La proposition suivante présente des résultats qui nous seront utiles pour la suite.

Proposition 2: Si S^k n'est pas réduit au seul patron nul,

- i) Tout point de S^k est un point extrême de son enveloppe convexe $\text{conv}(S^k)$.
- ii) Une solution X_{ij}^k ($k \in K, (i, j) \in A^k$) à (4.23)-(4.25) est binaire ssi dans (4.23)-(4.26), Y_p^k est binaire, $\forall k \in K, \forall p \in \Omega_0^k$.
- iii) Tout patron réalisable au problème de découpe est un point extrême du sous-problème.

Preuve:

- i) S^k est décrit par des contraintes linéaires sur des variables binaires. Donc, tout point réalisable (binaire) de S^k ne peut s'écrire comme combinaison convexe non entière d'autres points de S^k . Par conséquent, tout point de S^k est un point extrême de son enveloppe convexe $\text{conv}(S^k)$.

- ii) Les points extrêmes de $\text{conv}(S^k)$ étant tous binaires, toute combinaison convexe non entière de ces points est un point strictement intérieur de $\text{conv}(S^k)$, qui n'est donc pas binaire. Donc, si les X_{ij}^k ($k \in K, (i, j) \in A^k$) sont binaires, alors les Y_p^k ($k \in K, p \in \Omega^k$) sont binaires. La réciproque est triviale.
- iii) Tout patron réalisable peut être représenté par un élément de S^k , donc tout patron réalisable est un point extrême du sous-problème. La réciproque est aussi vraie. ■

Les variables Y_p^k ($k \in K, p \in \Omega_0^k$) sont appelées variables de chemin puisque chacune est associée à un chemin dans G^k . Il y a aussi une variable associée au chemin constitué du seul arc $(o(k), d(k))$ qui correspond à la non-utilisation du rouleau k . Nous choisissons d'associer à ce chemin l'indice 0.

Deux cas peuvent se présenter pour un sous-problème k ($k \in K$). Si

$$\sum_{(o(k), j) \in A^k} X_{o(k)j}^k = 0,$$

la seule solution admissible est $(0, \dots, 0)$ qui n'est pas intéressante pour l'optimisation du problème maître. Par contre, si $\sum_{(o(k), j) \in A^k} X_{o(k)j}^k = 1$, le sous-problème est un problème de plus court chemin avec contrainte de capacité (4.21). Nous allons nous intéresser aux seuls sous-problèmes non nuls. Ces sous-problèmes ont les mêmes paramètres et sont par conséquent identiques, i.e., le même domaine réalisable et le même objectif. Nous pouvons donc poser $\Omega = \Omega^k$ où Ω^k est l'ensemble des patrons extrémaux non vides du sous-problème k .

Problème maître: Le problème maître est constitué de la fonction objectif (4.16) et des contraintes de satisfaction des demandes (4.17). Nous allons l'écrire en terme des variables de chemin afin de pouvoir bénéficier de l'apport de la génération de colonnes dans la résolution. En utilisant les identités (4.23), on obtient les égalités suivantes:

$$\begin{aligned} \sum_{k \in K} \left(\sum_{v=0}^{p_i} \sum_{(i_v, j) \in A^k} X_{i_v j}^k \right) &= \sum_{k \in K} \sum_{p \in \Omega^k} Y_p^k \left(\sum_{v=0}^{p_i} \sum_{(i_v, j) \in A^k} x_{i_v j p}^k \right), \quad \forall i \in N \\ \sum_{k \in K} \sum_{(o(k), j) \in A^k} X_{o(k) j}^k &= \sum_{k \in K} \sum_{p \in \Omega^k} Y_p^k \left(\sum_{(o(k), j) \in A^k} x_{o(k) j p}^k \right). \end{aligned}$$

Posons $a_{ip}^k = \sum_{v=0}^{p_i} \sum_{(i_v, j) \in A^k} x_{i_v j p}^k$ ($i \in N, p \in \Omega^k$) qui est le nombre d'items i présents dans le patron (ou chemin) p , et remarquons que $\sum_{(o(k), j) \in A^k} x_{o(k) j p}^k = 0$ pour le patron vide et que $\sum_{(o(k), j) \in A^k} x_{o(k) j p}^k = 1$ pour tout patron non vide. La formulation du problème maître en terme des variables Y_p^k comporte l'objectif et les contraintes de satisfaction des demandes écrits en fonction des variables de patrons et les contraintes d'intégralité sur les variables de chemins. Ces dernières contraintes sont équivalentes aux contraintes d'intégralité sur les variables de flot grâce à la proposition 2. La formulation est donnée par (4.27)-(4.30).

$$\text{Minimiser } \sum_{k \in K} \sum_{p \in \Omega^k} Y_p^k \quad (4.27)$$

sujet à :

$$\sum_{k \in K} \sum_{p \in \Omega^k} a_{ip}^k Y_p^k \geq b_i, \quad \forall i \in N \quad (4.28)$$

$$\sum_{p \in \Omega_0^k} Y_p^k = 1, \quad \forall k \in K \quad (4.29)$$

$$Y_p^k \in \{0, 1\}, \quad \forall k \in K, \forall p \in \Omega^k. \quad (4.30)$$

Agrégation des commodités et formulation classique du problème maître :

Comme les sous-problèmes non-nuls sont identiques et admettent le même ensemble de solutions admissibles Ω_0 , on peut écrire $a_{ip} = a_{ip}^k$ ($k \in K, i \in N, p \in \Omega_0$). Définissons maintenant les variables non-négatives entières Y_p ($p \in \Omega_0$) par $Y_p = \sum_{k \in K} Y_p^k$ qui comptent le nombre de rouleaux sur lesquels on utilise le patron p . Il est clair qu'à partir d'une solution de (4.27)-(4.30), on obtient une solution entière en terme des variables agrégées. De plus, si on a une solution entière en terme des variables Y_p ($p \in \Omega_0$), on peut construire une solution de (4.27)-(4.30) en affectant chaque patron non vide de la solution à un rouleau non encore utilisé. Ceci est possible car les contraintes de convexité (4.29) impliquent que $\sum_{p \in \Omega_0} Y_p = |K|$, et donc le nombre de patrons non vides utilisés (ou de manière équivalente le nombre de rouleaux découpés donné par l'objectif (4.27)) est plus petit que $|K|$. Les rouleaux inutilisés dans une solution optimale se verront affecter le patron vide. Ainsi, on obtient la formulation suivante pour le problème maître:

$$\text{Minimiser } \sum_{p \in \Omega} Y_p \quad (4.31)$$

sujet à :

$$\sum_{p \in \Omega} a_{ip} Y_p \geq b_i, \quad \forall i \in N \quad (4.32)$$

$$Y_p \geq 0, \quad \forall p \in \Omega \quad (4.33)$$

$$Y_p \in \mathbb{Z}^+, \quad \forall p \in \Omega. \quad (4.34)$$

Résolution de la relaxation linéaire: La relaxation linéaire de cette formulation est obtenue en éliminant les contraintes d'intégralité sur les variables de chemin (4.34). Étant donné les variables duales optimales de la relaxation linéaire

du problème maître restreint, au nœud 0 de l'arbre de branchement, le coût réduit d'une colonne (ou chemin) est donné par :

$$\sum_{(o(k),j) \in A^k} X_{o(k)j}^k - \sum_{i \in N} \pi_i \left(\sum_{v=0}^{p_i} \sum_{(i_v,j) \in A^k} X_{i_v j}^k \right).$$

Or, pour les sous-problèmes non nuls, qui sont d'ailleurs identiques,

$$\sum_{(o(k),j) \in A^k} X_{o(k)j}^k = 1.$$

D'où l'objectif du sous-problème résolu :

$$\text{Minimiser } 1 - \sum_{i \in N} \pi_i \left(\sum_{v=0}^{p_i} \sum_{(i_v,j) \in A} X_{i_v j} \right). \quad (4.35)$$

Le but de la résolution est de déterminer le chemin de coût réduit minimum. Si la valeur optimale du sous-problème est non-négative, la solution actuelle de la relaxation linéaire du problème maître restreint est optimale pour la relaxation linéaire du problème maître. Dans le cas contraire, la solution peut encore être améliorée et au moins une colonne de coût réduit négatif (celle ayant le coût réduit minimum) est ajoutée pour donner un nouveau problème maître restreint qui sera optimisé.

Remarquons que dans la fonction coût du sous-problème (4.35), tous les arcs qui partent des nœuds qui correspondent au même item ont le même coût $-\pi_i$. Ainsi, le sous-problème SPC n'est dans ce cas qu'une formulation désagrégée du problème de sac-à-dos (3.9)-(3.11). Par conséquent, tous les patrons de découpe réalisables, qui sont pourtant les points extrêmes du sous-problème, ne pourront être générés. Seuls ceux qui sont des points extrêmes du domaine du problème de sac-à-dos correspondant (et éventuellement ceux situés sur une facette de ce domaine) peuvent être générés en utilisant la fonction coût (4.35). La relaxation linéaire (4.31)-(4.33) fournit donc la même borne inférieure (et la même valeur optimale continue)

que la formulation classique du problème maître avec comme sous-problème un problème de sac-à-dos.

Résolution en nombres entiers: La formulation avec les contraintes d'intégralité sur les variables de patrons a été proposée par Gilmore et Gomory (1961) mais, en utilisant KP comme sous-problème, ils ne pouvaient générer tous les patrons réalisables. Par contre tous les patrons de découpe réalisables sont des points extrêmes du problème de plus court chemin avec contrainte de capacité. Par conséquent, pour tout patron réalisable, il existe des valeurs particulières des coûts des arcs pour lesquelles ce patron est optimal. Il est donc possible de générer n'importe quel patron réalisable en modifiant seulement la fonction coût du sous-problème seulement. Donc, un schéma de branchement possible consistera à considérer les décisions de branchement comme des contraintes dans le problème maître. Les coûts des arcs dans l'objectif du sous-problème seront modifiés en fonction des variables duales associées à ces contraintes. Ainsi, il est possible de générer tous les patrons réalisables nécessaires à l'obtention d'une solution optimale entière tout en gardant le même domaine réalisable pour le sous-problème tout le long du processus de résolution.

Le schéma de branchement que nous proposons consiste à ajouter des contraintes au problème maître. Ainsi, durant tout le processus de résolution, la structure du sous-problème sera gardée la même, c'est-à-dire SPC. Le branchement se fera sur des variables agrégées. Mais, dans certains cas il sera nécessaire de désagréger des contraintes de demande afin d'accéder à l'information détaillée sur chaque copie d'item (ou bien sur chaque rouleau k). Ainsi, on profite de la facilité de la résolution

du cas binaire pour lequel chaque contrainte de demande correspond à un seul item de demande unitaire. Dans ce dernier cas, les décisions seront portées directement dans le réseau du sous-problème. Une étude plus détaillée du schéma de branchement pour le cas binaire est fournie dans le chapitre 5. Le chapitre 6 traite du cas général et discute plus profondément le schéma de branchement correspondant.

4.2.5 Procédure générale de résolution

La relaxation linéaire (4.27)-(4.29) du problème maître est résolue par la technique de génération de colonnes. Les colonnes sont générées en résolvant le sous-problème et le programme linéaire qui correspond au problème maître restreint peut être résolu par un algorithme de programmation linéaire. Le sous-problème est résolu comme un problème de plus court chemin avec contraintes de ressources (Desrochers et Soumis, 1988).

Dans la procédure de séparation et d'évaluation progressive, les décisions de branchement sont prises sur les variables de flots puisque les contraintes d'intégralité portent aussi sur ces dernières. Pour pallier aux inconvénients causés par la symétrie entre les rouleaux, nous utilisons les variables agrégées $X_{ij} = \sum_{k \in K} X_{ij}^k$ ($(i, j) \in A$) qui peuvent être plus grandes que 1 dans le cas général. Les variables choisies sont celles dont la partie fractionnaire est la plus proche de 0.5. Ceci permet d'avoir un arbre de branchement équilibré, d'autant plus que les variables ayant des parties fractionnaires petites devraient s'ajuster par elles-mêmes. Selon le type du problème de découpe, les décisions de branchement seront transférées directement dans le sous-problème (cas des demandes unitaires) ou bien considérées comme des contraintes

explicités dans le problème maître (cas général). Pour cette dernière situation, leur effet sur le sous-problème se traduira par une modification du coût des arcs concernés. Les détails de la résolution et la justification des schémas de branchement sont donnés dans les deux chapitres qui suivent.

Chapitre 5

Résultats expérimentaux: cas binaire

Dans ce chapitre, nous présentons les résultats numériques obtenus sur des problèmes de découpe binaire. Chaque item est considéré individuellement et correspond à une commande à part, mais plusieurs items peuvent avoir la même longueur. Nous commençons par donner la formulation du problème maître. Puis, nous présentons les détails du schéma de branchement utilisé. Ensuite, les caractéristiques de l'implémentation ainsi que les résultats préliminaires sont présentés. Enfin, plusieurs techniques d'amélioration sont décrites dont quelques unes sont testées. Les résultats obtenus sont alors présentés et commentés.

Le problème de découpe binaire (BCSP : *Binary Cutting Stock Problem*) a la particularité que toutes les demandes sont unitaires. La formulation classique est la même que celle de Kantorovich (3.1)-(3.5), sauf que les b_i sont tous égaux à 1 et les contraintes de satisfaction des demandes sont des égalités. Les mêmes caractéristiques sont présentes dans la formulation multiflot du problème. Ainsi, un patron non vide ne pourra pas être découpé plus d'une fois, et les variables de patrons (non vides) seront restreintes à être binaires. La formulation du problème maître qui devient un problème de partitionnement est donnée par (5.1)-(5.3).

$$\text{Minimiser } \sum_{p \in \Omega} Y_p \quad (5.1)$$

sujet à :

$$\sum_{p \in \Omega} a_{ip} Y_p = 1, \quad \forall i \in N, \quad (5.2)$$

$$Y_p \in \{0, 1\}, \quad \forall p \in \Omega. \quad (5.3)$$

Le réseau devient plus simple et ne contient pas de duplications des nœuds ou des arcs. Un exemple d'un tel réseau est donné par la figure 5.1.

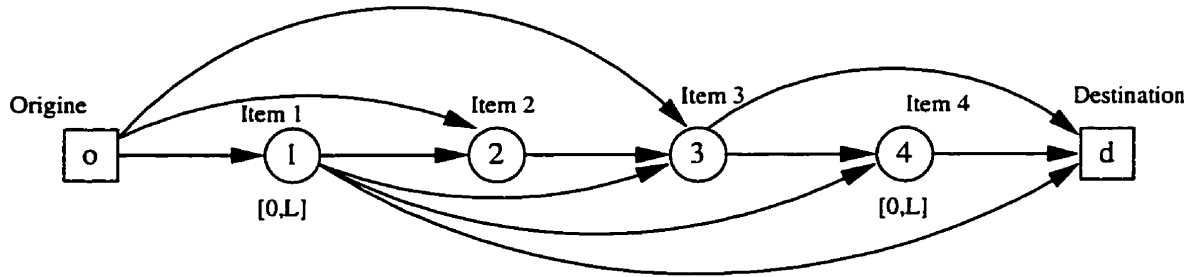


Figure 5.1: Exemple de réseau d'un problème de découpe binaire.

5.1 Branchement

Dans la procédure de séparation et d'évaluation progressive, les décisions de branchement sont prises sur les variables de flot, vu les difficultés énormes qui pourraient être causées par un branchement direct sur les variables Y_p . Pour pallier aux inconvénients causés par la symétrie entre les rouleaux, nous utilisons les variables agrégées $X_{ij} = \sum_{k \in K} X_{ij}^k$, pour $(i, j) \in A$, qui déterminent si oui ou non les items i et j sont découpés l'un après l'autre sur le même rouleau.

Le théorème suivant justifie l'utilisation de ce branchement :

Théorème 1 : Soit Y_p ($p \in \Omega$) une solution optimale à un nœud de l'arbre de branchement. Alors

- i) $X_{ij} = \sum_{p \in \Omega} x_{ijp} Y_p, \quad \forall (i, j) \in A$
- ii) $\exists p \in \Omega$ tel que Y_p est fractionnaire ($0 < Y_p < 1$) ssi $\exists (i, j) \in A$ tel que X_{ij} est fractionnaire ($0 < X_{ij} < 1$).

Preuve :

- i) $X_{ij} = \sum_{k \in K} X_{ij}^k = \sum_{k \in K} \sum_{p \in \Omega} x_{ijp} Y_p^k = \sum_{p \in \Omega} x_{ijp} Y_p$.
- ii) Supposons qu'à un nœud de l'arbre de branchement on ait obtenu la solution de la relaxation linéaire Y_p ($p \in \Omega$). Si la solution est entière, Y_p est une solution optimale entière au nœud actuel et en vertu de i), $X_{ij} \in \{0, 1\}, \forall (i, j) \in A$. Inversement, supposons qu'il existe une variable Y_p fractionnaire et que X_{ij} est binaire pour tout $(i, j) \in A$. Soit Ω^* l'ensemble des variables Y_p fractionnaires qui commencent par le plus petit nœud i^* (dans l'ordre topologique préalablement établi). Par définition, le flot qui entre au nœud i^* par les arcs autres que (o, i^*) est entier. Et comme ce nœud doit être couvert exactement une fois, alors $X_{oi^*} = 1 = \sum_{p \in \Omega^*} Y_p$. Si tous les flots sont binaires, l'unité de flot doit sortir du nœud i^* par un seul arc. Et ainsi, les patrons de Ω^* visiteront la même séquence de nœuds, et donc correspondent à un seul patron (i.e. $\Omega^* = \{p^*\}$ et $Y_{p^*} = 1$). Ce qui contredit le fait que Y_{p^*} est fractionnaire. On

en déduit que s'il existe une variable de patron fractionnaire, alors il existe une variable de flot agrégée fractionnaire dans le réseau sur laquelle on pourra brancher. ■

De plus, le réseau étant acyclique, le nombre d'arcs (i, j) est au plus $\frac{(n+1)(n+2)}{2}$. On est donc assuré de terminer la procédure de branchement en un nombre fini d'étapes. Ce branchement est d'autant plus utile que dans la plupart des colonnes de la solution optimale d'un BCSP le nombre d'items est réduit, et donc une suite de bonnes décisions permettra de trouver la solution optimale entière rapidement.

La structure du sous-problème (plus court chemin avec contrainte de capacité) permet d'imposer les décisions de branchement directement dans le réseau du sous-problème. Une décision $X_{ij} = 0$ se traduit par l'omission de l'arc (i, j) du réseau. Par conséquent, cet arc ne sera présent dans aucune colonne générée par le sous-problème. Une décision $X_{ij} = 1$ se traduit par l'omission de tous les arcs qui partent de i et de tous ceux qui arrivent à j , à l'exception de l'arc (i, j) . Ainsi, le seul moyen d'atteindre le nœud j est l'arc (i, j) et la décision est respectée par toutes les colonnes générées par le sous-problème. Ceci est illustré par la figure 5.2. Bien sûr, un pré-traitement est nécessaire au début de chaque nœud de branchement pour ne conserver dans le problème maître que les colonnes encore valides.

Nous examinons maintenant diverses caractéristiques de l'implantation que nous avons faite pour résoudre le problème.

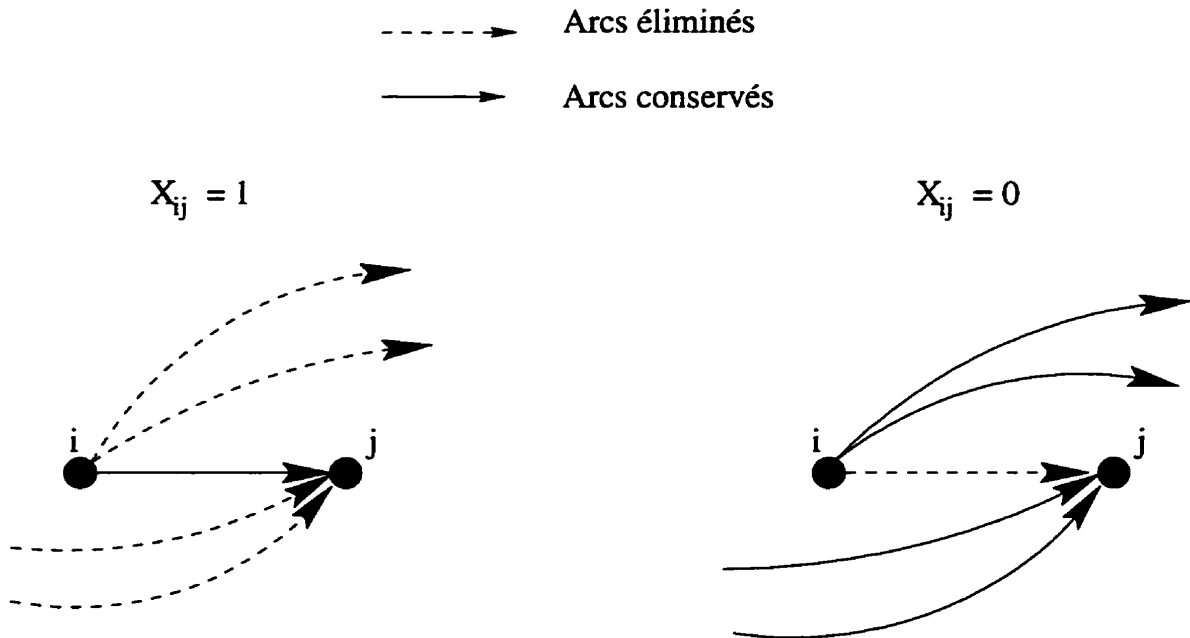


Figure 5.2: Effet des décisions de branchement sur le réseau de génération des patrons.

5.2 Caractéristiques de l'implémentation

Pré-traitement Nous utilisons le résultat suivant que l'on peut trouver dans Martello et Toth (1990) pour les paires d'items.

Proposition 3: Si deux items i et j vérifient $l_i + l_j = L$, alors il existe une solution optimale dans laquelle le patron (l_i, l_j) est utilisé.

Preuve: Supposons qu'on ait une solution optimale où i et j sont découpés sur des patrons différents $p_i = (l_i, R_i)$ et $p_j = (R_j, l_j)$ où R_i et R_j sont les parties restantes de chacun des patrons. Comme $l_i + l_j = L$, $R_j + R_i \leq (L - l_j) + (L - l_i) = L$. Il est donc possible d'échanger l_j et R_i pour obtenir les patrons (l_i, l_j) et (R_j, R_i)

sans altérer la valeur de la solution.■

Les colonnes ainsi trouvées sont fixées à 1 et les items correspondants sont éliminés du problème. Ceci permet de réduire le nombre de contraintes dans le problème maître et le nombre de colonnes réalisables.

Initialisation La procédure de génération de colonnes est initialisée par une matrice identité utilisant ainsi un rouleau par item. Nous avons aussi fourni les patrons issus de l'heuristique *Best-Fit Decreasing* (BFD). Cette procédure BFD consiste à affecter chaque item à un rouleau dans l'ordre décroissant de leur longueur. L'item est assigné au rouleau présentant le plus petit espace résiduel permettant ce placement. Si aucun rouleau ne peut prendre l'item, il est placé sur un nouveau rouleau. Cette heuristique tourne en $O(n \log(n))$ et donne au pire des cas une solution égale à 1.5 fois la valeur de la solution optimale entière (Simchi-Levi, 1994).

Sous-problème Le sous-problème est résolu en utilisant un algorithme de programmation dynamique pour les problèmes de plus court chemin avec contraintes de ressources (Desrochers et Soumis, 1988). La contrainte (4.21) du sous-problème est remplacée par les contraintes suivantes

$$\begin{aligned} X_{ij}^k (L_i^k + l_j - L_j^k) &\leq 0, & \forall k \in K, (i, j) \in A^k \\ L_0^k &= 0, & \forall k \in K \\ l_i &\leq L_i^k \leq L, & \forall k \in K, \forall i \in N, \end{aligned}$$

où les variables L_i^k cumulent la longueur du chemin partant de $o(k)$ et arrivant au nœud i . Cela permet d'utiliser un graphe d'états qui contient les informations sur

le coût et la longueur des chemins.

5.3 Description des problèmes résolus

Nous utilisons deux classes de problèmes pour tester la procédure de résolution proposée pour le problème de découpe binaire. Ces problèmes sont générés aléatoirement suivant deux schémas différents et certains sont disponibles dans la librairie OR sur le WEB. Pour chaque problème, nous résolvons 20 instances et nous considérons la performance de la méthode en moyenne sur toutes ces instances.

Problèmes générés uniformément Ces problèmes sont générés uniformément dans un intervalle donné. Il y a vingt-sept problèmes dans cette classe qui sont définis en fonction de trois paramètres : la longueur d'un grand rouleau, le nombre d'items générés et la borne inférieure sur la taille des items. La borne supérieure sur la taille des items générés est égale à 100 pour tous les problèmes. Pour chacun de ces problèmes, nous utilisons la notation $n/L/l_{min}/l_{max}$ qui veut dire que n items sont générés uniformément dans l'intervalle $[l_{min}, l_{max}]$ et que la longueur standard des rouleaux est L . Les instances du problème 500/150/20/100 sont disponibles dans la librairie OR accessible via le WEB.

Triplets Les problèmes de triplets sont connus pour être très difficiles à résoudre. Leur nom provient du fait que la solution optimale d'un tel problème comporte des rouleaux formés exactement de 3 items. La source principale de difficulté est que les tailles des items sont proches et qu'on ne peut pas distinguer clairement des gros

items et des petits items. Chaque rouleau contient un item généré aléatoirement suivant la loi $U(380, 490)$. Si on note par S l'espace restant, le deuxième item est généré suivant la loi $U(250, \frac{S}{2})$ et l'espace qui reste définit le troisième item. La longueur des grands rouleaux est fixée à 1000 et le nombre d'items à générer prend 4 valeurs possibles : 60, 120, 249 et 501. Ce qui définit 4 problèmes. Toutes les instances de ces problèmes sont disponibles dans la librairie OR.

Le tableau 5.1 contient les caractéristiques ainsi que les résultats du pré-traitement et de l'heuristique appliqués à chacun des problèmes. La colonne *nom* contient les noms des problèmes; l_{min} , l_{max} et L donnent respectivement, les longueurs minimale et maximale des items et la taille des grands rouleaux. Les colonnes *Prep* et *Gap_h* donnent le pourcentage d'items moyens éliminés après le pré-traitement et la différence moyenne entre la valeur de la solution fournie par BFD et la valeur de la solution optimale entière.

La procédure de pré-traitement est particulièrement efficace sur les problèmes uniformes. Elle permet d'éliminer jusqu'à 79% des items, et jusqu'à 48% pour les problèmes uniformes les plus difficiles (u_9 et u_{18}). Dans le cas des triplets, aucun item n'est éliminé à cause de la structure particulière des problèmes. Il faut noter néanmoins que cette procédure n'affecte pas le comportement de la méthode de résolution car les paires seraient de toute façon générées par l'heuristique BFD.

La solution heuristique obtenue est souvent très bonne pour les problèmes uniformes. Pour le plus gros problème ayant $l_{min} = 20$ (u_{18}), le saut moyen est de 2.7. Dans le cas des problèmes avec $l_{min} = 50$, la solution heuristique est toujours optimale; finalement pour les problèmes ayant $l_{min} = 1$ (u_1 à u_9) elle est optimale pour au moins 14 instances et le saut, s'il existe, est toujours de 1. Pour les triplets, la

qualité de la solution est nettement moindre et elle se dégrade considérablement avec la taille des problèmes.

L'optimiseur GENCOL4.1 fournit tous les outils nécessaires à l'implémentation de notre méthode. Le problème maître est résolu par CPLEX4.0, et la méthode de branchement proposée et l'algorithme de programmation dynamique pour la résolution du sous-problème y sont implantés. Nous utilisons la stratégie de recherche en profondeur d'abord *Depth First*. Ceci permet d'obtenir une solution réalisable rapidement, et dans le cas du BCSP ces solutions sont généralement optimales.

5.4 Résultats préliminaires

Nous présentons les résultats obtenus sur les problèmes u_1 à u_{27} , t_{60} , t_{120} et t_{249} . Vu le temps de calcul relativement élevé pour t_{249} , nous n'avons pas testé les instances de t_{501} .

Les résultats sont présentés au tableau 5.2. Nous donnons dans ce qui suit les notations qui y sont utilisées ainsi que celles qui seront utilisées dans le reste du mémoire. Tous les temps de calculs sont en secondes.

- t_{tot} temps de calcul total pour l'instance résolue
- \bar{t}_{tot} temps de calcul total moyen obtenu sur les 20 instances résolues
- t_{tot}^{min} temps de calcul total minimum obtenu sur les 20 instances résolues
- t_{tot}^{max} temps de calcul total maximum obtenu sur les 20 instances résolues

Tableau 5.1: Description des problèmes de tests: uniformes (u) et triplets (t)

<i>nom</i>	<i>n</i>	<i>min</i>	<i>max</i>	<i>L</i>	<i>Prep</i> (%)	<i>Gap_h</i>
u1	100	1	100	100	44.0	0.00
u2	200	1	100	100	40.5	0.15
u3	500	1	100	100	74.0	0.30
u4	100	1	100	120	37.0	0.20
u5	200	1	100	120	49.0	0.15
u6	500	1	100	120	60.0	0.15
u7	100	1	100	150	24.0	0.00
u8	200	1	100	150	30.0	0.15
u9	500	1	100	150	37.0	0.15
u10	100	20	100	100	39.0	0.05
u11	200	20	100	100	49.5	0.10
u12	500	20	100	100	59.0	0.10
u13	100	20	100	120	50.0	0.05
u14	200	20	100	120	63.0	0.15
u15	500	20	100	120	76.0	0.20
u16	100	20	100	150	33.0	0.75
u17	200	20	100	150	41.0	1.45
u18	500	20	100	150	48.0	2.70
u19	100	50	100	100	1.0	0.00
u20	200	50	100	100	1.5	0.00
u21	500	50	100	100	2.0	0.00
u22	100	50	100	120	25.0	0.00
u23	200	50	100	120	30.0	0.00
u24	500	50	100	120	35.0	0.00
u25	100	50	100	150	58.0	0.00
u26	200	50	100	150	70.0	0.00
u27	500	50	100	150	79.0	0.00
t60	60	250	490	1000	0.0	3.20
t120	120	250	490	1000	0.0	5.00
t249	249	250	490	1000	0.0	12.00
t501	501	250	490	1000	0.0	24.00

- \bar{t}_{mp} temps de calcul moyen consommé par la résolution du problème maître
- \bar{t}_{sp} temps de calcul moyen consommé par la résolution du sous-problème
- t_{lr} temps de calcul consommé par la résolution de la relaxation linéaire pour l'instance résolue
- \bar{t}_{lr} temps de calcul moyen consommé par la résolution de la relaxation linéaire obtenu sur les 20 instances résolues
- t_{mp}^{lr} temps de calcul consommé par la résolution du problème maître au nœud 0, pour l'instance résolue
- t_{bb} temps de calcul consommé par le branchement pour l'instance résolue
- n_{lr} nombre d'itérations de génération de colonnes lors de la résolution de la relaxation linéaire pour l'instance résolue
- n_{tot} nombre total d'itérations de génération de colonnes pour l'instance résolue
- \bar{n}_{tot} nombre total moyen d'itérations de génération de colonnes calculé sur les 20 instances résolues
- n_{bb} nombre de nœuds résolus dans l'arbre de branchement pour l'instance résolue
- i_{bb} nombre d'instances pour lesquelles l'heuristique ne donne pas la solution optimale entière
- \bar{n}_{bb} nombre moyen de nœuds résolus dans l'arbre de branchement obtenu sur les i_{bb} instances

- n_{bb}^{min} nombre minimum de nœuds résolus dans l'arbre de branchement obtenu sur les i_{bb} instances
- n_{bb}^{max} nombre maximum de nœuds résolus dans l'arbre de branchement obtenu sur les i_{bb} instances

Tableau 5.2: Résultats préliminaires (version standard)

nom	cpu(s)						Branchement				
	\bar{t}_{tot}	t_{tot}^{min}	t_{tot}^{max}	\bar{t}_{mp}	\bar{t}_{sp}	\bar{t}_{lr}	\bar{n}_{tot}	\bar{n}_{bb}	n_{bb}^{min}	n_{bb}^{max}	i_{bb}
u1	0.4	0.1	1.4	0.2	0.1	0.4	13.0	0.0	0	0	0
u2	0.9	0.2	2.9	0.4	0.2	0.8	15.0	5.0	3	6	3
u3	3.5	0.5	12.1	1.5	0.8	2.6	20.0	11.0	3	16	6
u4	1.8	0.3	4.3	0.8	0.5	1.6	21.0	10.0	8	14	4
u5	8.1	1.3	19.1	3.1	3.0	7.3	26.0	25.0	17	29	3
u6	120.5	7.5	225.8	50.36	59.4	42.0	59.0	59.0	44	81	3
u7	5.6	3.5	8.5	1.9	2.1	5.4	22.0	0.0	0	0	0
u8	34.2	18.1	53.2	11.7	33.7	31.1	35.0	43.0	36	51	3
u9	494.6	278.7	739.3	216.7	249.4	445.7	72.9	178.0	165	185	3
u10	0.2	0.1	0.6	0.1	0.1	0.2	10.0	3.0	3	3	1
u11	0.4	0.2	1.1	0.2	0.1	0.3	13.0	5.5	2	9	2
u12	1.4	0.4	3.2	0.4	0.4	1.1	23.0	10.0	8	12	2
u13	0.3	0.1	0.7	0.1	0.1	0.3	11.0	5.0	5	5	1
u14	0.7	0.2	2.4	0.2	0.3	0.6	17.0	8.0	5	12	3
u15	2.1	0.1	11.1	0.7	1.0	1.6	19.0	16.0	8	25	3
u16	3.2	1.8	5.7	1.5	0.9	2.4	33.0	17.0	10	25	15
u17	16.6	6.3	28.3	7.5	5.8	11.5	68.0	35.0	9	66	19
u18	182.6	124.0	254.6	81.4	88.7	130.9	154.6	98.0	68	130	20
u19	0.1	0.1	0.1	0.1	0.0	0.1	3.0	0.0	0	0	0
u20	0.2	0.2	0.2	0.1	0.0	0.2	3.0	0.0	0	0	0
u21	0.7	0.6	0.9	0.4	0.1	0.6	3.0	0.0	0	0	0
u22	0.1	0.1	0.2	0.1	0.0	0.1	5.0	0.0	0	0	0
u23	0.3	0.2	0.5	0.1	0.0	0.2	7.0	0.0	0	0	0
u24	0.6	0.4	1.4	0.2	0.1	0.5	10.0	0.0	0	0	0
u25	0.2	0.1	0.4	0.1	0.0	0.1	11.0	0.0	0	0	0
u26	0.3	0.1	0.7	0.2	0.1	0.2	13.0	0.0	0	0	0
u27	1.2	0.3	2.4	0.4	0.6	1.0	22.0	0.0	0	0	0
t60	3.9	3.3	4.5	1.3	2.3	3.6	51.8	0.6	0	3	20
t120	24.3	21.2	27.1	7.7	14.5	18.5	75.0	10.5	3	19	20
t249	311.0	255.0	391.7	75.9	221.0	223.2	217.3	64.2	29	101	20

La plupart des problèmes sont très faciles à résoudre. La difficulté des problèmes uniformes augmente avec l'augmentation de L , et cette difficulté est intensifiée par

l'augmentation de n . La difficulté principale des problèmes u_6 , u_8 et u_9 réside dans la résolution de la relaxation linéaire, puisque pour 17 instances sur 20, l'heuristique fournit la solution optimale entière. La résolution du sous-problème par programmation dynamique est beaucoup moins efficace que la résolution du problème de sac-à-dos binaire par un algorithme de séparation et évaluation progressive à cause des coûts et des coefficients positifs. De plus, le sous-problème a tendance à générer au départ des colonnes contenant beaucoup de petits items. Ce qui fait que la décroissance de l'objectif est très lente, le nombre d'itérations est élevé et la fin du processus de convergence est très coûteuse en temps de calcul. Le nombre de nœuds de branchement est élevé pour les 3 autres instances malgré un saut petit ($= 1$). Ceci est dû au fait que les décisions de branchement ne tiennent pas compte des poids des items. Par exemple, une décision $X_{ij} = 1$ avec $l_i = 100$ et $l_j = 1$ ne conduira pas à une bonne solution, mais produira un arbre profond avec beaucoup de retours arrières.

Le nombre de petits items et leur taille minimale ($l_{min} = 20$) et le plus grand nombre d'items éliminés par le pré-traitement font que la relaxation linéaire est plus facile à résoudre pour les problèmes u_{17} et u_{18} . Par contre, la qualité de la solution heuristique est moins bonne. Ce qui fait que le branchement est nécessaire pour presque toutes les instances.

Les mêmes difficultés existent pour les triplets, mais on remarque que le nombre d'itérations pour l'obtention de la solution de la relaxation linéaire est très élevé, principalement parce que les colonnes fournies par le sous-problème ne sont pas forcément les meilleures. Ceci illustre le fait que le critère de Dantzig (pour choisir la variable d'entrée dans l'algorithme du simplexe) n'est pas toujours efficace.

Le tableau 5.3 compare nos résultats avec ceux de Vance *et al.* (1994). Les problèmes testés sont tous du même type que les problèmes uniformes que nous utilisons. Le problème u_{18} est pris, tel que nous l'avons fait, dans la librairie OR. Nous avons remarqué que les deux méthodes présentent le même comportement par rapport aux types de problèmes. Mais, en dépit de la différence entre les technologies utilisées (IBM RS600/550 avec MINTO3.1 et CPLEX2.0, initialisation par FFD), notre méthode se comporte nettement mieux. Le temps de calcul, le nombre d'itérations de génération de colonnes et le nombre de nœuds de branchement sont généralement meilleurs.

Tableau 5.3: Comparaison avec les résultats de Vance *et al.* (1994)

		$L = 100$			$L = 120$			$L = 150$		
l_{min}	n	\bar{n}_{bb}	\bar{n}_{tot}	\bar{t}_{tot}	\bar{n}_{bb}	\bar{n}_{tot}	\bar{t}_{tot}	\bar{n}_{bb}	\bar{n}_{tot}	\bar{t}_{tot}
1	100	0.0	13.0	0.4	10.0	21.0	1.8	0.0	22	5.6
	Vance	2.2	149.5	8.4	5.1	290.2	26.5	0.0	258.1	28.3
1	200	5.0	15.0	0.9	25.0	26.0	8.1	43.0	35.0	34.2
	Vance	3.3	372.7	101.7	20.1	871.6	405.4	23.2	729.3	315.0
1	500	11.0	20.0	3.5	59.0	59.0	120.5	178.0	72.9	494.6
	Vance	24.4	1425.0	4741.0	non résolu			non résolu		
20	100	3.0	10.0	0.2	5.0	11.0	0.3	17.0	33.0	3.2
	Vance	0.2	97.7	2.9	1.9	142.8	5.9	37.0	364.0	21.4
20	200	5.5	13.0	0.4	8.0	17.0	0.7	35.0	68.0	16.6
	Vance	2.8	220.2	21.2	5.8	322.7	58.3	112.2	918.2	199.4
20	500	10.0	23.0	1.4	16.0	19.0	2.1	98.0	154.6	182.6
	Vance	7.0	642.9	251.7	35.2	968.5	883.8	306.2	2344.8	3307.6
50	100	0.0	3.0	0.1	0.0	5.0	0.1	0.0	11.0	0.2
	Vance	0.0	2.4	0.0	0.0	34.4	0.4	0.0	84.3	2.6
50	200	0.0	3.0	0.2	0.0	7.0	0.3	0.0	13.0	0.3
	Vance	0.0	5.0	0.0	0.0	68.2	2.3	0.0	173.5	10.1
50	500	0.0	3.0	0.7	0.0	10.0	0.6	0.0	22.0	1.2
	Vance	0.0	8.5	0.3	0.0	195.1	14.4	0.0	452.6	50.3

Par contre nos résultats se comparent défavorablement à ceux présentés par Valério de Carvalhò (1996) et Vanderbeck (1996) pour les gros problèmes. Nous allons donc présenter quelques techniques d'améliorations pour notre méthode.

5.5 Améliorations

Nous choisissons 2 instances de chacun des problèmes u_9 , u_{18} et t_{249} sur lesquelles nous effectuerons les prochains tests. $u9i4$ et $u9i15$ sont les instances retenues pour le problème u_9 , $u18i8$ et $u18i17$ celles retenues pour u_{18} et $t249i3$ et $t249i14$ celles retenues pour t_{249} . Le tableau 5.5 montre les différentes caractéristiques de l'optimisation de ces instances par notre méthode.

Tableau 5.4: Caractéristiques des instances choisies

Instances	t_{tot}	t_{lr}	t_{bb}	t_{sp}	t_{mp}	n_{bb}	n_{tot}
u9i4	739.3	445.0	294.3	378.1	361.3	184	275
u9i15	704.6	367.5	337.1	338.0	366.0	165	244
u18i8	203.6	149.3	54.3	92.8	91.9	97	148
u18i17	230.1	174.7	55.4	117.6	91.4	119	187
t249i3	323.7	194.9	128.8	225.4	81.6	58	196
t249i14	332.2	180.2	152.0	219.3	96.6	62	218

5.5.1 Branchement

Choix de la variable de branchement Nous avons testé divers critères pour le choix de la variable de branchement X_{ij} . Le tableau 5.5 donne les résultats obtenues sur les instances choisies au paragraphe précédent. Les arbres de branchement ne contiennent presque pas de retour arrière. Ceci peut s'expliquer par la présence

d'items de même longueur et par l'existence de plusieurs solutions optimales entières différentes qui peuvent être obtenues en échangeant des sous-ensembles d'items.

Nous avons d'abord comparé les branchements sur les variables de flot fractionnaires proches de $\frac{1}{2}$ ($X_{ij} \sim 0.5$ pour $(i, j) \in A$) et sur les variables de flot fractionnaires proches de 1 ($X_{ij} \sim 1$ pour $(i, j) \in A$). Le choix de la variable dont le flot est le plus proche de 1 produit des arbres de branchement plus profonds mais est beaucoup moins coûteux en temps de calcul. Ceci est dû au fait que chaque colonne participe à la fonction coût par sa valeur et qu'en général, la plus grande partie du flot X_{ij} passe sur une seule colonne. Par conséquent si on fixe X_{ij} à 1, très peu de colonnes sont enlevées du problème maître. Donc, la réalisabilité du problème n'est pas affectée et la valeur du problème maître restreint résultant est facile à obtenir et est souvent égale à la valeur optimale. Ainsi, le nombre de pivots requis est faible et généralement aucune résolution du sous-problème n'est requise.

Nous avons ensuite testé un critère qui tient compte des poids des items: $1 - \frac{l_i - l_j}{L}$ pour $(i, j) \in A$. Ce critère tend à éviter le choix des arcs avec l_i grand et l_j très petit. Ce branchement donne des arbres de branchement plus réduits, mais est très coûteux en temps de calcul. Ceci est dû au fait qu'avec ce branchement, on peut choisir des variables $X_{ij} \sim 0$. Le problème maître restreint résultant est probablement non réalisable et plusieurs itérations de génération de colonnes sont effectuées. Une méthode qui serait efficace est de sélectionner, parmi les variables de flot $X_{ij} > 0.75$ par exemple, celle ayant $\frac{l_i - l_j}{L}$ le plus petit. Mais faute de pouvoir tester ceci directement, nous avons défini une fonction score qui est une combinaison convexe des deux critères

$$f(X_{ij}, \frac{l_i - l_j}{L}) = \frac{X_{ij} + 2(1 - \frac{l_i - l_j}{L})}{3}.$$

Les résultats sont assez encourageants et suggèrent que la méthode proposée ci-dessus produirait probablement de meilleurs résultats.

Tableau 5.5: Utilisation de différents critères pour le choix de la variable de branchement

Instances	u9i4		u9i15		u18i8		u18i17		t249i3		t249i14	
Score	t_{bb}	n_{bb}	t_{bb}	n_{bb}	t_{bb}	n_{bb}	t_{bb}	n_{bb}	t_{bb}	n_{bb}	t_{bb}	n_{bb}
$X_{ij} \sim .5$	294.3	184	337.1	165	54.3	97	55.4	119	128.8	58	152.0	62
$X_{ij} \sim 1$	75.0	197	113.3	211	19.8	162	17.0	152	34.1	66	63.2	125
$1 - \frac{l_i - l_j}{L}$	499.2	134	530.0	149	65.8	83	91.1	86	119.6	44	154.2	52
$f(X_{ij}, \frac{l_i - l_j}{L})$	85.0	161	87.7	167	13.9	84	15.4	112	46.1	67	60.0	73

Pré-traitement à chaque nœud de l'arbre La proposition suivante montre comment on peut se servir de la procédure de pré-traitement de Martello et Toth pour accélérer le processus de branchement.

Proposition 4 : Supposons qu'une décision de branchement $X_{ij} = 1$ ait été prise à un nœud de branchement donné. Si on peut trouver un item k tel que :

- i) k est libre, i.e., n'est impliqué dans aucune autre contrainte du type $X_{kl} = 1$,
 $X_{lk} = 1$, $X_{jk} = 0$ ou $X_{ki} = 0$ au nœud considéré,
- ii) $k < i$ ou $k > j$,
- iii) $(l_i + l_j) + l_k = L$,

alors, il existe une solution optimale entière au nœud considéré avec $Y_p = 1$, où p est le patron $\{l_i, l_j, l_k\}$.

Preuve: La décision $X_{ij} = 1$ revient à agréger les deux items i et j en un seul item ij de longueur $l_{ij} = l_i + l_j$ et à interdire l'association de cet item avec un item $k \in [i, j]$ (car le réseau est acyclique). Donc si k est libre (i), le patron $\{l_i, l_j, l_k\}$, noté p , est réalisable pour le nœud considéré. Et d'après la proposition 3, il existe une solution optimale entière au problème de ce nœud avec $Y_p = 1$.■

Ce résultat peut permettre de réduire la taille du problème de découpe au nœud courant et d'avoir une meilleure borne inférieure. Cette procédure sera probablement efficace dans le cas des triplets grâce à la structure particulière de la solution de ces problèmes. Mais aussi pour les autres problèmes, car la solution contient souvent beaucoup de colonnes composées de trois items. Le résultat peut être étendu au cas d'un plus grand nombre d'items, mais la procédure devient plus compliquée. Cependant, si les contraintes $X_{ij} = 1$ sont éliminées en agrégeant les items, cette procédure sera plus simple à implanter.

Ce résultat est aussi valable pour le branchement proposé par Vance *et al.* (1994) pour le problème de découpe binaire avec la particularité qu'il n'est plus nécessaire de vérifier la condition (ii) car le sous-problème est un problème de sac-à-dos binaire.

Plusieurs décisions à la fois Dans le cas du choix de la variable X_{ij} la plus proche de 1, le nombre de nœuds de branchement est élevé alors que le temps consommé est petit et il n'y a pratiquement pas de retour arrière. Ceci nous suggère de prendre plusieurs décisions à la fois dans le but d'obtenir la solution optimale entière en n'examinant qu'un nombre restreint de nœuds. Le schéma de branchement est illustré par la figure 5.3. Les nœuds intermédiaires sont créés, mais le nœud suivant

traité est le dernier.

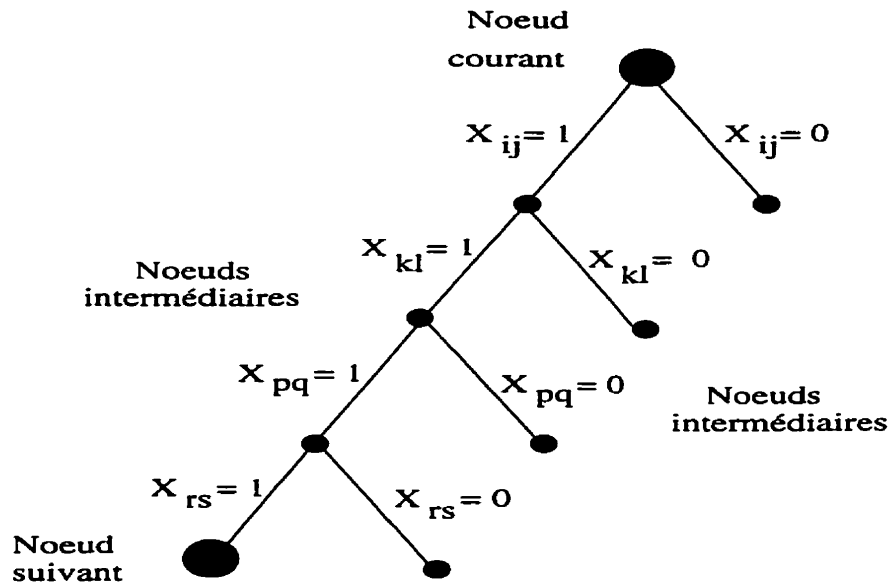


Figure 5.3: Branchement sur plusieurs variables

Les résultats sont présentés dans le tableau 5.6. Dans ce tableau, z^* donne le nombre de rouleaux dans une solution optimale entière et z^h est le nombre de rouleaux obtenu par la version de la procédure décrite sans retour arrière. Le tableau compare la procédure qui branche sur les variables proches de 1 et la procédure qui fixe à 1 toutes les variables supérieures à un seuil donné ($\{X_{ij} \sim 1\}$). On déduit que la procédure proposée est probablement très efficace. Le temps passé dans le branchement est nettement réduit ainsi que le nombre de nœuds, et la solution optimale a été obtenue pour toutes les instances testées. Cependant, il est fort probable que pour une autre instance la version heuristique de cette procédure ne fournirait pas la solution optimale. Mais la valeur de la solution obtenue serait proche de l'optimum et peu de retours arrières seraient nécessaires afin d'atteindre l'optimalité.

Tableau 5.6: Branchement sur plusieurs variables simultanément

Instances	u9i4		u9i15		u18i8		u18i17		t249i3		t249i14	
z^*	82		71		84		86		83		83	
z^h	82		71		84		86		83		83	
Score	t_{bb}	n_{bb}	t_{bb}	n_{bb}	t_{bb}	n_{bb}	t_{bb}	n_{bb}	t_{bb}	n_{bb}	t_{bb}	n_{bb}
$X_{ij} \sim 1$	75.0	197	113.3	211	19.8	162	17.0	152	34.1	66	63.2	125
$\{X_{ij} \sim 1\}$	19.4	25	15.8	24	3.0	19	2.5	12	18.6	15	17.3	39

5.5.2 Relaxation linéaire

Borne inférieure de Farley (1990) La borne proposée permet d'arrêter l'optimisation de la relaxation linéaire en obtenant une borne inférieure sur la valeur de la solution optimale entière.

Proposition 5 : Soit \bar{z} la valeur optimale d'un problème maître restreint et \bar{c}_{min} le coût réduit minimum sur toutes les colonnes correspondant à ce problème maître restreint, i.e. , $\bar{c}_{min} = \min_{p \in \Omega} \{\bar{c}_p\}$.

Si on note par z_{lp} et z^* les valeurs de la solution optimale de la relaxation linéaire et de la solution optimale entière, Alors

- i) $\frac{\bar{z}}{1-\bar{c}_{min}} \leq z_{lp}$
- ii) si $\lceil \frac{\bar{z}}{1-\bar{c}_{min}} \rceil \geq \lceil \bar{z} \rceil$ alors $z^* \geq \lceil \frac{\bar{z}}{1-\bar{c}_{min}} \rceil = \lceil \bar{z} \rceil = \lceil z_{lp} \rceil$

Preuve :

- i) Étant donné les multiplicateurs (variables duales) optimaux π_i ($i \in N$) correspondant au problème maître restreint considéré, le coût réduit d'une variable Y_p ($p \in \Omega$) est donné par

$$\bar{c}_p = 1 - \sum_{i \in N} \pi_i a_{ip}.$$

La borne lagrangienne sur z_{lp} obtenue en dualisant les contraintes de demande est

$$\sum_{p \in \Omega} Y_p + \sum_{i \in N} \pi_i \sum_{p \in \Omega} (1 - a_{ip} Y_p).$$

En regroupant les termes contenant Y_p on obtient alors

$$\sum_{p \in \Omega} Y_p \bar{c}_p + \sum_{i \in N} \pi_i \leq z_{lp}.$$

Comme $Y_p \geq 0$, $\bar{c}_p \geq \bar{c}_{min}$ ($\forall p \in \Omega$) et $\bar{z} = \sum_{i \in N} \pi_i$, alors

$$\sum_{p \in \Omega} Y_p \bar{c}_{min} + \bar{z} \leq z_{lp}.$$

Donc, si M est une borne supérieure sur le nombre de rouleaux à l'optimalité linéaire et sachant que $\bar{c}_{min} \leq 0$, on obtient la relation suivante :

$$\bar{z} + M \bar{c}_{min} \leq z_{lp}.$$

En particulier dans le cas de BCSP, le nombre de rouleaux à l'optimalité linéaire est exactement z_{lp} . Donc, on peut remplacer M par z_{lp} et obtenir la relation suivante

$$\bar{z} + z_{lp} \bar{c}_{min} \leq z_{lp}.$$

Et par conséquent, on obtient la borne dite de Farley :

$$\frac{\bar{z}}{(1 - \bar{c}_{min})} \leq z_{lp}.$$

- ii) De la relation i), on déduit que

$$\lceil \frac{\bar{z}}{(1 - \bar{c}_{min})} \rceil \leq \lceil z_{lp} \rceil.$$

Donc, si $\lceil \frac{\bar{z}}{(1 - \bar{c}_{min})} \rceil \geq \lceil \bar{z} \rceil \geq \lceil z_{lp} \rceil$, alors

$$z^* \geq \lceil \frac{\bar{z}}{(1 - \bar{c}_{min})} \rceil = \lceil \bar{z} \rceil = \lceil z_{lp} \rceil. \blacksquare$$

Cette borne domine la borne lagrangienne, mais n'est active que très près de l'optimum, et donc ne permet pas d'arrêter le processus de résolution rapidement. Les résultats des tests du tableau 5.7 montrent qu'au nœud 0, le temps de calcul et le nombre d'itérations sont très semblables pour la version standard ou celle utilisant la borne de Farley.

Tableau 5.7: Utilisation de la borne de Farley

Instances	u9i4		u9i15		u18i8		u18i17		t249i3		t249i14	
Score	t_{lr}	n_{lr}	t_{lr}	n_{lr}	t_{lr}	n_{lr}	t_{lr}	n_{lr}	t_{lr}	n_{lr}	t_{lr}	n_{lr}
<i>standard</i>	445.0	56	367.5	43	149.3	44	174.7	54	194.9	118	180.2	133
<i>Farley</i>	433.4	55	367.5	43	123.4	38	174.7	54	194.9	118	180.2	133

Nous proposons maintenant une amélioration de la borne de Farley pour le problème BCSP basée sur les items (ou les nœuds) qui n'ont pas de prédécesseur autre que le nœud origine dans le réseau (acyclique).

Proposition 6 : Soit U l'ensemble des nœuds sans prédécesseur (autre que l'origine o) dans le réseau. Pour $i \in U$, on note par P_i l'ensemble des chemins réalisables débutant par le nœud i et $\bar{c}_{imin} = \min_{p \in P_i} \{\bar{c}_p\}$. Notons aussi $UP = \cup_{i \in U} P_i$. \bar{z} et

\bar{c}_{min} sont tels que définis dans la proposition 5. Alors,

$$\frac{\bar{z} + \sum_{i \in U} (\bar{c}_{imin} - \bar{c}_{min})}{1 - \bar{c}_{min}} \leq z_{lp}.$$

Preuve : L'expression de la borne inférieure de Lagrange peut s'écrire :

$$\bar{z} + \sum_{i \in U} \sum_{p \in P_i} \bar{c}_p Y_p + \sum_{p \in (\Omega - UP)} \bar{c}_p Y_p.$$

Par le même raisonnement que précédemment et en remarquant que $\sum_{p \in P_i} Y_p = 1$ ($i \in U$) et $\sum_{p \in (\Omega - UP)} Y_p = z_{lp} - |U|$, on obtient

$$\frac{\bar{z} + \sum_{i \in U} (\bar{c}_{imin} - \bar{c}_{min})}{1 - \bar{c}_{min}} \leq z_{lp}. \blacksquare$$

Cette borne domine la borne de Farley pour le problème de découpe car $\bar{c}_{imin} \geq \bar{c}_{min}$ ($i \in U$). On peut aussi obtenir une borne semblable qui domine la borne de Farley en remplaçant \bar{c}_{min} par

$$\bar{c}'_{min} = \min_{p \in (\Omega - UP)} \{\bar{c}_p\}$$

dans la relation précédente.

Mais, ces bornes présentent le même inconvénient que la borne de Farley pour le problème de découpe binaire. Elles ne deviennent actives que très près de l'optimum. Cependant, l'extension au cas général pourrait être intéressante si elle profite du fait que les demandes b_i des items peuvent être plus grande que 1.

Borne inférieure de Martello et Toth (1990) Pour $\alpha \leq \frac{L}{2}$ donné, la procédure suivante fournit une borne inférieure sur le nombre de rouleaux à l'optimalité entière du problème de découpe, notée $LB(\alpha)$. Chaque item de longueur $l_i > \frac{L}{2}$ est affecté à

un rouleau. On calcule ensuite l'espace minimum nécessaire pour découper les items dont les longueurs l_i vérifient $\alpha \leq l_i \leq L - \alpha$. On soustrait de cette valeur le nombre de rouleaux initialisés par des items tels que $\frac{L}{2} \leq l_i \leq L - \alpha$. La borne est alors calculée en ajoutant le nombre de rouleaux initialisés. Formellement, cette borne est donnée par

$$LB(\alpha) = |\{i : l_i > \frac{L}{2}\}| + \max\{0, \lceil \frac{\sum_{i \in (I_1 \cup I_2)} l_i}{L} \rceil - |I_2|\}$$

où $I_1 = \{i : \alpha \leq l_i \leq \frac{L}{2}\}$ et $I_2 = \{i : L - \alpha \geq l_i > \frac{L}{2}\}$.

La borne proposée par Martello et Toth (1990) est le meilleur $LB(\alpha)$ pour $\alpha \leq \frac{L}{2}$. C'est-à-dire

$$LB^2 = \max\{LB(\alpha) : 0 \leq \alpha \leq \frac{L}{2}\}.$$

Les auteurs ont montré que cette borne peut être calculée par la formule suivante

$$LB^2 = \max \{LB(\alpha) : \alpha = l_i \leq \frac{L}{2} \text{ pour } i \in N\},$$

et que si on note par $LB^1 = \lceil \frac{\sum_{i \in N} l_i}{L} \rceil$ la borne inférieure fournie par la formulation de Kantorovich (proposition 1, section 3.1), alors la relation suivante est vérifiée :

$$LB^1 \leq LB^2 \leq \lceil z_{lp} \rceil.$$

Cette borne est de très bonne qualité en pratique et elle peut être calculée en $O(n)$, une fois les items triés suivant leur longueur. Pour les problèmes difficiles que nous testons, elle fournit toujours la valeur de la solution optimale qui est égale à $\lceil z_{lp} \rceil$. Ceci permettra d'arrêter la résolution de la relaxation linéaire dès que cette borne est atteinte. Or, nous avons constaté que les dernières itérations sont les plus coûteuses en temps de calcul. Par conséquent, l'utilisation de la borne LB^2 permettra de

réduire le temps de résolution et le nombre d'itérations de la relaxation linéaire considérablement. Cependant, dans le cas des triplets $LB^2 = LB^1 = z_{lp}$ et cette borne n'a pas d'effet sur l'optimisation.

Génération au préalable des meilleurs paires et des triplets Comme nous l'avons remarqué plus haut, les solutions optimales linéaire et entière contiennent beaucoup de colonnes avec un nombre restreint d'items, i.e., 2 ou 3. Nous avons donc pensé à résoudre le problème en imposant une limite sur le nombre d'items par rouleau. Mais, l'implantation dans l'optimiseur GENCOL nécessite au moins une ressource additionnelle et présente les difficultés de convergence d'une procédure de génération de colonnes. Nous avons donc choisi de générer les meilleurs paires et triplets pour les "gros" items, et de les proposer avec les colonnes de la solution heuristique pour initialiser le problème. Cette procédure nécessite un temps de calcul de $O(n^2)$ mais comme seuls les "gros" items sont traités, elle est très efficace en pratique. Les résultats sont très bons pour les triplets, mais aussi pour les problèmes uniformes.

Le tableau 5.8 résume les résultats obtenus avec les deux dernières techniques pour le nœud 0. *BFD* est la valeur de la solution fournie par l'heuristique BFD. *MT* désigne la version de la procédure utilisant la borne LB^2 de Martello et Toth et *PT* désigne la version utilisant l'initialisation par les meilleurs paires et triplets. Le tableau présente également les résultats obtenus pour l'utilisation de cette technique conjointement avec la borne LB^2 (*MT&PT*). Les résultats sont très nettement améliorés. Le nombre d'itérations devient très petit et le temps de calcul est sensiblement réduit.

Tableau 5.8: Au nœud 0, utilisation de la borne LB^2 et initialisation par les meilleurs paires et triplets

Instances	u9i4	u9i15	u18i8	u18i17	t249i3	t249i14
z^*	82	71	84	86	83	83
z_{lp}	81.8	70.8	83.7	85.4	83.0	83.0
LB^1	81	70	83	85	83	83
LB^2	82	71	84	86	83	83
BFD	83	72	87	89	95	96

Version	t_{lr}	n_{lr}	t_{lr}	n_{lr}	t_{lr}	n_{lr}	t_{lr}	n_{lr}	t_{lr}	n_{lr}	t_{lr}	n_{lr}
<i>standard</i>	445.0	56	367.5	43	149.3	44	174.7	54	194.9	118	180.2	133
<i>MT</i>	264.6	41	232.9	34	64.1	26	86.0	41	194.9	118	180.2	133
<i>PT</i>	105.7	9	108.8	10	57.0	13	74.2	15	14.7	4	2.3	3
<i>MT&PT</i>	10.2	3	2.7	2	22.8	6	25.9	7	14.7	4	2.3	3

Le tableau 5.9 donne les résultats obtenus en utilisant simultanément la borne LB^2 , l'initialisation par les meilleurs paires et les triplets et la version heuristique du branchement simultané sur plusieurs variables ($MT\&PT\&\{X_{ij}\}$). Il y a une nette amélioration du temps de calcul et du nombre de nœuds de branchement. Les temps de calcul deviennent comparables à ceux de Valério de Carvalhò (1996) et Vanderbeck (1996) qui ont présenté les résultats les plus performants.

Cependant, nous remarquons que les itérations de résolution du problème maître et du sous-problème coûtent assez cher. Les instances les plus faciles (u18i8 et u18i17) coûtent presque aussi cher que les autres instances qui sont plus difficiles. Le temps du sous-problème dépend du nombre d'itérations de génération de colonnes effectuées. Mais, le temps du problème maître est assez élevé comparé à celui du sous-problème. Valério de Carvalhò et Vanderbeck ont attribué l'efficacité de leur techniques de résolution respectives aux économies d'échelle qui résultent de l'agrégation des items de longueurs égales. De plus, les difficultés que nous avons rencontrées dans la résolution de la relaxation linéaire sont semblables à celles présentes dans l'ap-

Tableau 5.9: Résultats combinant l'utilisation de la borne LB^2 , l'initialisation par les meilleurs paires et triplets et le branchement simultané sur plusieurs variables de flot

Pb	Version	t_{tot}	t_{lp}	t_{bb}	t_{sp}	t_{mp}	n_{bb}	n_{tot}
u9i4	standard	739.3	445.0	294.3	378.1	361.3	184	275
u9i4	$MT\&PT\&\{X_{ij}\}$	31.4	10.2	21.2	12.5	18.9	17	34
u9i15	standard	704.6	367.5	337.1	338.0	366.0	165	244
u9i15	$MT\&PT\&\{X_{ij}\}$	6.7	2.7	4.0	3.1	3.6	6	9
u18i8	standard	203.6	149.3	54.3	92.8	91.9	97	148
u18i8	$MT\&PT\&\{X_{ij}\}$	26.1	22.8	3.3	11.8	14.3	14	39
u18i17	standard	230.1	174.7	55.4	117.6	91.4	119	187
u18i17	$MT\&PT\&\{X_{ij}\}$	32.2	25.9	6.3	14.1	18.1	27	47
t249i3	standard	323.7	194.9	128.8	225.4	81.6	58	196
t249i3	$MT\&PT\&\{X_{ij}\}$	18.0	14.7	3.3	11.4	6.6	11	17
t249i14	standard	332.2	180.2	152.0	219.3	96.6	62	218
t249i14	$MT\&PT\&\{X_{ij}\}$	5.9	2.3	3.6	1.1	2.8	33	37

proche de Vance *et al.* (1994) qui considèrent les items individuellement (c'est-à-dire demande = 1). Dans ce qui suit, nous allons considérer la formulation agrégée du problème maître et étudier son effet sur la méthode de résolution proposée.

Chapitre 6

Généralisation

Dans ce chapitre, nous étudions l'effet de l'agrégation des items de longueur égale sur la formulation du problème maître. Le réseau du cas binaire (figure 5.1) reste valide pour le cas général. Chaque unité de demande est représentée par un nœud. C'est ce réseau qui sera utilisé pour la résolution du sous-problème. Nous proposons également une généralisation du schéma de branchement qui devient sensiblement différent. Des résultats pour la résolution de la relaxation linéaire et pour le branchement sont présentés. Enfin, nous terminons le chapitre par la présentation de diverses techniques d'amélioration pour la résolution du sous-problème ainsi que pour le branchement.

6.1 Agrégation des items

La formulation d'un problème de découpe binaire comme un problème de découpe générale impose implicitement que les items de même longueur aient des variables duales égales, i.e., $l_i = l_j \Rightarrow \pi_i = \pi_j$. La relaxation de ces contraintes dans la formulation de BCSP introduit de l'instabilité dans le processus de convergence des variables duales vers des valeurs optimales et fait que cette convergence est très lente. La dégénérescence du dual entraîne la décroissance de l'objectif primal avec des variables duales mal ajustées. Ainsi, au voisinage de l'optimalité, la stabilisation

des valeurs des variables duales nécessite un effort considérable en temps de calcul et en itérations de génération de colonnes.

Nous allons donc résoudre le problème en considérant chaque ensemble d'items de longueur égale comme un seul item dont la demande est égale au nombre d'items correspondants. L'ensemble des items résultant sera noté par E . Le problème maître comportera une seule contrainte de couverture pour l'ensemble des items identiques; le second membre de la contrainte pourra alors être strictement plus grand que 1. La taille du problème maître est ainsi réduite et sa résolution devient plus facile et plus rapide. Le sous-problème de génération de colonnes reste inchangé. Chaque nœud correspond à un item de demande 1 avec la différence que les items ayant la même longueur auront la même valeur pour la variable duale correspondante. Par conséquent, les arcs qui partent de ces nœuds auront tous le même coût réduit.

Les résultats de l'utilisation de cette approche pour la résolution de la relaxation linéaire sont résumés dans le tableau 6.1. AG indique que le problème maître est résolu en utilisant la formulation agrégée. L'utilisation de la borne LB^2 et l'initialisation par les meilleures paires et triplets accélèrent le processus d'optimisation. La deuxième technique fournit une meilleure solution initiale et la première permet d'arrêter l'optimisation dès l'obtention de la borne inférieure $\lceil z_{lp} \rceil$ sur l'objectif entier. Dans tous les cas, le problème maître devient stable et sa résolution très efficace. On remarquera que les temps de calcul de la relaxation linéaire sont tous inférieurs à 1 seconde pour les 6 problèmes tests retenus.

Tableau 6.1: Au nœud 0, agrégation des items de même longueur, utilisation de la borne LB^2 et initialisation par les meilleurs paires et triplets

Instances	u9i4			u9i15			u18i8		
z^*	82			71			84		
z_{lp}	81.8			70.8			83.7		
LB^1	81			70			83		
LB^2	82			71			84		
BFD	83			72			87		
Version	t_{lr}	n_{lr}	t_{mp}^{lr}	t_{lr}	n_{lr}	t_{mp}^{lr}	t_{lr}	n_{lr}	t_{mp}^{lr}
<i>standard</i>	445.0	56	148.8	367.5	43	110.2	149.3	44	59.2
AG	79.9	12	0.6	100.6	17	0.7	30.4	15	0.4
AG&MT	26.4	7	0.3	61.2	11	0.4	12.5	10	0.2
AG&PT	22.5	4	0.2	23.4	4	0.2	3.9	3	0.1
AG&MT&PT	0.3	2	0.1	0.2	2	0.1	0.2	2	0.1

Instances	u18i17			t249i3			t249i14		
z^*	86			83			83		
z_{lp}	85.4			83.0			83.0		
LB^1	85			83			83		
LB^2	86			83			83		
BFD	89			95			95		
Version	t_{lr}	n_{lr}	t_{mp}^{lr}	t_{lr}	n_{lr}	t_{mp}^{lr}	t_{lr}	n_{lr}	t_{mp}^{lr}
<i>standard</i>	174.7	54	67.4	194.9	118	44.6	180.2	133	43.4
AG	32.3	16	0.4	119.8	97	6.9	107.4	93	7.8
AG&MT	12.2	10	0.2	119.8	97	6.9	107.4	93	7.8
AG&PT	7.3	4	0.1	0.5	2	0.4	0.6	2	0.4
AG&MT&PT	0.2	2	0.1	0.5	2	0.4	0.6	2	0.4

6.2 Branchement

Nous utilisons une méthode de branchement sur les variables de flot (arcs)

$$X_{ij} = \sum_{k \in K} X_{ij}^k.$$

Le nombre d'arcs dans le réseau étant fini ($\frac{(n+2)(n+1)}{2}$) et chaque variable X_{ij} étant bornée supérieurement par le nombre d'items du même type que i et par le nombre d'items du même type que j , le nombre de décisions possibles est donc fini.

À la différence du cas binaire, si on agrège les contraintes associées aux items de même longueur, un item de longueur donnée peut être couvert plus d'une fois et un patron de découpe p peut être utilisé plus d'une fois dans une solution du problème maître. Les variables X_{ij} peuvent alors prendre des valeurs plus grandes que 1. Une variable de flot X_{ij} est candidate au choix de la variable de branchement si à l'optimalité du problème maître restreint $X_{ij} = f_{ij}$ où f_{ij} est fractionnaire, $f_{ij} > 0$ et éventuellement $f_{ij} > 1$. Les décisions de branchement correspondantes sont

$$X_{ij} \leq \lfloor f_{ij} \rfloor \tag{6.1}$$

et

$$X_{ij} \geq \lceil f_{ij} \rceil. \tag{6.2}$$

Problème maître Ces contraintes ne peuvent être imposées directement dans le sous-problème de génération de colonnes sans altérer sa structure. Pour qu'elles puissent être ajoutées au problème maître, il faut les exprimer en terme des variables

Y_p ($p \in \Omega$). Or, nous pouvons écrire

$$X_{ij} = \sum_{p \in \Omega} x_{ijp} Y_p.$$

Ainsi, on peut exprimer toute contrainte qui porte sur un sous-ensemble des variables X_{ij} ($(i, j) \in A$) en fonction des variables Y_p .

Pour chaque ensemble d'items I_e ($e \in E$) de même longueur, on note par $b_e = |I_e|$ la demande totale associée à ce type d'items et par $a_{ep} = \sum_{i \in I_e} a_{ip}$ la contribution de la colonne $p \in \Omega$ à la couverture des items du type e . Si pour un nœud de l'arbre de branchement u , on désigne par L^u l'ensemble des arcs impliqués dans une décision de branchement du type (6.1) et par U^u l'ensemble des arcs impliqués dans une décision de branchement du type (6.2), la formulation du problème maître est alors la suivante

$$\text{Minimiser } \sum_{p \in \Omega} Y_p \quad (6.3)$$

sujet à :

$$\sum_{p \in \Omega} a_{ep} Y_p \geq b_e, \quad \forall e \in E \quad (6.4)$$

$$\sum_{p \in \Omega} x_{ijp} Y_p \leq \lfloor f_{ij} \rfloor, \quad \forall (i, j) \in L^u \quad (6.5)$$

$$\sum_{p \in \Omega} x_{ijp} Y_p \geq \lceil f_{ij} \rceil, \quad \forall (i, j) \in U^u \quad (6.6)$$

$$Y_p \geq 0, \quad \forall p \in \Omega \quad (6.7)$$

$$Y_p \in Z^+, \quad \forall p \in \Omega. \quad (6.8)$$

La relaxation linéaire de cette formulation est obtenue en éliminant les contraintes (6.8). La résolution de celle-ci par génération de colonnes utilise le sous-problème décrit dans le paragraphe suivant.

Sous-problème Si $\alpha_{ij} \geq 0$ ($(i, j) \in L^u$) sont les variables duales associées aux contraintes (6.5) et $\beta_{ij} \geq 0$ ($(i, j) \in U^u$) sont les variables duales associées aux contraintes (6.6), alors la fonction coût du sous-problème est donnée par

$$1 - \sum_{(i,j) \in A} (\pi_i - \alpha_{ij} + \beta_{ij}) X_{ij} \quad (6.9)$$

avec les conventions $\pi_0 = 0$, $\alpha_{ij} = 0$ si $(i, j) \notin L^u$ et $\beta_{ij} = 0$ si $(i, j) \notin U^u$.

La structure du sous-problème reste ainsi inchangée et l'effet des décisions de branchement se traduit par la modification des coûts réduits des arcs concernés en utilisant les variables duales qui correspondent aux contraintes ajoutées au problème maître. Cependant, la structure de problème de sac-à-dos n'est pas conservée car tous les arcs qui partent d'un même nœud n'ont plus nécessairement le même coût. De même si la décision de branchement porte sur plusieurs arcs, l'effet de l'ajout de la contrainte au problème maître se traduira par la modification des coûts réduits des arcs concernés.

Les décisions de branchement peuvent aussi avoir un effet sur le réseau du sous-problème de génération de colonnes. Par exemple, si la décision est du type $X_{ij} \leq 0$, alors l'arc (i, j) peut être éliminé du réseau et la contrainte ne sera pas ajoutée au problème maître. Aussi, si la décision $X_{ij} \geq f$ où $i \in I_e$ (resp. $j \in I_e$) et $f \geq b_e$ est prise et qu'aucun autre nœud $i' \in I_e$ (resp. $j' \in I_e$) n'est présent dans une décision du même type, alors tous les arcs (i', l) (resp. (l, j')) peuvent être éliminés du réseau. De même, si une décision du type $X_{ij} \geq f$ où $i \in I_e$ et f est plus grand ou égal à la demande restante compte tenu des décisions de branchement précédentes, alors les arcs (i', l) ($i' \in I_e, i' \neq i$) non concernés par des décisions du type $X_{i'l} \geq f'$ peuvent être éliminés du réseau. L'ajout de la contrainte au problème maître devient alors

inutile.

Un travail similaire peut être fait au niveau du problème maître pour éliminer les contraintes redondantes et déduire des bornes supérieures ou inférieures sur les seconds membres des contraintes ajoutées.

L'existence d'une variable fractionnaire dans la solution du problème maître ne garantit pas l'existence d'une variable de flot fractionnaire. La figure 6.1 montre un exemple où les variables de chemins sont fractionnaires alors que les flots sur les arcs sont tous entiers. Tous les chemins représentent des patrons maximaux et portent un flot de 0.5.

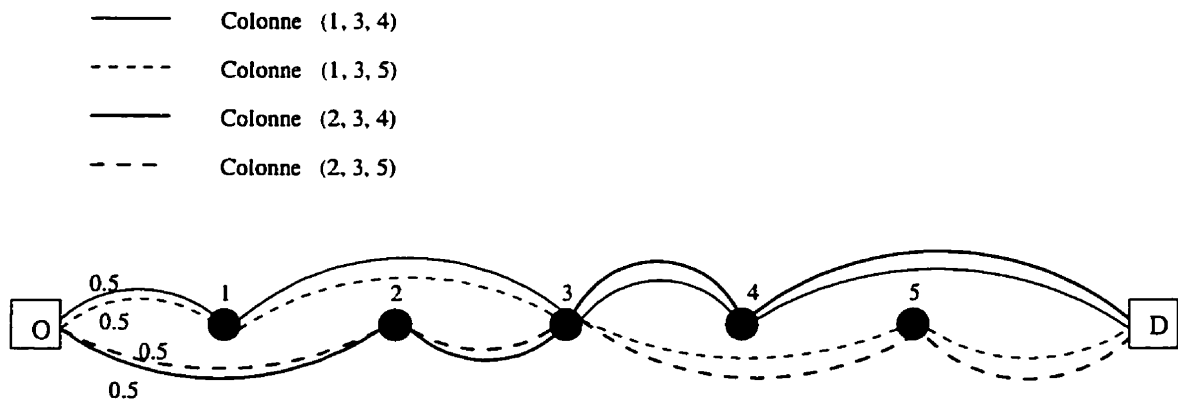


Figure 6.1: Exemple d'une solution fractionnaire en terme des chemins qui donne des flots entiers sur les arcs.

Mais, le résultat du théorème 1 (section 5.1) sur le branchement dans le cas binaire garantit dans cette situation l'existence d'une copie d'un item (un nœud) qui n'est pas couvert par l'ensemble des colonnes de la solution. Ceci est justifié par le théorème qui suit.

Théorème 2: Soit Y_p ($p \in \Omega$) une solution optimale linéaire à un nœud de l'arbre de branchement. Alors,

- 1) Si Y_p est entier pour tout $p \in \Omega$, alors X_{ij} est entier pour tout $(i, j) \in A$.
- 2) Si $\exists p \in \Omega$ tel que Y_p est fractionnaire, alors l'une au moins des deux assertions suivantes est vraie :
 - i) $\exists (i, j) \in A$ tel que X_{ij} est fractionnaire
 - ii) \exists un item i (considéré individuellement) qui n'est pas couvert par la solution Y_p ($p \in \Omega$).

Preuve:

- 1) Le résultat découle directement de la relation

$$X_{ij} = \sum_{p \in \Omega} x_{ijp} Y_p, \quad \forall (i, j) \in A.$$

- 2) S'il existe un nœud du réseau par lequel passe un flot > 1 , alors ii) est vérifiée (mais il est aussi possible qu'il existe $(i, j) \in A$ tel que X_{ij} est fractionnaire (i)).

Par contre, si tous les nœuds sont couverts par au plus 1 unité de flot, alors deux situations peuvent se présenter

- Il existe un nœud i^* ($i^* \notin \{o, d\}$) par lequel passe un flot < 1 , c'est-à-dire $0 < \sum_{\{j: (i^*, j) \in A\}} X_{i^*, j} < 1$. Alors, il existe forcément un nœud j^* tel que $0 < X_{i^*, j^*} < 1$ (i).

- Tous les nœuds $i \in N - \{o, d\}$ sont couverts par 1 unité de flot. On se retrouve alors dans le cas binaire. Donc, d'après le théorème 1 (section 5.1) et par l'hypothèse $\exists p \in \Omega$ tel que Y_p est fractionnaire, il existe $(i, j) \in A$ tel que X_{ij} est fractionnaire (i).■

Ainsi, si la solution du problème maître restreint n'est pas entière, on cherche un arc (i, j) tel que X_{ij} est fractionnaire. Si un tel arc existe, alors le branchement se fera par rapport à la variable X_{ij} . Dans le cas contraire et si $i \in I_e$ est un item qui vérifie (ii), on remplace I_e par $I_e - \{i\}$ et la contrainte (6.4) correspondante par

$$\sum_{p \in \Omega} a_{ep} Y_p \geq b_e - 1$$

et

$$\sum_{p \in \Omega} a_{ip} Y_p \geq 1.$$

Cette façon de prendre une décision de branchement peut être vue comme la détermination d'une variable X_{ij}^k fractionnaire. La modification correspondante du problème maître revient à exiger que l'ensemble des colonnes qui couvrent l'item i ait un flot total entier (unitaire).

Ceci garantit la convergence de la procédure de branchement proposée. Au pire cas, nous retrouverons la formulation binaire du problème maître pour laquelle chaque unité de demande est considérée individuellement. Cependant, il est rare de rencontrer une telle situation en pratique. Les décisions de branchement correspondront presque toujours à des variables de flot X_{ij} fractionnaires.

Tests préliminaires Nous utiliserons les mêmes instances que précédemment pour ces tests. La variable choisie est celle qui présente la plus grande partie fractionnaire. Une seule variable X_{ij} est impliquée dans une décision de branchement. Les simplifications décrites plus haut ne sont pas effectuées. Une décision a pour effet d'ajouter une contrainte au problème maître. Par conséquent, le nombre de contraintes dans le problème maître risque de croître rapidement. Nous utilisons une stratégie en profondeur d'abord. La première branche explorée est celle qui correspond à $X_{ij} \geq \lceil f_{ij} \rceil$.

Le réseau utilisé pour ces tests est celui qui correspond à la formulation en problème BCSP. Il présente beaucoup de symétrie et comporte beaucoup de redondances. Sa résolution est donc coûteuse, d'autant plus que, contrairement au cas binaire, il n'est pas réduit tout au long du processus de résolution.

Les résultats des instances des problèmes u18 et t249 sont présentés dans le tableau 6.2. Un nombre maximum de 100 nœuds de branchement sont résolus à cause du coût de résolution élevé du sous-problème. Dans le cas des instances u9i4 et u9i15, une solution optimale n'a pu être trouvée après la résolution de 100 nœuds.

Tableau 6.2: Résultats préliminaires pour la formulation agrégée du problème maître

Pb	Version	t_{tot}	t_{lr}	t_{bb}	t_{mp}	t_{sp}	n_{bb}	n_{tot}
u18i8	AG&MT&PT	37.6	0.2	37.4	13.2	24.4	79	93
u18i17	AG&MT&PT	16.4	0.2	16.2	9.1	7.3	45	50
t249i3	AG&MT&PT	35.3	0.5	34.8	27.6	7.7	49	54
t249i14	AG&MT&PT	17.1	0.6	16.5	17.0	0.1	32	34

Dans le cas des instances résolues, le nombre de nœuds de branchement n'est pas très élevé. Par contre, les itérations sont coûteuses en temps de calcul pour les raisons évoquées ci-dessus. Diverses techniques pour l'élimination des redondances et la réduction de la taille du problème maître peuvent être utilisées. On peut citer l'élimination des contraintes redondantes et l'agrégation de certaines contraintes. Les deux sections qui suivent présentent des techniques qui devraient être étudiées pour accélérer le branchement en imposant plus rapidement l'intégralité et rendre plus efficace la résolution du sous-problème.

Une technique que nous discutons dans ce paragraphe est l'utilisation d'une heuristique pour obtenir une solution entière après la résolution de chaque nœud de l'arbre de branchement en arrondissant la solution continue. Un exemple de telle heuristique consiste à fixer les colonnes dont le flot Y_p est proche de $\lceil Y_p \rceil$ à cette dernière valeur et appliquer ensuite l'heuristique BFD à l'ensemble des items qui ne sont pas couverts par les colonnes fixées. Nous avons testé l'effet de l'utilisation de cette heuristique sur la résolution des instances u9i4 et u9i14. Le tableau 6.3 montre qu'une solution optimale entière a été obtenue très rapidement. Ceci est dû en partie au fait que la solution produite par l'heuristique pour les problèmes du même type que le problème u9 est souvent de très bonne qualité. Ceci justifie l'étude de l'utilisation de telles techniques pour accélérer l'obtention des solutions entières dans le cadre de la résolution des programmes linéaires en nombres entiers.

Tableau 6.3: Utilisation d'une heuristique d'arrondissement à chaque nœud de l'arbre de branchement pour la résolution des deux instances du problème u9

Pb	Version	t_{tot}	t_{lr}	t_{bb}	t_{mp}	t_{sp}	n_{bb}	n_{tot}
u9i4	AG&MT&PT&RH	1.3	0.3	1.2	1.2	0.1	9	11
u9i15	AG&MT&PT&RH	3.0	0.2	2.8	2.9	0.1	22	24

6.3 Autres techniques de branchement

Plusieurs techniques peuvent être utilisées pour rendre le branchement plus efficace. Ces branchements utilisent des ensembles d'arcs et réduisent les possibilités de permutation de la fractionnarité entre les arcs. Nous en présentons quelques unes qui ne sont pas encore implémentées.

Flot sortant d'un nœud Comme les colonnes qui contribuent à la couverture d'un ensemble d'items de même longueur peuvent passer par des nœuds différents, alors dans une solution fractionnaire, il peut y avoir des nœuds ayant un flot sortant fractionnaire. Or, dans une solution entière, ces flots doivent être entiers. Donc, si pour un nœud i , $\sum_{j:(i,j) \in A} X_{ij} = f_i$ est fractionnaire, les décisions de branchement correspondantes sont

$$\sum_{(i,j) \in A} X_{ij} \leq \lfloor f_i \rfloor$$

et

$$\sum_{(i,j) \in A} X_{ij} \geq \lceil f_i \rceil.$$

Ces contraintes peuvent être exprimées en terme des variables Y_p , et pourront donc être ajoutées au problème maître.

Flot passant entre deux ensembles de nœuds Si e_1 et e_2 sont deux ensembles de nœuds correspondant à des items de même longueur, alors si le flot passant entre les deux ensembles de nœuds

$$\sum_{(i,j) \in I_{e_1} \times I_{e_2}} X_{ij}$$

est fractionnaire, les décisions de branchement correspondantes imposeront une borne inférieure ou supérieure sur ce flot. Ceci revient à imposer que l'ensemble des colonnes qui couvrent les nœuds de I_{e_1} et I_{e_2} successivement ait un flot total entier.

Flot passant sur un sous-ensemble d'arcs Si un sous-ensemble E d'arcs fait passer un flot total fractionnaire, alors des décisions de branchement peuvent être prises par rapport à ce flot. Cette technique de branchement généralise les deux précédentes. Elle serait particulièrement intéressante dans le cas où E contient un sous-ensemble des arcs qui partent d'un même nœud (ou ensemble de nœuds ayant des longueurs égales).

Utilisation de la borne LB^2 Certaines décisions de branchement peuvent être interprétées comme l'agrégation d'un sous-ensemble d'items. Ceci permettrait d'obtenir une meilleure borne LB^2 sur la valeur de la solution entière du nœud de branchement courant. Comme les autres contraintes ne font que restreindre le domaine du problème, alors la borne est toujours valide.

Pré-traitement La procédure de pré-traitement à un nœud de l'arbre de branchement proposée pour le cas binaire, peut être généralisée en considérant les décisions de branchement du type $X_{ij} \geq \lceil f \rceil$. Les colonnes obtenues sont alors fixées, mais les items couverts par ces colonnes ne sont pas forcément éliminés du problème. Seules les demandes des ensembles d'items de longueur égale sont mises à jour.

6.4 Autres techniques d'accélération

Réduction du réseau Le réseau du problème de découpe binaire (BCSP) présente beaucoup de symétrie qui peut affecter le processus de branchement et rendre plus difficile d'imposer l'intégralité dans le problème maître. Mais, le réseau proposé pour le problème général de découpe (CSP) (figure 4.4, section 4.2) peut être utilisé car le problème maître utilise la formulation agrégée. La taille du réseau est alors réduite de moitié, et la symétrie est éliminée. Le schéma de branchement s'adapte facilement à ce réseau. De plus, le nouveau réseau permet de manipuler facilement l'information sur le nombre d'items d'une longueur donnée que peut contenir un rouleau.

Génération des colonnes pleines L'utilisation de la dominance dans la résolution du sous-problème favorise la génération des colonnes les moins remplies. Ceci est désavantageux dans le cas du problème CSP. Un moyen d'encourager la génération des colonnes remplies est de modifier le coût réduit d'une colonne pour la résolution du problème de plus court chemin de la manière suivante :

$$\bar{c}_p = \bar{c}_p + \alpha(L - \sum_i l_i),$$

où $\alpha = \frac{1}{L}$ par exemple.

Si des chemins de coût réduit \bar{c}_p négatif sont générés, alors leur coût réduit original est négatif et les colonnes correspondantes peuvent être ajoutées au problème maître restreint. Par contre si aucun chemin n'est généré, le SPC est résolu avec l'expression initiale du coût pour vérifier l'optimalité et éventuellement générer des colonnes de coût réduit $\bar{c}_p < 0$.

Accélération de la résolution au voisinage de l'optimum Les dernières itérations de résolution du sous-problème sont particulièrement coûteuses en temps de calcul. La principale raison est que les valeurs optimales des variables duales sont corrélées avec les valeurs $\frac{l_i}{L}$. Et comme la longueur consommée par un sous-chemin C arrivant au nœud i est $\sum_{j \in C} l_j$, alors la dominance élimine très peu de sous-chemins et l'optimisation est très lourde. Cependant, au voisinage de l'optimum on peut écrire

$$\pi_i = \frac{l_i}{L} + \varepsilon_i$$

où ε_i est petit et peut être positif ou négatif. Ainsi, le coût réduit d'un chemin réalisable P est donné par

$$\bar{c}_P = 1 - \sum_{i \in P} \frac{l_i}{L} - \sum_{i \in P} \varepsilon_i.$$

Donc, si on résout le problème de plus court chemin en éliminant $1 - \sum_{i \in P} \frac{l_i}{L}$ de \bar{c}_P , la corrélation est éliminée et l'optimisation est plus rapide. Puisque $1 - \sum_{i \in P} \frac{l_i}{L} \geq 0$ pour tout chemin réalisable P , alors si le chemin optimum obtenu vérifie

$$\bar{c}''_{P^*} = - \sum_{i \in P^*} \varepsilon_i \geq 0,$$

tout chemin réalisable vérifie

$$\bar{c}_P \geq 0$$

et le processus de génération de colonnes peut être arrêté. Dans le cas contraire et si

$$\bar{c}''_{P^*} < 0 \text{ et } -\bar{c}''_{P^*} > 1 - \sum_{i \in P^*} \frac{l_i}{L},$$

alors

$$\bar{c}_{P^*} < 0.$$

Alors, le chemin P^* (et éventuellement d'autres chemins) peut être ajouté au problème maître. Autrement, le sous-problème doit être résolu avec l'expression originale du coût pour vérifier l'optimalité.

Chapitre 7

Conclusion

Nous avons étudié le problème de découpe unidimensionnelle qui possède des propriétés intéressantes qui en font un bon problème pour tester l'efficacité des méthodes de résolution. Sa structure est simple et son saut d'intégrité est très petit, mais il présente beaucoup de symétrie qui introduit de l'instabilité dans le processus de résolution. De plus, c'est un problème très fractionnaire dans le sens où pour une certaine valeur entière de l'objectif, il existe beaucoup de solutions fractionnaires qui donnent cette valeur.

Le point de vue tournées de véhicules que nous avons adopté nous a permis de tirer des conclusions sur la résolution des problèmes de tournées par une procédure de génération de colonnes. La méthode de résolution du sous-problème peut favoriser la génération des colonnes ayant une structure particulière, ce qui peut dans certains cas tel que le problème de découpe affecter défavorablement l'efficacité de l'optimisation, car le sous-problème a tendance à générer les colonnes qui contiennent beaucoup d'items.

Pour le branchement, la prise en compte des caractéristiques des items et l'utilisation de diverses techniques pour réduire le problème permettent de réduire la taille de l'arbre de branchement et d'obtenir une solution optimale entière plus rapidement.

Une technique de branchement qui s'est avérée très efficace consiste à sauter la résolution de certains nœuds de l'arbre tout en les créant pour un éventuel retour arrière. Les résultats obtenus sur les instances testées sont impressionnants et suggèrent un développement plus approfondi de cette technique.

Nous avons également montré que l'agrégation des contraintes qui correspondent à des items de longueur égale permet de stabiliser le processus de résolution et d'améliorer nettement les temps de calcul pour la résolution du problème maître restreint. La même remarque a été faite par Vanderbeck (1996) et Valério de Carvalho (1996) concernant la résolution du problème de découpe binaire. Cette approche permet de réduire la dégénérescence du problème dual. Ceci suggère que les méthodes de stabilisation des variables duales peuvent être d'un apport considérable dans le cadre d'une méthode de génération de colonnes. Dans le cas du problème de découpe unidimensionnelle, les valeurs $\frac{1}{L}$ seraient de bonnes valeurs initiales pour une telle méthode (dans le cas des triplets, ces valeurs donnent une solution optimale au problème dual, et pour les autres problèmes pratiques résolus, ces valeurs fournissent une bonne approximation.)

Pour la formulation agrégée, le branchement sur les variables de flot résulte en l'ajout de contraintes dans le problème maître. La structure du sous-problème n'est pas altérée, mais les coûts des arcs sont affectés par ces décisions par l'intermédiaire des variables duales correspondantes. Ces contraintes doivent être judicieusement choisies pour avoir un branchement efficace. Un travail plus approfondi sur les types de décisions, ainsi que l'élimination des redondances et la simplification de l'effet des décisions sur le sous-problème devra être effectué pour avoir une méthode de résolution efficace.

Bibliographie

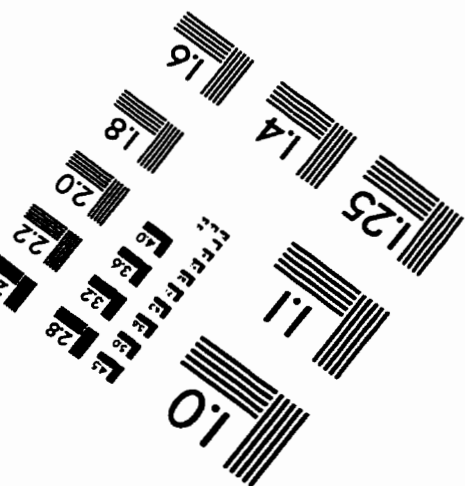
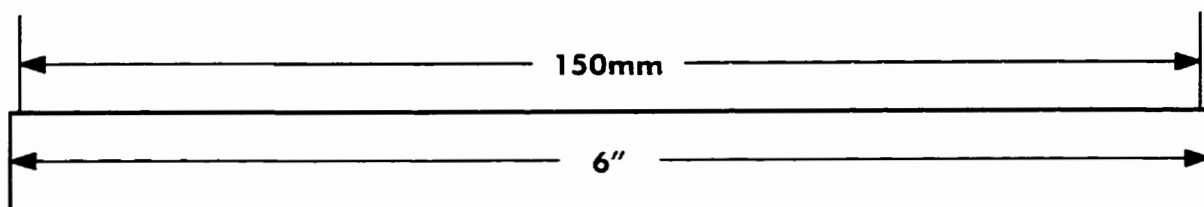
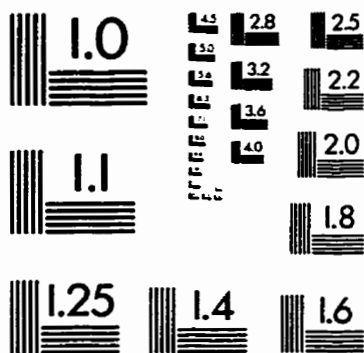
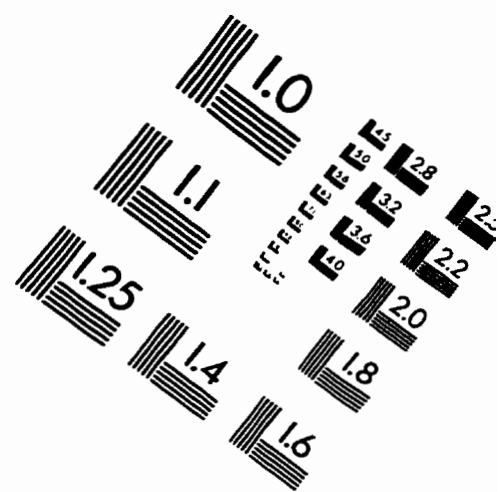
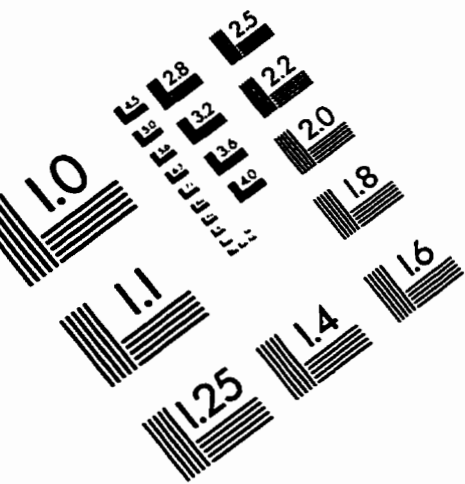
- [1] BEASLEY, J.E. (1990). OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Research Society* Vol. 41, 1069–1072.
- [2] BERGE, C. et JOHNSON, E.L. (1976). *Coloring the Edges of a Hypergraph and Linear Programming Techniques*. Rapport de recherche CORR 76/4, Department of Combinatorics and Optimization, University of Waterloo, Ontario, Canada.
- [3] DANTZIG, G.B. et WOLFE, P. (1960). Decomposition Principle for Linear Programs. *Operations Research* Vol. 8, 101–111.
- [4] 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. Dans G. Laporte (ed.), *Fleet Management and Logistics* (à paraître).
- [5] DESROCHERS, M. et SOUMIS, F. (1988). A Generalized Permanent Labeling Algorithm for the Shortest Path Problem with Time Windows. *INFOR* Vol. 26, 191–212, 1988.
- [6] DESROSIERS, J., DUMAS, Y., SOLOMON, M.M., et SOUMIS, F. (1995). Time Constrained Routing and Scheduling. Dans M.O. Ball *et al.* (ed.), *Network Routing*, Handbooks in Operations Research and Management Science 8. Elsevier Science, Amsterdam, 35–139.

- [7] DYCKHOFF, H. (1981). A New Linear Programming Approach to the Cutting Stock Problem. *Operations Research* Vol. 29, No 6, 1092–1104.
- [8] EILON, S. et CHRISTOFIDES, N. (1971). The Loading Problem. *Management Science* Vol. 17, No 5, 259–268.
- [9] FARLEY, A.A. (1990). A Note on Bounding a Class of Linear Programming Problems, Including Cutting Stock Problems. *Operations Research* Vol. 38, 922–923.
- [10] GILMORE, P.C. et GOMORY, R.E. (1961). A Linear Programming Approach to the Cutting Stock Problem. *Operations Research* Vol. 9, No 6, 849–859.
- [11] GILMORE, P.C. et GOMORY, R.E. (1963) A linear Programming Approach to the Cutting Stock Problem: Part II. *Operations Research* Vol. 11, No 6, 863–888.
- [12] GOULIMIS, C. (1990). Optimal Solution for the Cutting Stock Problem. *European Journal of Operational Research* Vol. 44, 197–208.
- [13] HAESSLER, R.W. (1971). A Heuristic Programming Solution to a Nonlinear Cutting Stock Problem. *Management Science* Vol. 17, No 12, B793–B802.
- [14] HAESSLER, R.W. et SWEENEY, P.E. (1990). Cutting Stock Problems and Solution Procedures. *European Journal of Operational Research* Vol. 54, 141–150.
- [15] KANTOROVICH, L.V. (1960). Mathematical Methods of Organizing and Planning Production. *Management Science* Vol. 6, No 4, 366–393.

- [16] MARCOTTE O. (1985). The Cutting Stock Problem and Integer Rounding. *Mathematical Programming* Vol. 33, 89–92.
- [17] MARCOTTE O. (1986). An Instance of the Cutting Stock Problem for which the Rounding Property Does Not Hold. *Operations Research Letters* Vol. 4, No 5, 239–243.
- [18] MARTELLO, S. et TOTH, P. (1990). *Knapsack Problems: Algorithms and Computer Implementations*. Wiley-Interscience in Discrete Mathematics and Optimization.
- [19] PAQUETTE, V. (1996). *Le problème de découpe binaire*. Mémoire de Maîtrise, École des Hautes Études Commerciales, Montréal, Canada.
- [20] ROODMAN, G.M. (1986). Near Optimal Solutions to One-Dimensional Cutting Stock Problems. *Computers and Operations Research* Vol. 13, 713–719.
- [21] SIMCHI-LEVI, D. (1994). New Worst-case Results for the Bin-packing Problem. *Naval Research Logistics* Vol. 41, 579–585.
- [22] STADTLER, H. (1990). A One-Dimensional Cutting Stock Problem in the Aluminium Industry and its Solution. *European Journal of Operational Research* Vol. 44, 209–223.
- [23] SWEENEY, P.E. et PATERNOSTER, E. (1991). *Cutting and Packing Problems: an Updated Literature Review*. Rapport de recherche #654. The Division of Research, University of Michigan, School of Business Administration, Ann Arbor, Michigan 48109-1234.

- [24] VALÉRIO de CARVALHÔ, J.M. (1996). *Exact Solution of Bin-Packing Problems Using Column Generation and Branch-and-Bound*. Rapport de recherche. Dept. Produção e Sistemas, Universidade de Minho, 4709 Braga Codex, Portugal.
- [25] VANCE, P.H., BARNHART, C., JOHNSON, E.L. et NEMHAUSER, G.L. (1994). Solving Binary Cutting Stock Problems by Column Generation and Branch-and-Bound. *Computational Optimization and Applications* Vol. 3, 111-130.
- [26] VANCE, P.H. (1996). *Branch-and-Price Algorithms for the One-Dimensional Cutting Stock Problem*. Rapport de recherche, Department of Industrial Engineering, Auburn University, Auburn, Alabama 36849-5364.
- [27] VANDERBECK, F. (1995). *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.
- [28] VANDERBECK, F. (1996). *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.

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

