



Problèmes de placement 2D et application à l'ordonnancement : modélisation par la théorie des graphes et approches de programmation mathématique

Cédric Joncour

► To cite this version:

Cédric Joncour. Problèmes de placement 2D et application à l'ordonnancement : modélisation par la théorie des graphes et approches de programmation mathématique. Recherche opérationnelle [math.OC]. Université Sciences et Technologies - Bordeaux I, 2011. Français. NNT : . tel-00661534

HAL Id: tel-00661534

<https://theses.hal.science/tel-00661534>

Submitted on 19 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

N ° d'ordre: 4173



THÈSE

présentée à



L'UNIVERSITÉ DE BORDEAUX I

Ecole doctorale de Mathématiques et Informatique de Bordeaux

par

Cédric JONCOUR

pour obtenir le grade de

DOCTEUR

SPÉCIALITÉ MATHÉMATIQUES APPLIQUÉES

*Problèmes de placement 2D et application à l'ordonnancement :
modélisation par la théorie des graphes et approches de
programmation mathématique*

Soutenue le 14 décembre 2010 à l'Institut de Mathématique de Bordeaux

Après avis de :

M. Mourad BAIYOU,	CR	LIMOS, Clermont Ferrand
M. Sylvain GRAVIER,	DR	Université de Grenoble
M. Frédéric MESSINE,	MCF	Université de Toulouse

Devant la commission d'examen:

M. Sylvain GRAVIER,	DR	Université de Grenoble	Rapporteur
M. Frédéric MESSINE,	MCF	Université de Toulouse	Rapporteur
M. Andrew MILLER,	PR	Université de Bordeaux	Président
M. Arnaud PECHER,	PR	Université de Bordeaux	Directeur
M. Francis SOURD,	Res	SNCF Innovation et Recherche	Examineur
M. François VANDERBECK,	PR	Université de Bordeaux	Directeur

Problèmes de placement 2D et application à l'ordonnancement : modélisation par la théorie des graphes et approches de programmation mathématique

Résumé

Le problème de placement sur deux dimensions consiste à décider s'il existe un rangement d'objets rectangulaires dans une boîte donnée. C'est un problème combinatoire difficile (à la complexité du respect des capacités s'ajoute celle du positionnement des objets). Dans cette thèse, nous considérons les variantes sans rotation des objets et avec ou sans optimisation de la valeur des objets placés.

Nous menons une étude exploratoire des méthodologies qui peuvent être développées à l'interface de la programmation mathématique, de l'optimisation combinatoire et de la théorie des graphes. Notre objectif est aussi de développer des approches non basées sur une discrétisation de la boîte, les plus performantes à l'heure actuelle.

Dans ce mémoire, nous effectuons d'abord une étude théorique des qualités de bornes qui peuvent être obtenues avec les différentes formulations classiques. Au cours de cette étude, nous renforçons certaines de ces formulations et en proposons de nouvelles formulations. Une étude qualitative des bornes issues de la relaxation linéaire des formulations testés sur des jeux d'instances classiques de la littérature confirme l'étude théorique. Cette étude permet de se rendre compte des facteurs déterminant la qualité des bornes et des enjeux à relever par la programmation mathématique.

Par la suite, nous avons développé et testé deux approches de résolution innovantes. L'une est basée sur la décomposition de Dantzig-Wolfe associée à un branchement sur les contraintes disjonctives de non recouvrement des objets. Cette approche a permis une amélioration des résultats obtenus par la programmation mathématique.

L'autre approche constitue une approche combinatoire basée sur diverses caractérisations des graphes d'intervalles (modélisant le chevauchement des objets selon leur projection sur chaque axe). Un premier algorithme est basé sur l'énumération de matrices de uns-consécutifs. Un autre utilise des arbres étiquetés pour éliminer plus efficacement les cas de symétries entre placements. Ces approches ont l'avantage de ne pas dépendre d'une discrétisation du conteneur.

Mots-clefs : problèmes de placement, théorie des graphes, génération de colonnes, modèles mathématiques, étude de branchement, graphes d'intervalles

2D-orthogonal packing and scheduling problems: modelling by graph theory and mathematical approach

Abstract

The two dimensional orthogonal packing problem consists in deciding whether there exists a packing of rectangular items in a given bin. This is a hard combinatorial problem (in addition to capacity constraints, one has to face the complexity of item positionning). In this thesis, we consider the case without item rotation and with or without packing value optimization.

We explore methodologies at the interface of mathematical programming, combinatorial optimization and graph theory. Our aim is also to develop approaches not based on a bin discretization (i.e. an alternative to such methods that are currently the most effective).

In this work, we perform a theoretical study of the quality of bounds of different classical formulations. We tighten some formulations and we propose new formulations. We perform a numerical study to test bound quality on classical instances. This study permits to identify the determinant factor in the quality of mathematical programming formulations.

We develop and test two resolution approaches. The first is based on Dantzig-Wolfe decomposition associated with a branching on no-overlapping disjunctive constraints. This approach permits to improve results obtained by mathematical programming.

The second approach establish a combinatorial approach based on multiple interval graph characterization (modelling the item no-overlapping according to their projection on each axis). The first algorithm is based on consecutive ones matrices enumeration. An other use labelled tree to eliminate more efficiently symmetry in packing. These approaches have to advantage of being independent from bin discretization.

Keywords : orthogonal packing problems, graph theory, column generation, mathematical models, branching study, interval graph



Thèse soutenue le 14 décembre 2010

Institut de Mathématiques de Bordeaux
Université Bordeaux 1
351, cours de la Libération
33405 TALENCE cedex

Ecole doctorale de Mathématiques et
Informatique de Bordeaux
U.F.R. de Mathématiques et Informatique
Bâtiment A33

Remerciements

Voici les lignes que tout doctorant rêve d'écrire dès le début de sa thèse et aujourd'hui c'est mon tour. Et c'est avec la plus grande joie que j'écris ces quelques mots.

Je tiens en premier lieu à exprimer toute ma gratitude à mes deux directeurs de thèse, Arnaud Pêcher et François Vanderbeck, qui ont su m'apporter leur confiance et sans qui ces travaux n'auraient pas été réalisés. Je vous remercie pour tous les conseils, le soutien et la bonne humeur apportés au quotidien durant ces trois dernières années. Je vous suis totalement reconnaissant.

Je remercie aussi Mourad Baiou, Frédéric Messine et Sylvain Gravier pour le temps passé à rapporter cette thèse. Leurs remarques constructives m'ont permis d'améliorer ce mémoire. Je vous témoigne ma plus grande reconnaissance.

J'ai été honoré par la présence à ma soutenance de Francis Sourd. Je le remercie d'avoir accepté d'examiner ma thèse en intégrant mon jury.

Je remercie grandement Andrew Miller d'avoir accepté de présider mon jury de thèse. J'en profite également pour remercier chaudement les personnes composant mon équipe de recherche. Leurs lumières m'ont permis de développer mes connaissances dans le domaine de la recherche. Je remercie plus particulièrement Pierre Pesneau qui m'a prodigué de nombreux conseils au cours de ma thèse.

Je remercie aussi mes collègues chercheurs membres de l'IMB, du Labri et de l'INRIA. Ce fut un plaisir de travailler à vos côtés. Je tiens à remercier plus particulièrement mes amis doctorants qui m'ont apporté beaucoup. Je pense plus particulièrement à Damiano, Jean-Baptiste, Johanna, Joyce, Karen, Marco, Michele, Peng, Vanessa et Xiong. Je remercie aussi les non-doctorants Adrien, Céline, Frédéric, Franck ainsi qu'Eva. En écrivant ces lignes, je repense aux nombreux fous-rires qui ont jallonné notre parcours. Je remercie aussi Petru avec qui ce fut un plaisir de travailler et Pascal dont la collaboration au bureau de l'association des doctorants fut fort agréable. Enfin, je remercie de tout mon cœur les collègues de bureau que j'ai eu la chance de cotoyer durant ces années. Merci Aurélie, Jade et Ludivine... Les moments passés ensemble resteront graver dans ma mémoire.

Mes remerciements s'adressent aussi aux secrétaires de l'IMB, de l'INRIA et de l'UFR qui m'ont toujours apporté leur soutien et leur bonne humeur. En particulier, Annie, Brigitte, Catherine, Karine, Marie Christine, Nathalie et Patricia. Je remercie aussi les ingénieurs que j'ai tant de fois dérangés. Merci Christian, Jacques, Kodor, Philippe, Rémi, Sandrine et Sylvain. Enfin, je remercie les personnes responsables de l'entretien de l'IMB et les personnels du domaine du haut-carré pour leurs gentilleses.

Je tiens à adresser mes plus chaleureux remerciements à mes amis de master, Amélie, Céline et Suzy qui m'ont apporté un soutien constant au cours de cette thèse.

Pour finir, je remercie ma famille. Rien n'est possible sans vous.

Merci à tous et à bientôt sous d'autres horizons!! ☺

« Soit A un succès dans la vie. Alors $A = x y z$,
où $x = \text{travailler}$, $y = \text{s'amuser}$, $z = \text{se taire}$ ».

[EINSTEIN]

Voici la solution obtenue au cours de ma thèse

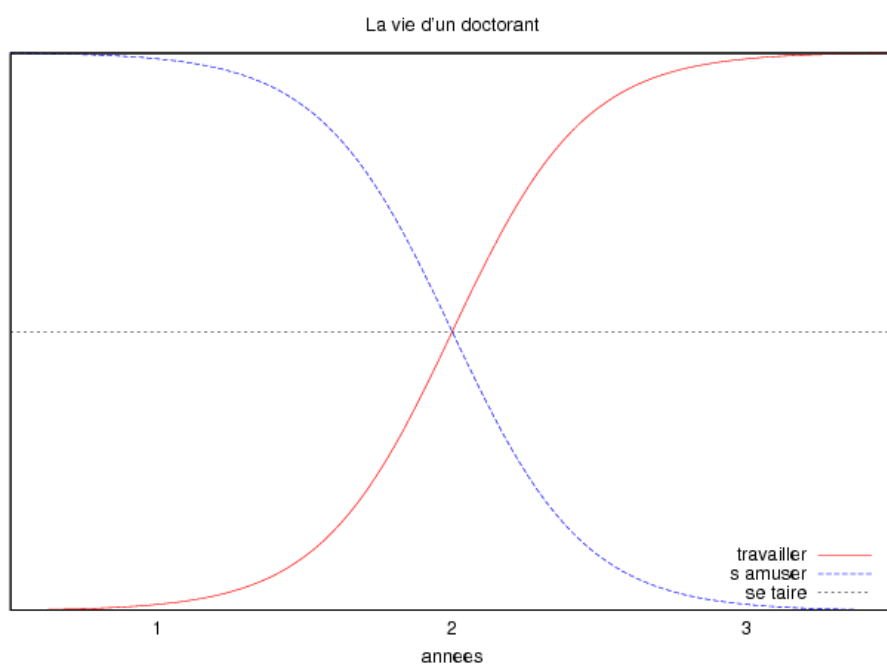


Table des matières

THESE

Introduction	15
Chapitre I Combinatoire des problèmes de placement	19
1.1 Les problèmes de placement	20
1.1.1 Le problème décisionnel traité	20
1.1.2 Les problèmes d'optimisation	23
1.2 La théorie des graphes	24
1.2.1 Définitions générales	24
1.2.2 Les graphes d'intervalles	26
1.2.2.1 Principales caractérisations des graphes d'intervalles . . .	27
1.2.2.2 Les matrices des uns-consécutifs	28
1.2.2.3 Les MPQ-arbres	28
1.3 L'algorithme de Banch-and-Price	31
1.4 Les différentes modélisations	33
1.4.1 Modélisation par la programmation mathématique	33
1.4.2 Modélisation par la théorie des graphes	35
1.4.3 Approches par des algorithmes combinatoires et/ou par la programmation par contraintes	37
1.4.4 Bornes duales	38

1.4.5	Instances des problèmes de sac-à-dos sur deux dimensions résolues dans ce mémoire	39
-------	---	----

Chapitre II	Formulations par programmation mathématique	43
--------------------	--	-----------

2.1	Modélisation utilisant une position relative des objets	44
2.1.1	Lexique des variables pour les formulations définissant une position relative entre les objets	44
2.1.2	Formulation définissant une position relative entre les objets	44
2.2	Modélisations basées sur une discrétisation de l'espace	46
2.2.1	Lexique des variables pour les formulations définissant les coordonnées selon une discrétisation de l'espace	47
2.2.2	Formulation avec des variables indiquant la coordonnée de positionnement de chaque objet sélectionné	48
2.2.3	Formulation avec des variables indépendantes indiquant la coordonnée de positionnement de chaque objet sélectionné	50
2.2.4	Formulation avec des variables séparant les décisions sur les abscisses et les ordonnées de chaque objet sélectionné	59
2.3	Modélisation basée sur un découpage par cliques maximales d'un placement	61
2.3.1	Lexique des variables pour les formulations définissant un découpage par des cliques maximales d'un placement	61
2.3.2	Formulation définissant un découpage par cliques d'un placement	63
2.4	Décomposition des formulations	73
2.4.1	Lexique des variables pour les décompositions des formulations utilisant une discrétisation de l'espace	73
2.4.2	Décomposition de la formulation $F^{x_{ia}y_{ib}}$	75
2.4.3	Décomposition de la formulation $F_{relax}^{x_{ia}y_{ib}}$	77
2.4.4	Lexique des variables pour les décompositions des formulations compactes	79
2.4.5	Décomposition de la formulation $F_{relax}^{w^k h^l}$	80
2.5	Comparaison entre les différentes approches	81

Chapitre III	Implémentation de contraintes disjonctives par voie de branchement	87
---------------------	---	-----------

3.1	Couplage des bandes horizontales et verticales	88
3.1.1	Propriété de non intersection des objets	89
3.1.1.1	Stratégie 1 : Vérification de la contrainte non-intersection pour l'ensemble des bandes	90
3.1.1.2	Stratégie 2 : Vérification de la contrainte non-intersection pour un sous-ensemble de bandes	97
3.2	Résultats numériques	99

Chapitre IV	Résolution par approches combinatoires	103
--------------------	---	------------

4.1	Le problème de placement multi-dimensionnel	104
4.2	Approche par génération de matrices de uns-consécutifs	105
4.2.1	Décomposition par bandes d'un placement	105
4.2.2	L'algorithme	109

4.2.3	Améliorations des performances de l'algorithme	110
4.2.3.1	Détection d'irréalissabilité	110
4.2.3.2	Elimination de décompositions par bandes équivalentes . .	114
4.3	Approche par génération de MPQ-arbres	121
4.3.1	L'algorithme de Korte et Möhring	121
4.3.2	Application aux problèmes de placement	123
4.3.3	Le cœur de l'algorithme de contrôle de la réalisabilité d'un ensemble d'objets	128
4.3.4	Améliorations des performances de l'algorithme	129
4.3.4.1	Elimination des cas dégénérés	129
4.3.4.2	Ajout d'inégalités valides pour détecter des irréalissabilités au plus tôt	132
4.3.4.3	Elimination de quelques symétries	132
4.4	Résultats numériques	133
Conclusion		137
Bibliographie		139

ANNEXES

Annexe A	Power Plant Production Planning	147
1	Formulation	148
1.1	Notation	148
1.2	Variables	149
1.2.1	Decision Variables	149
1.2.2	Required State Variables (resulting from the above decisions)	150
1.2.3	Redundant State Variables (resulting from the above decisions)	150
1.3	Constraints	150
1.4	Objective	156
2	A 2-Stage Hierarchical Optimisation Approach	157
2.1	Stage 1 - phase A: a Dantzig-Wolfe decomposition approach	157
2.1.1	The master	157
2.1.2	The pricing subproblem	158
2.1.3	Alternative Column Definition	161
2.2	Stage 1: an extended formulation	161
2.2.1	State and Transition Variables	161
2.2.2	Formulation	162
2.3	Stage 2: in one pass	164
2.3.1	Variables	164
2.3.2	Formulation	165
3	Selected Approach	166

Annexe B	Column Generation based Primal Heuristics	167
1	Introduction	168
2	An overview of column generation based heuristics	168
3	Diving heuristics	169
4	Computational results	171
	Bibliography	172

*

* *

THESE

Introduction

Les problèmes consistant à placer des objets sans chevauchement dans un conteneur sont des problèmes connus dans la littérature sous le nom de « orthogonal packing problems ». Différentes applications comme la découpe de matériaux (textile, bois, métaux), l'ordonnancement de projets ou tout simplement le rangement de cartons sur une palette montrent l'intérêt d'étudier de tels problèmes. Dans notre étude, la rotation des objets est interdite.

Dès 1961, Gilmore et Hoffman [38, 39] modélisèrent la variante découpe du problème (« cutting stock problem »). Ceci constitue la première définition d'un problème de découpe. Cependant, il s'agissait d'un problème unidimensionnel. La première définition du problème de placement sur deux dimensions fut réalisé en 1985 par Beasley [4]. Le problème de placement sur deux dimensions est NP-difficile au sens fort car le problème de « bin-packing » est un cas particulier du problème de placement sur deux dimensions. A la complexité du respect des capacités s'ajoute celle du positionnement des objets et des symétries qu'il existe entre les différents placements. Beasley proposa une modélisation utilisant une discrétisation de l'espace. Par la suite, de nombreuses modélisations du problème de placement ont été proposés ; la plupart utilisant une discrétisation du conteneur. Une autre approche a été développée par Fekete et Schepers dès 1997 [27, 32]. Elle se base sur une représentation des classes de placements par des graphes.

Les travaux de cette thèse ont été motivés par le manque d'approches combinant l'approche de programmation mathématique et la modélisation par les graphes d'un placement. Notre objectif est aussi de développer des approches non basées sur une discrétisation de la boîte.

Dans le chapitre I, nous présentons la définition du problème que nous allons étudier.

Nous rappelons aussi des définitions classiques de théorie des graphes et présentons l'approche de « Branch-and-Price ». Nous faisons une revue de la littérature des différentes approches développées pour le problème de placement sur deux dimensions et présentons les instances de la littérature sur lesquelles nous testons nos approches.

Dans le chapitre II, nous répertorions et étudions l'ensemble des formulations proposées dans la littérature. Cette étude nous a permis d'ajouter de nouvelles inégalités valides qui améliorent les bornes obtenues à partir de ces formulations. De plus, une nouvelle formulation compacte est proposée. Elle est basée sur l'approche de modélisation par les graphes. La deuxième partie du chapitre reprend les formulations étendues obtenues à partir des formulations précédemment définies. De nouvelles approches de décomposition de Dantzig-Wolfe sont aussi développées. Des relations de dominance sont alors établies entre les formulations. Les bornes duales qui peuvent être obtenues en résolvant la relaxation linéaire de ces formulations sont comparées aux bornes de la littérature. En faisant cette étude, nous mettons en évidence les avantages et les inconvénients des différentes formulations en particulier en comparant les qualités des relaxations. Cela nous permet de faire une analyse des facteurs déterminant la qualité des bornes duales obtenues par la programmation mathématique. A notre connaissance, aucune étude comparée des formulations du problème de placement sur deux dimensions n'était aussi complète que la nôtre. Ces travaux ont été présentés à la conférence de la ROADEF en 2008 [46].

Au chapitre III, nous nous concentrons sur une reformulation de Dantzig-Wolfe et une approche de génération de colonnes avec des problèmes de sac-à-dos comme sous-problème. L'originalité de l'approche développée réside dans l'ajout de coupes de réalisabilité pour imposer la non-superposition des objets. Nous implémentons des contraintes disjonctives par voie d'un schéma de branchement dans le cadre d'un algorithme de « Branch-and-Price ». Les contraintes de branchement sont implémentées dans les sous-problèmes de « pricing » en partitionnant l'ensemble des solutions aux sous-problèmes. Ceci évite de modifier la structure du problème de sac-à-dos. L'ajout des coupes de réalisabilité induit une amélioration des bornes duales.

Dans le chapitre IV, de nouvelles approches de résolution basées sur la caractérisation des placements par les graphes sont proposées. La première utilise une caractérisation par les matrices de uns-consécutifs. L'avantage de cette approche est qu'elle est facile à implémenter car elle utilise comme seule structure de données des matrices. La deuxième approche utilise une caractérisation des graphes d'intervalles par des arbres étiquetés. Contrairement à l'approche précédente, la structure de donnée est plus difficile à mettre en œuvre. En revanche, elle est plus efficace dans son traitement des cas symétriques. De plus, ces approches ont aussi l'avantage de ne pas dépendre de la taille de la boîte qui représente un des principaux inconvénients des autres approches combinatoires de la littérature. Ces nouveaux algorithmes nous ont permis d'obtenir des temps de calculs compétitifs comparés aux approches existantes. Ces travaux ont été présentés à plusieurs conférences dont ISMP 2009 [44] et ISCO 2010. Ils ont fait l'objet de deux publications dans *Electronics Notes in Discrete Mathematics* [45, 47] et ont été aussi soumis à la revue « *Journal of Mathematical Modelling and Algorithms* » dans leur version longue.

En annexe, nous présentons d'autres travaux effectués durant ma thèse indépendamment du sujet original. Tout d'abord, nous avons participé au challenge ROADEF 2010

[1] dont l'objectif était de résoudre problème de planification de la production de grande taille comportant des contraintes diversifiées. Le problème consiste à programmer les maintenances dans les centrales nucléaires sur un horizon de temps de 5 ans pour un ensemble de scénarios probabilistes. La première annexe expose l'analyse du problème et l'approche de résolution que nous avons mis en œuvre. Nous avons utilisé une approche à deux niveaux pour résoudre le problème. Le premier niveau consiste à agréger l'horizon de temps (en semaines) et les demandes des scénarios (demande moyenne par semaine). Ce niveau permet de fixer les périodes de production et de maintenance ainsi que la quantité de combustible à recharger. Le deuxième niveau consiste à déterminer pour chaque scénario la production effective pour chaque période de temps. Chaque niveau est modélisé sous la forme d'un programme linéaire en nombre entier et est passé au solveur CPLEX 11 avec un temps maximal de résolution fixé. Les résultats obtenus nous ont permis de nous classer parmi les meilleures équipes lors de la phase de qualification. J'ai été personnellement très impliqué dans l'implémentation et la mise au point de l'approche de résolution.

La deuxième annexe est une étude sur les méthodes d'heuristiques primales génériques dans le cadre d'une approche de génération de colonnes. Nous avons développé des heuristiques d'arrondi et de plongeon avec des mécanisme de diversification de la recherche de type « least discrepancy search ». Ces procédures peuvent se comprendre comme des recherches heuristiques dans un arbre de Branch-and-Price. Cette méthode nous permet d'obtenir des solutions valides d'une qualité similaire à des approches méta-heuristiques. Ces travaux ont été exposés à la conférence ISCO 2010 et ont fait l'objet d'une publication dans *Electronics Notes in Discrete Mathematics* [43].

*
* *

Sommaire

1.1	Les problèmes de placement	20
1.2	La théorie des graphes	24
1.3	L'algorithme de Banch-and-Price	31
1.4	Les différentes modélisations	33

CHAPITRE I

Combinatoire des problèmes de placement

Les problèmes de découpage de matériaux et de placement sur deux dimensions sont des problèmes classiques de la littérature [25, 62]. Il consiste à placer un ensemble de boîtes rectangulaires (ou objets) de taille donnée dans un conteneur de taille fixe de telle manière que deux objets ne se chevauchent pas et qu'ils ne dépassent pas la taille du conteneur. Les applications sont nombreuses dans l'industrie (textile, bois, métallurgie) mais aussi dans des problèmes d'ordonnancement de tâches et/ou de ressource.

De nombreux problèmes différents peuvent être étudiés. Cependant, nous nous intéressons à l'ensemble des problèmes de placement dit « orthogonaux » avec une contrainte interdisant la rotation des objets à placer. Nous étudions le cas des problèmes de placement sur deux dimensions même si une généralisation à D dimensions est faite au chapitre IV. Ces problèmes sont connus pour être des problèmes NP-difficiles au sens fort car les problèmes d'ordonnancement ou de « bin packing » peuvent être vus comme des cas particuliers de problèmes de placement [37]. De plus, les symétries liées aux nombreuses possibilités de positionnement des objets sont une autre source de difficulté. En effet, en appliquant des permutations entre les positions, nous obtenons différents placements possibles pour un même ensemble d'objets.

Dans un premier temps, nous définissons le problème décisionnel de placement. Ensuite, nous présentons les différents problèmes de découpage et de placement pouvant être résolus. Par la suite, nous faisons des rappels sur la théorie des graphes et sur les caractérisations des graphes d'intervalles. Nous effectuons aussi un rappel sur les approches de décomposition. Enfin, nous rappelons les différentes modélisations et approches développées dans la littérature pour résoudre ce problème.

1.1 Les problèmes de placement

Nous commençons par présenter et définir le problème que nous allons étudier durant ce mémoire.

1.1.1 Le problème décisionnel traité

Soit n le nombre d'objets rectangulaires de taille donnée à ranger. Notons $\mathcal{I} = \{1, \dots, n\}$ l'ensemble d'objets tel que chaque objet $i \in \mathcal{I}$ a une taille fixe $(\mathbf{w}_i, \mathbf{h}_i) \in \mathbb{R}_+^{*2}$. Et soit un conteneur de taille $(\mathbf{W}, \mathbf{H}) \in \mathbb{R}_+^{*2}$. Le problème de placement sur deux dimensions (« two dimensional orthogonal packing problem ») consiste à déterminer si l'ensemble des objets $\mathcal{I}$ peut être rangé dans le conteneur sans chevauchement des objets les uns par rapport aux autres, ni dépassement de la taille du conteneur. De plus, la rotation des objets est interdite. Ainsi, l'ensemble $\mathcal{I}$ est réalisable si, pour chaque objet $i \in \mathcal{I}$, nous associons une coordonnée $(x_i, y_i) \in [0, \mathbf{W}] \times [0, \mathbf{H}]$ indiquant la position du coin inférieur-gauche de l'objet i dans le conteneur et vérifiant certaines contraintes décrites ci-après (cf. figure 1.1).

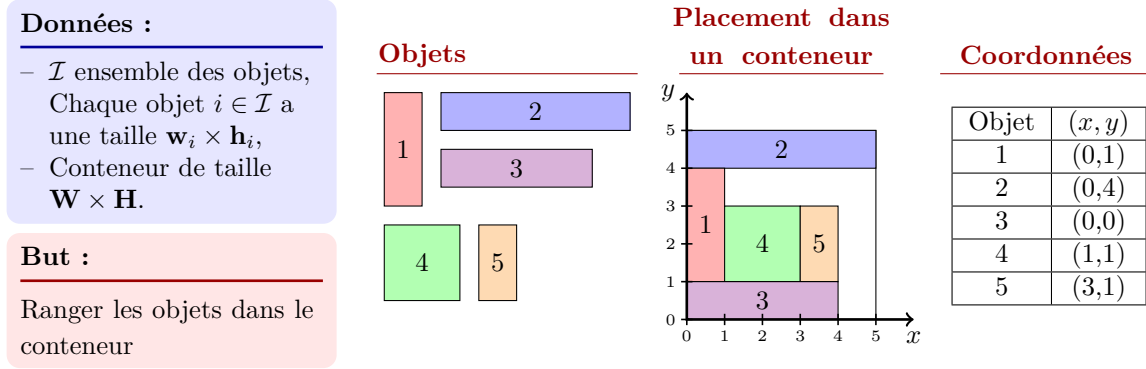


FIGURE 1.1: Illustration d'un placement sur deux dimensions avec un système de coordonnées vérifiant les conditions (1.1a)-(1.1b)

Définition 1.1

Soit $\mathcal{I}$ un ensemble d'objets et les vecteurs de taille $(\vec{\mathbf{w}}, \vec{\mathbf{h}})$. Et soit un conteneur de taille $(\mathbf{W}, \mathbf{H})$. L'ensemble $\mathcal{I}$ est dit réalisable si et seulement s'il existe une coordonnée $(x_i, y_i) \in [0, \mathbf{W}] \times [0, \mathbf{H}]$ associée à chaque objet $i \in \mathcal{I}$ vérifiant les conditions suivantes :

$$\forall i \in \mathcal{I} : x_i + \mathbf{w}_i \leq \mathbf{W} \quad \text{et} \quad y_i + \mathbf{h}_i \leq \mathbf{H}, \quad (1.1a)$$

$$\forall i, j \in \mathcal{I} (i \neq j) : [x_i, x_i + \mathbf{w}_i[\cap [x_j, x_j + \mathbf{w}_j[= \emptyset \quad \text{ou} \quad [y_i, y_i + \mathbf{h}_i[\cap [y_j, y_j + \mathbf{h}_j[= \emptyset. \quad (1.1b)$$

Nous disons alors que l'ensemble $\mathcal{I}$ peut être placé dans le conteneur. Et, pour chaque ensemble de coordonnées $(x_i, y_i)_{i \in \mathcal{I}}$ réalisable associé à $\mathcal{I}$, nous parlons de placement réalisable sur deux dimensions. ♦

La condition (1.1a) impose que chaque objet ne dépasse pas la taille du conteneur en fonction de son positionnement dans la boîte. La condition (1.1b) impose que chaque paire d'objets soit disjointe sur au moins une des deux dimensions. Donc, les objets ne se chevauchent pas.

Remarque La condition (1.1b) revient à vérifier qu'au moins l'une des quatre conditions suivantes est vérifiée pour chaque paire d'objets $i, j \in \mathcal{I} (i \neq j)$:

- $x_i + \mathbf{w}_i \leq x_j$,
- $x_j + \mathbf{w}_j \leq x_i$,
- $y_i + \mathbf{h}_i \leq y_j$,
- $y_j + \mathbf{h}_j \leq y_i$.

Donc, chaque paire d'objets a une distance suffisante entre leurs coordonnées sur au moins une dimension pour éviter un chevauchement. $\blacklozenge$

Définition 1.2

Un placement sur deux dimensions vérifie la condition « calée en bas à gauche » si chaque objet $i \in \mathcal{I}$ est immédiatement à la droite d'un autre objet ou touche le bord gauche du conteneur selon l'axe des abscisses :

$$x_i \in \{0\} \cup \{x_j + \mathbf{w}_j : j \in \mathcal{I} \setminus \{i\}\}, \quad \forall i \in \mathcal{I}.$$

Et si chaque objet $i \in \mathcal{I}$ est immédiatement au dessus d'un autre objet ou touche le bord inférieur du conteneur selon l'axe des ordonnées :

$$y_i \in \{0\} \cup \{y_j + \mathbf{h}_j : j \in \mathcal{I} \setminus \{i\}\}, \quad \forall i \in \mathcal{I}.$$

$\blacklozenge$

Lemme 1.3

Pour chaque ensemble $\mathcal{I}$ réalisable, il existe un placement associé vérifiant la condition « calée en bas à gauche » sur chaque dimension.

Preuve. Pour tout $i \in \mathcal{I}$, soit $(x_i, y_i) \in [0, \mathbf{W}[\times [0, \mathbf{H}[$ un placement réalisable associé à l'ensemble d'objets $\mathcal{I}$ ne satisfaisant pas la condition « calée à gauche ». Donc, il existe un objet $i \in \mathcal{I}$ tel que :

$$x_i \notin \{0\} \cup \{x_j + \mathbf{w}_j : j \in \mathcal{I} \setminus \{i\}\} \quad \text{ou} \quad y_i \notin \{0\} \cup \{y_j + \mathbf{h}_j : j \in \mathcal{I} \setminus \{i\}\}$$

Définissons alors un nouveau système de coordonnées $(\tilde{x}_j, \tilde{y}_j)_{j \in \mathcal{I}}$ réalisable vérifiant cette condition.

Cas 1: Supposons que $x_i \notin \{0\} \cup \{x_j + \mathbf{w}_j : j \in \mathcal{I} \setminus \{i\}\}$.

Posons $\tilde{x}_i = \max\{x_j + \mathbf{w}_j : j \in \mathcal{I}, x_j + \mathbf{w}_j \leq x_i\}$ la plus grande abscisse inférieure à x_i tel qu'il existe un objet se terminant sur cette abscisse. Pour chaque objet $i' \in \mathcal{I}$ avec $i' \neq i$, posons $\tilde{x}_{i'} = x_{i'}$. Montrons que $(\tilde{x}_j, y_j)_{j \in \mathcal{I}}$ est un placement réalisable associé à $\mathcal{I}$.

Nous avons $\tilde{x}_i \leq x_i$, et donc $\tilde{x}_i + \mathbf{w}_i \leq x_i + \mathbf{w}_i \leq \mathbf{W}$. De même, pour tout $i' \in \mathcal{I}$ avec $i' \neq i$, $\tilde{x}_{i'} + \mathbf{w}_{i'} = x_{i'} + \mathbf{w}_{i'} \leq \mathbf{W}$.

D'où, il reste à montrer que la condition (1.1b) est vraie. $\forall i', i'' \in \mathcal{I}$ avec $i' \neq i$ et $i'' \neq i$, les coordonnées ne changent pas, d'où la condition (1.1b) reste vraie.

Donc, $\forall i' \in \mathcal{I} (i' \neq i)$, nous devons prouver que :

$$[\tilde{x}_i, \tilde{x}_i + \mathbf{w}_i[\cap [\tilde{x}_{i'}, \tilde{x}_{i'} + \mathbf{w}_{i'}[= \emptyset \quad \text{ou} \quad [y_i, y_i + \mathbf{h}_i[\cap [y_{i'}, y_{i'} + \mathbf{h}_{i'}[= \emptyset.$$

Or, pour tout $i' \in \mathcal{I} (i' \neq i)$,

$$[x_i, x_i + \mathbf{w}_i[\cap [x_{i'}, x_{i'} + \mathbf{w}_{i'}[= \emptyset \quad \text{ou} \quad [y_i, y_i + \mathbf{h}_i[\cap [y_{i'}, y_{i'} + \mathbf{h}_{i'}[= \emptyset.$$

Donc, nous rencontrons alors deux cas :

- Soit $[x_i, x_i + \mathbf{w}_i] \cap [x_{i'}, x_{i'} + \mathbf{w}_{i'}] = \emptyset$. Dans ce cas, $x_{i'} + \mathbf{w}_{i'} \leq x_i$ ou $x_i + \mathbf{w}_i \leq x_{i'}$.
 Si $x_{i'} + \mathbf{w}_{i'} \leq x_i$, alors $x_{i'} + \mathbf{w}_{i'} \leq \max\{x_j + \mathbf{w}_j : j \in \mathcal{I}, x_j + \mathbf{w}_j \leq x_i\} = \tilde{x}_i$.
 Si $x_i + \mathbf{w}_i \leq x_{i'}$, alors $\tilde{x}_i + \mathbf{w}_i \leq x_i + \mathbf{w}_i \leq x_{i'}$.
 - Soit $[y_i, y_i + \mathbf{h}_i] \cap [y_{i'}, y_{i'} + \mathbf{h}_{i'}] = \emptyset$. Dans ce cas, la propriété reste vraie.
- D'où, $[\tilde{x}_i, \tilde{x}_i + \mathbf{w}_i] \cap [\tilde{x}_{i'}, \tilde{x}_{i'} + \mathbf{w}_{i'}] = \emptyset$ ou $[y_i, y_i + \mathbf{h}_i] \cap [y_{i'}, y_{i'} + \mathbf{h}_{i'}] = \emptyset$.

Cas 2: Nous procédons de même pour le cas où $y_i \notin \{0\} \cup \{y_j + \mathbf{h}_j : j \in \mathcal{I} \setminus \{i\}\}$.

Et donc, l'objet i vérifie maintenant la condition « calée en bas gauche ». En itérant de la même manière en considérant à chaque fois l'objet de plus petite coordonnée, nous construisons un placement réalisable associé à $\mathcal{I}$ satisfaisant la condition « calée en bas à gauche » pour tous les objets. $\square$

Définition 1.4

Un placement réalisable *entier* est un placement réalisable selon la définition 1.1 tel que, pour tout objet $i \in \mathcal{I}$, $(x_i, y_i) \in \mathbb{N}^2$. $\blacklozenge$

Lemme 1.5

Si nous avons un ensemble d'objets réalisable $\mathcal{I}$ tel que, pour tout $i \in \mathcal{I}$, $(\mathbf{w}_i, \mathbf{h}_i) \in \mathbb{N}^2$ et un conteneur de taille $(\mathbf{W}, \mathbf{H}) \in \mathbb{N}^2$; alors, il existe au moins un placement réalisable entier.

Preuve. D'après le lemme 1.3, pour chaque ensemble $\mathcal{I}$ réalisable, il existe au moins un placement réalisable avec la condition « calée en bas à gauche ». Supposons que nous avons un tel placement. Montrons que ce placement est forcément entier pour tout $i \in \mathcal{I}$. Supposons qu'il existe au moins un objet $i \in \mathcal{I}$ tel que $x_i \notin \mathbb{N}$ ou $y_i \notin \mathbb{N}$.

Supposons qu'il existe $i \in \mathcal{I}$ tel que $x_i \notin \mathbb{N}$. Alors, il existe un objet $i \in \mathcal{I}$ avec la plus petite abscisse non entière : $x_i = \min_{j \in \mathcal{I}} \{x_j : x_j \notin \mathbb{N}\}$.

Or, d'après la condition « calée en bas à gauche », $x_i \in \{0\} \cup \{x_j + \mathbf{w}_j : j \in \mathcal{I} \setminus \{i\}\}$. Donc, comme $\forall j \in \mathcal{I}$ tel que $x_j < x_i$, nous avons $x_j + \mathbf{w}_j \in \mathbb{N}$; alors $x_i \in \mathbb{N}$. Cela contredit la condition $x_i \notin \mathbb{N}$. Donc, $\forall i \in \mathcal{I}$, $x_i \in \mathbb{N}$.

Nous démontrons de manière similaire que $\forall i \in \mathcal{I}$, $y_i \in \mathbb{N}$.

Donc, un placement avec la condition « calée en bas à gauche » est bien un placement réalisable entier. $\square$

Remarque Tout au long de ce mémoire, les variables suivantes désignent les coordonnées d'un objet $i \in \mathcal{I}$:

$$\left\{ \begin{array}{ll} x_i \in [0, \mathbf{W} - \mathbf{w}_i] \text{ est l'abscisse du coin inférieur-gauche de l'objet } i, & \forall i \in \mathcal{I}, \\ y_i \in [0, \mathbf{H} - \mathbf{h}_i] \text{ est l'ordonnée du coin inférieur-gauche de l'objet } i, & \forall i \in \mathcal{I}. \end{array} \right.$$

Nous introduisons aussi d'autres variables indiquant si deux objets partagent une abscisse x (c'est-à-dire si deux objets s'intersectent selon la verticale) ou une ordonnée y (c'est-à-dire si deux objets s'intersectent selon l'horizontal) :

$$\left\{ \begin{array}{l} x_{ij} = \begin{cases} 1 & \text{si les objets } i \text{ et } j \text{ s'intersectent selon une abscisse } x, \\ 0 & \text{sinon} \end{cases}, \quad \forall i, j \in \mathcal{I} (i < j), \\ y_{ij} = \begin{cases} 1 & \text{si les objets } i \text{ et } j \text{ s'intersectent selon une ordonnée } y, \\ 0 & \text{sinon} \end{cases}, \quad \forall i, j \in \mathcal{I} (i < j). \end{array} \right. \blacklozenge$$

1.1.2 Les problèmes d'optimisation

Définition 1.6

Soit $\mathcal{I}$ un ensemble d'objet. Pour chaque $i \in \mathcal{I}$, nous associons une valeur profit ρ_i . Le problème de sac-à-dos sur deux dimensions consiste à sélectionner un sous-ensemble $\mathcal{I}' \subseteq \mathcal{I}$ tel que $\mathcal{I}'$ est un ensemble réalisable selon la définition 1.1 avec un profit maximal :

$$\max_{\mathcal{I}' \subseteq \mathcal{I}} \left\{ \sum_{i \in \mathcal{I}'} \rho_i : \mathcal{I}' \text{ vérifie les conditions (1.1a) - (1.1b)} \right\}.$$

Le but est donc de trouver l'ensemble $\mathcal{I}'$ avec le meilleur profit (cf figure 1.2). $\blacklozenge$

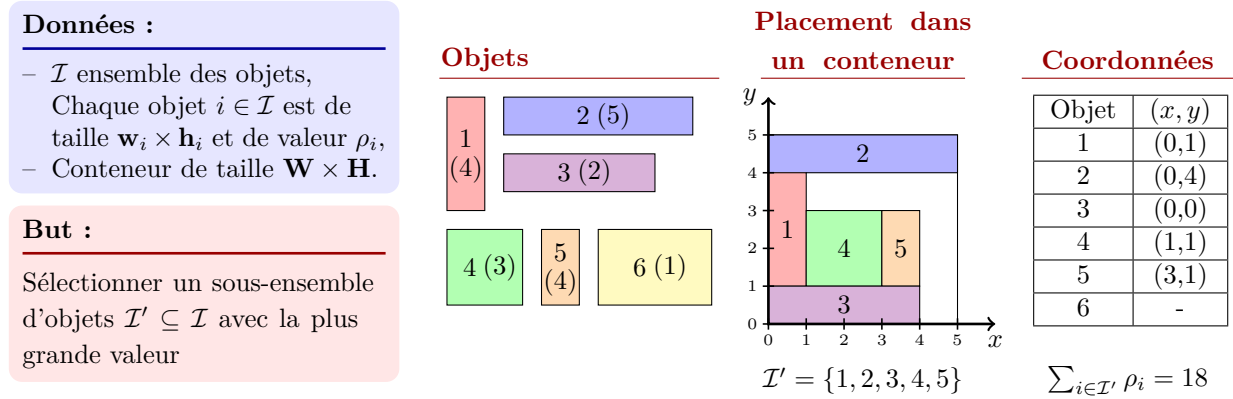


FIGURE 1.2: Illustration d'une solution optimale pour le problème de sac-à-dos sur deux dimensions avec ρ_i entre parenthèses $\forall i \in \mathcal{I}$

Remarque Pour savoir si un objet i est sélectionné, nous utilisons une variable binaire :

$$\left\{ \delta_i = \begin{cases} 1 & \text{si l'objet } i \text{ est sélectionné} \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}. \right.$$

Posons alors $\mathcal{I}' = \{i \in \mathcal{I} : \delta_i = 1\}$. $\blacklozenge$

Définition 1.7

Soit $\mathcal{I}$ un ensemble d'objets. Le problème de « bin packing » sur deux dimensions consiste à déterminer le nombre minimal de conteneurs nécessaires pour ranger l'ensemble des objets tel que chaque conteneur vérifie la définition 1.1 :

$$\min_{B \in \mathbb{N}} \left\{ B : \begin{array}{l} \mathcal{I}_1 \cup \dots \cup \mathcal{I}_B = \mathcal{I}, \mathcal{I}_1 \cap \dots \cap \mathcal{I}_B \neq \emptyset, \\ \mathcal{I}_b \text{ vérifie les conditions (1.1a) - (1.1b), } \forall b \in \{1, \dots, B\} \end{array} \right\}.$$

$\blacklozenge$

D'autres problèmes d'optimisation peuvent être aussi étudiés tel que le problème de « strip packing » sur deux dimensions consistant à calculer la hauteur minimale nécessaire pour ranger l'ensemble des objets disponibles. De plus, certaines contraintes peuvent être ajoutées, comme des contraintes de coupes guillottes ; ou relâchées comme la contrainte de non rotation des objets. Cependant, ces variantes ne sont pas considérées dans notre étude.

1.2 La théorie des graphes

Nous allons maintenant faire quelques rappels généraux sur la théorie des graphes. Puis, nous rappelons les principales caractérisations des graphes d'intervalles dont celle utilisant les matrices de uns-consécutifs et celle utilisant les MPQ-arbres.

1.2.1 Définitions générales

Soit V un ensemble de n sommets. Une arête est une paire de sommets $\{u, v\}$ avec $u, v \in V$ qui sont reliés entre eux et généralement noté uv . Soit E un ensemble d'arêtes. Deux sommets qui sont reliés par une arête sont dits adjacents. Un sommet est dit incident à une arête $uv \in E$ s'il est soit u , soit v . Une boucle est une arête vv avec $v \in V$. Un graphe non orienté (sans boucle), noté G , est défini par une paire (V, E) (cf. figure 1.3). A partir de maintenant, nous supposons que les graphes sont non orientés (sauf précision contraire) et sans boucle.

Nous appelons degré d'un sommet le nombre d'arêtes incidentes à ce sommet. Le degré d'un sommet $u \in V$ est noté $\delta(u)$. Le voisinage d'un sommet u , noté $V(u)$, représente l'ensemble des sommets adjacents à u . Un graphe G de n sommets est complet si toutes les paires de sommets du graphe sont adjacentes. Il est noté en général K_n .

Graphe G :

- $V = \{1, 2, 3, 4, 5\}$,
- $E = \left\{ \{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \right. \\ \left. \{2, 5\}, \{3, 4\}, \{3, 5\} \right\}$.

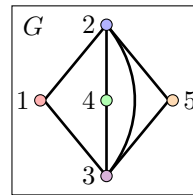


FIGURE 1.3: Exemple de représentation graphique d'un graphe G

Le complémentaire d'un graphe $G = (V, E)$ est un graphe $\overline{G} = (V, \overline{E})$ avec

$$\overline{E} = \{(u, v) \in V^2 : u \neq v \text{ et } uv \notin E\}.$$

Un sous-graphe de $G = (V, E)$ est un graphe $G' = (V', E')$ tel que $V' \subseteq V$ et $E' \subseteq E$. Un sous-graphe de G est induit si $E' = \{uv \in E : u \in V' \text{ et } v \in V'\}$.

Une clique de G est un ensemble de sommets Q de V tel que toute paire de sommets de Q est adjacente. Un ensemble stable (ou stable) de G est un ensemble de sommets S de V tel que son sous-graphe induit complémentaire est une clique. Donc, une clique de G est un

ensemble stable dans $\overline{G}$ et inversement. Une clique de G est maximale s'il n'existe pas de clique de G contenant strictement cette clique. Le stable maximal est un ensemble stable S de G tel qu'il n'existe pas de stable de cardinalité plus grande contenant strictement S (cf. figure 1.4 pour visualiser différents exemples illustrant ces notions).

Notons $\mathcal{Q}$ l'ensemble des cliques maximales d'un graphe G et $\mathcal{S}$ l'ensemble des stables maximaux. Pour tout sommet $u \in V$, notons $\mathcal{Q}(u)$ l'ensemble des cliques maximales contenant le sommet u .

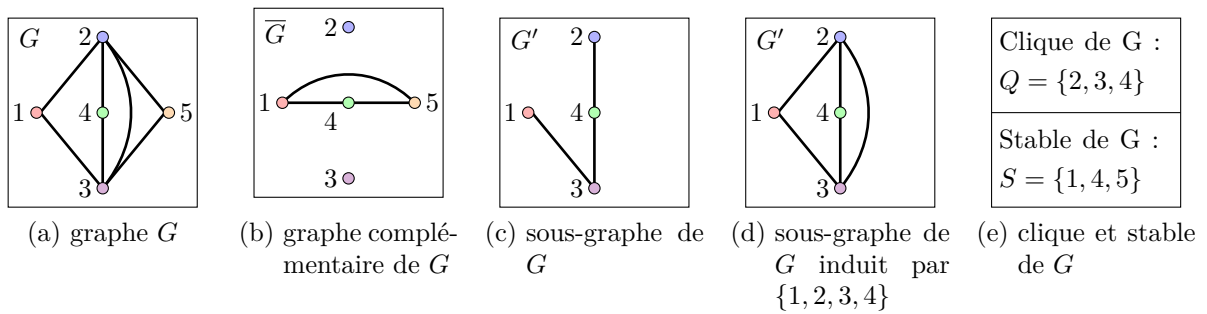


FIGURE 1.4: Quelques exemples de graphes

Un chemin d'un graphe de longueur k est une séquence de sommets $[v_0, \dots, v_k]$ tel que $v_{i-1}v_i \in E$ pour tout $i \in \{1, \dots, k\}$. Un chemin est simple si $v_i \neq v_j$ lorsque $0 \leq i \neq j \leq k-1$. Pour un chemin simple, les arêtes $v_iv_j \in E$ avec $(|i-j| \bmod k) > 1$ sont appelées des cordes. Un chemin simple est sans corde s'il ne possède pas de corde. Un chemin simple sans cordes de n sommets est noté P_n . Un graphe est connexe s'il existe un chemin entre chaque paire de sommets.

Un cycle (ou chemin fermé) de longueur k est une séquence de sommets du graphe $[v_0, v_1, \dots, v_{k-1}, v_k = v_0]$ tel que $v_{i-1}v_i \in E$ pour tout $i \in \{1, \dots, k\}$. Un cycle est simple si $v_i \neq v_j$ lorsque $0 \leq i \neq j \leq k-1$. Pour un cycle simple, les arêtes $v_iv_j \in E$ avec $(|i-j| \bmod k) > 1$ sont appelées des cordes. Les arêtes $v_iv_j \in E$ avec $(|i-j| \bmod k) = 2$ sont appelées des 2-cordes. Par exemple, pour le graphe G de la figure 1.4, la séquence $[1, 2, 4, 3, 1]$ représente un cycle de longueur 4 et l'arête $\{2, 3\}$ est une 2-corde. Un cycle simple est sans (2-)corde s'il ne possède pas de (2-)cordes. Un cycle de n sommets sans corde est noté C_n . Un graphe G est un graphe triangulé si tous les cycles de longueur supérieur ou égale à 4 possèdent au moins une corde. Par exemple, le graphe G de l'exemple 1.4 est triangulé.

Un arbre est un graphe connexe et sans cycle. Les sommets de degré 1 d'un arbre sont nommés feuilles et les autres sommets sont appelés sommets internes. Les sommets sont alors nommés nœuds dans un soucis de clarification afin de distinguer les notions d'arbre et de graphe quelconque. Un arbre possède les propriétés suivantes :

1. $|E| = |V| - 1$
2. il existe un chemin simple unique entre chaque paire de nœuds.
3. un arbre possède au moins deux feuilles.

Un arbre est dit enraciné s'il possède un nœud particulier nommé racine, noté r . Dans un arbre enraciné, la profondeur d'un nœud est la distance comptée en nombre d'arête séparant ce nœud de la racine. Pour un nœud $u \in V$, le nœud adjacent à u de profondeur inférieure est appelé père et les nœuds adjacents à u de profondeur supérieure sont appelés fils. Une branche est un chemin entre la racine et une feuille de l'arbre.

Le parcours en profondeur est un algorithme classique de parcours de graphe en partant d'un sommet donné. Il est rappelé par l'algorithme 1. Pour le cas des arbres enracinés, nous considérons le parcours en profondeur en partant du nœud racine.

Algorithme 1 Algorithme de parcours en profondeur

Require: a graph $G = (V, E)$ and a vertex source $s \in V$

Ensure: an ordering σ of the vertices

DFS (graph G , vertex s , integer **sequenceNumber**)

begin

$\sigma[\text{sequenceNumber}] = s$

vertex s is now visited

choose a vertex not visited $u \in V(s)$

DFS(G , u , **sequenceNumber**+1)

end

Une orientation de l'arête uv est une direction donnée à l'arête. Elle est notée $\vec{uv}$. Une orientation du graphe, noté F , est un ensemble d'orientation des arêtes du graphe tel que, pour toute arête $uv \in E$, $\vec{uv} \in F \wedge \vec{vu} \notin F$ ou $\vec{vu} \in F \wedge \vec{uv} \notin F$. Une orientation est dite transitive si pour tout $\vec{uv} \in F$ et $\vec{vw} \in F$ implique $\vec{uw} \in F$. Un graphe G est un graphe de comparabilité s'il existe au moins une orientation transitive des arêtes de G (cf. figure 1.5).

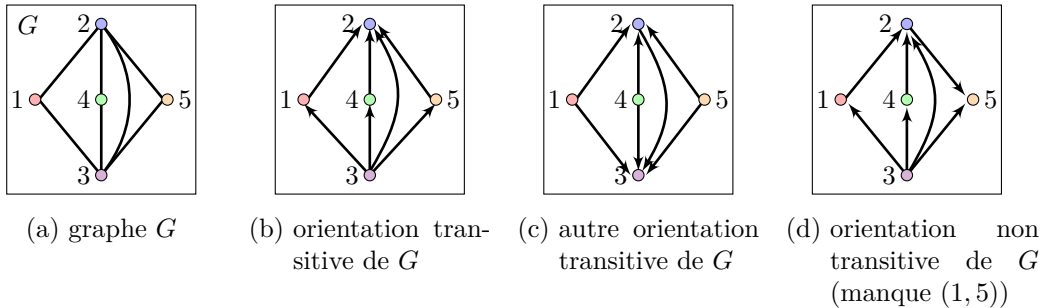


FIGURE 1.5: Exemple d'orientation transitive et non transitive sur le graphe de comparabilité G

1.2.2 Les graphes d'intervalles

Un graphe G est un graphe d'intervalles s'il existe une représentation par des intervalles de $\mathbb{R}$ du graphe tel que chaque sommet $v \in V$ de G est représenté par un intervalle I_v avec $uv \in E \Leftrightarrow I_u \cap I_v \neq \emptyset$ (cf. figure 1.6).

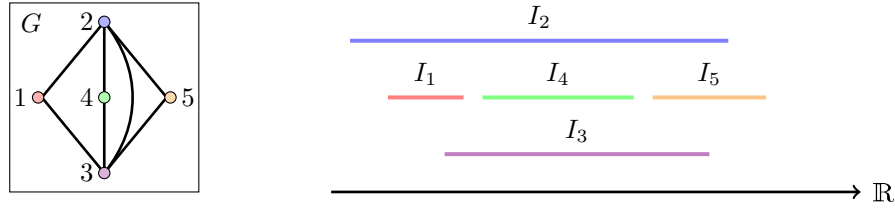


FIGURE 1.6: Exemple de représentation par des intervalles d'un graphe d'intervalles
\$G\$

1.2.2.1 Principales caractérisations des graphes d'intervalles

Les principales caractérisations des graphes d'intervalles ont été découvertes au cours des années 60.

Théorème 1.8 (Gilmore et Hoffman 1964 [40])

Un graphe est d'intervalles si et seulement si le graphe est triangulé et son graphe complémentaire est de comparabilité.

Théorème 1.9 (Gilmore et Hoffman 1964 [40])

Un graphe est d'intervalles si et seulement si le graphe ne contient pas de cycle de longueur 4 sans corde comme sous-graphe induit et son graphe complémentaire est de comparabilité.

Remarque Une orientation transitive dans le graphe complémentaire peut être vue comme un positionnement des intervalles dans la représentation par des intervalles du graphe. En effet, $\vec{uv} \in F$ peut être interprété par l'intervalle I_u est à gauche de l'intervalle I_v . ♦

Théorème 1.10 (Olariu 1991 [52])

Un graphe est d'intervalles si et seulement s'il existe une fonction d'ordonnancement des sommets $\sigma : V \mapsto \mathbb{N}$ tel que, pour tout triplet de sommets $(u, v, w) \in V^3$ avec $\sigma(u) < \sigma(v) < \sigma(w)$:

$$uw \in E \Rightarrow uv \in E.$$

Remarque Cette fonction d'ordonnancement peut être vue comme une numérotation suivant l'ordre de lecture gauche-droite en fonction du début des intervalles dans la représentation par des intervalles du graphe. Par exemple, pour la figure 1.6, l'ordre est $[2, 1, 3, 4, 5]$. De plus, cet ordre sur les sommets implique une orientation transitive dans le graphe complémentaire en suivant la règle : $[\sigma(u) < \sigma(v) \text{ et } uv \in E] \Leftrightarrow [\vec{uv} \in F]$. ♦

Théorème 1.11 (Fulkerson et Gross 1965 [35])

Un graphe est d'intervalles si et seulement si les cliques maximales du graphe peuvent être ordonnées tel que, pour tout sommet $u \in V$, les cliques contenant ce sommet u apparaissent consécutivement. C'est-à-dire que les cliques maximales $Q_1, \dots, Q_m$ sont classées de telle sorte que :

$$u \in Q_i \text{ et } u \in Q_j \text{ avec } 1 \leq i < j \leq m \Rightarrow u \in Q_k, \forall k \text{ tel que } i < k < j. \quad (1.2)$$

L'ordre définie par la règle dans le théorème 1.11 est appelé ordre de classement des cliques dans mon mémoire.

1.2.2.2 Les matrices des uns-consécutifs

Soit $\mathcal{M}_{n,m}(\{0,1\})$ l'ensemble des matrices booléennes de taille $n \times m$. Une matrice $M \in \mathcal{M}_{n,m}(\{0,1\})$ a la propriété des uns-consécutifs sur les lignes si pour chaque ligne $i \in \{1, \dots, n\}$ et chaque colonnes $k, k' \in \{1, \dots, m\}$ avec $k < k'$, $M_{ik} = 1$ et $M_{ik'} = 1$ implique $M_{il} = 1$ pour tout $k \leq l \leq k'$: pour chaque ligne, nous avons un seul ensemble de 1 apparaissant consécutivement.

Soit un graphe d'intervalles G avec n sommets et m cliques maximales $Q_1, \dots, Q_m$. Une matrice $M \in \mathcal{M}_{n,m}(\mathbb{B})$ est une matrice sommets/cliques de G si pour chaque indice $i \in \{1, \dots, n\}$ et $k \in \{1, \dots, m\}$, le sommet i appartient à la clique Q_k si et seulement si $M_{ik} = 1$ (cf. figure 1.7). Chaque colonne représente alors un vecteur caractéristique de la clique associée.

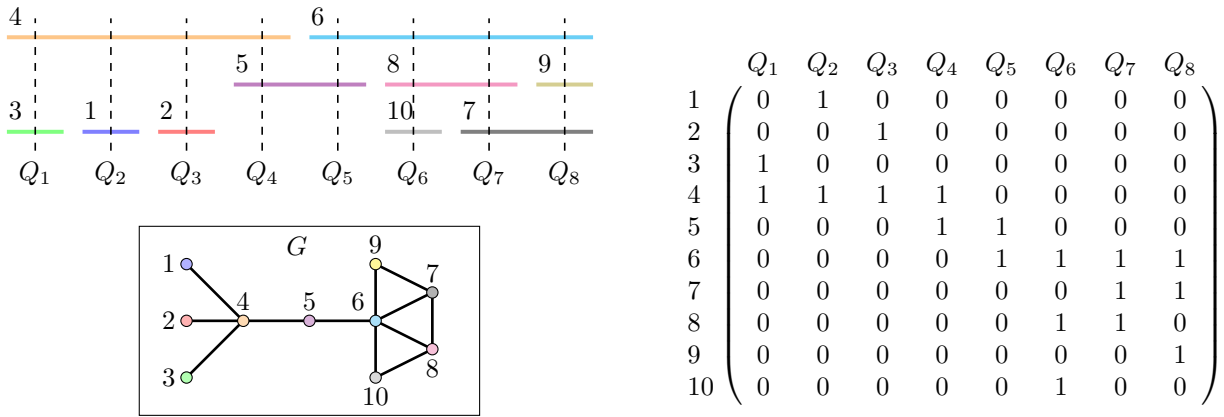


FIGURE 1.7: Exemple de matrice sommets/cliques maximales pour un graphe d'intervalles G

Le théorème qui suit est une réinterprétation du théorème 1.11 utilisant les notions de matrices de uns-consécutifs :

Théorème 1.12 (Fulkerson et Gross (1965))

Un graphe G est d'intervalles si et seulement s'il existe une matrice sommets/cliques maximales de G avec la propriété des uns-consécutifs sur les lignes.

1.2.2.3 Les MPQ-arbres

Les PQ-arbres sont définis par Booth et Lueker en 1976 [11]. Il s'agit d'une structure de données permettant de représenter l'ensemble des permutations permises d'un ensemble d'éléments U . Les permutations permises doivent respecter les restrictions de consécutivité sur une collection $\mathcal{U}$ de sous-ensembles de U . En effet, un ordre associé à U respectant la règle de consécutivité sur un sous-ensemble $W \in \mathcal{U}$ est un ordre sur les éléments de U tel que la consécutivité des éléments de W est respectée. Par exemple, si $U = \{A, B, C\}$ et

$W = \{A, B\}$ alors l'ordre BAC est valide et l'ordre ACB est invalide. Ainsi, l'ensemble des permutations permises est l'ensemble des ordres possibles respectant la consécutivité sur $\mathcal{U}$. Notons Π l'ensemble des permutations permises.

Un PQ-arbre est un arbre T enraciné avec deux types de nœuds internes P (représenté par un cercle) et Q (représenté par un rectangle) et tel que chaque feuille de l'arbre est labelisée par un élément de U (cf. figure 1.8). La frontière d'un PQ-arbre T , noté $F(T)$, est l'ordre des cliques obtenu par la lecture gauche-droite des feuilles de l'arbre. Ainsi, une frontière représente une permutation permise. Deux PQ-arbres T et T' sont équivalents si l'un peut être obtenu par l'autre en appliquant les règles suivantes (cf. figure 1.8) :

1. Permutation arbitraire des fils d'un nœud de type P ,
2. Inversion des fils d'un nœud de type Q .

Notons $T \equiv T'$ si T et T' sont équivalents. L'ensemble des permutations permises est noté $cons(T) = \{F(T') : T' \equiv T\}$.

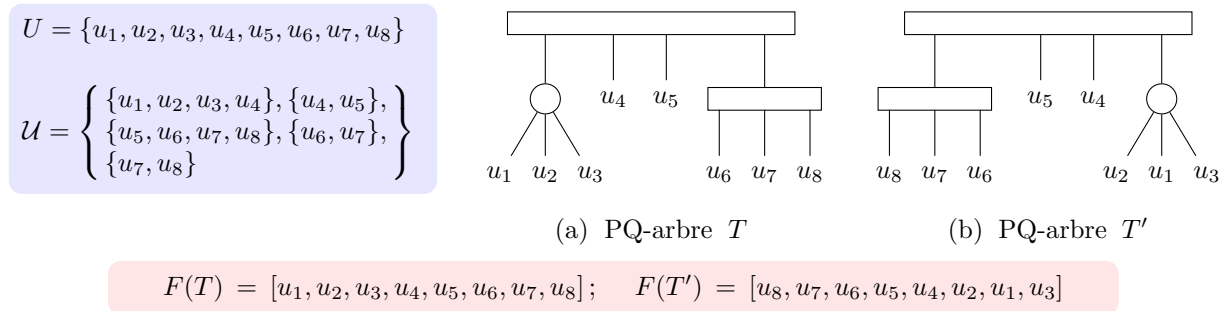


FIGURE 1.8: Exemple de deux PQ-arbres arbres T et T' équivalents

Un PQ-arbre est dit propre si les règles suivantes sont respectées :

- les feuilles de l'arbre doivent contenir exactement un seul élément de U ,
- les nœuds P doivent contenir aux moins deux fils,
- les nœuds Q doivent contenir aux moins trois fils.

Nous supposons maintenant que les PQ-arbres que nous considérons sont propres.

Dans le cas des graphes d'intervalles, prenons $U = \mathcal{Q}$ l'ensemble des cliques maximales et $\mathcal{U}$ l'ensemble des sous-ensembles de cliques maximales partageant un sommet en commun :

$$\mathcal{U} = \left\{ W \subseteq \mathcal{Q} : \exists u \in V \text{ avec } W = \mathcal{Q}(u) \right\}.$$

Alors, en utilisant le théorème 1.11, il suit le théorème :

Théorème 1.13 (Booth et Lueker [11])

Un graphe G est un graphe d'intervalles si et seulement s'il existe un PQ-arbre T tel que $F(T)$ est un arrangement consécutif valide des cliques maximales de G (cf. figure 1.9).

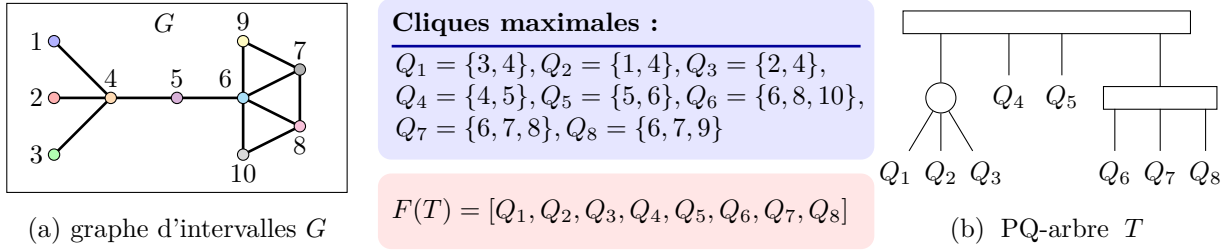


FIGURE 1.9: Exemple d'un graphe d'intervalles et d'un PQ-arbre associé

Les PQ-arbres modifiés (ou MPQ-arbres) sont introduits par Korte et Möhring en 1989 [50] afin de corriger le fait que l'approche développée par Booth et Lueker n'était pas incrémentale. Il s'agit d'un PQ-arbre étiqueté, c'est-à-dire qu'une étiquette peut être associée à chaque nœud de l'arbre.

Pour les graphes d'intervalles, les sommets du graphe représentent les étiquettes et sont affectés aux nœuds de l'arbre de telle manière que chaque branche est une clique maximale du graphe. Le sommet est placé au nœud avec la plus petite profondeur possible telle que chaque clique maximale contenant ce sommet est une branche fille de ce nœud référence. Nous parlons alors de MPQ-arbre associé à G . (cf. figure 1.10).

Pour les graphes d'intervalles, les nœuds de l'arbre T doivent respecter les règles d'étiquetage suivantes :

- Une feuille est étiquetée par un ensemble de sommets de G qui sont contenus dans une seule et même clique maximale ;
- Un nœud P est étiqueté par un ensemble de sommets de G qui sont contenus uniquement dans toutes les cliques maximales représentées par le sous-arbre de T enraciné en ce nœud P ;
- Un nœud Q , avec m fils $T_1, \dots, T_m$, est étiqueté par une liste d'ensemble de sommets notée $A_1, \dots, A_m$. Pour tout $k \in \{1, \dots, m\}$, chaque ensemble A_k est nommé section et correspond au fils T_k . Chaque fils T_k peut-être vu comme un sous-arbre enraciné en A_k . Un ensemble de sommets est associé à chaque section. Chaque sommet d'une section est contenu uniquement dans toutes les cliques maximales représentées par les sous-arbres de T enraciné en cette section et les autres sections du nœud Q contenant le sommet étudié. Toutes les sections contenant un même sommet apparaissent alors consécutivement.

Soit N un nœud quelconque du MPQ-arbre. Nous notons V_N l'ensemble des sommets du graphe contenus dans le nœud N . Pour un nœud de type Q , nous avons donc $V_N = A_1 \cup \dots \cup A_m$. Notons aussi $T(G)$ le MPQ-arbre associé au graphe d'intervalles G .

Théorème 1.14 (Korte et Möhring 1989 [50])

Un graphe G est un graphe d'intervalles si et seulement s'il existe un MPQ-arbre T associé à G (cf. figure 1.10).

Remarque Chaque clique maximale de G correspond à un chemin de la racine vers une

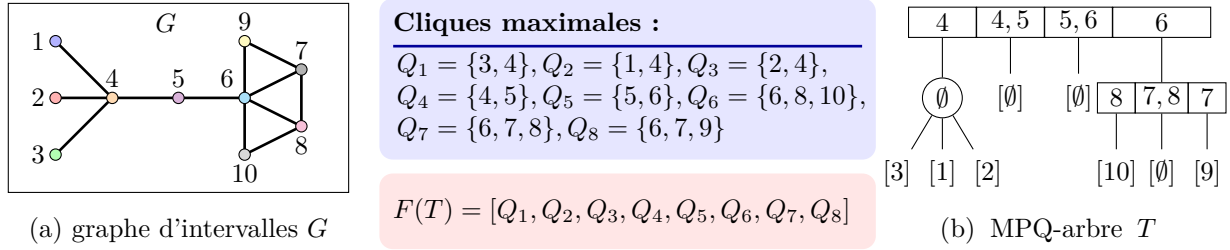


FIGURE 1.10: Exemple d'un graphe d'intervalles et d'un MPQ-arbre associé

feuille de l'arbre où chaque sommet associé à un nœud du chemin appartient à la clique. Et réciproquement. Notons $\mathcal{B}$ l'ensemble des branches du MPQ-arbre. De plus, nous étendons la notion de frontière aux MPQ-arbres en considérant les cliques composant les branches. Deux MPQ-arbres sont équivalents si l'un peut être obtenu à partir de l'autre arbre en appliquant les mêmes règles que pour les PQ-arbres. ♦

L'utilisation des MPQ-arbres permet donc d'avoir une nouvelle caractérisation des graphes d'intervalles. L'intérêt des MPQ-arbres réside dans le fait que les items peuvent être ajoutés un à un en ne modifiant qu'une branche de l'arbre. Il s'agit donc d'un algorithme incrémental (cf chapitre IV pour plus de précision).

Enfin, nous définissons l'ordonnancement associé au MPQ-arbre T comme étant l'ordre des sommets obtenus par le parcours en profondeur de l'arbre en partant prioritairement vers les branches les plus à gauche. Par exemple, pour le MPQ-arbre illustré par la figure 1.10, l'ordre est $[4, 3, 1, 2, 5, 6, 8, 10, 7, 9]$. L'ordre d'un MPQ-arbre est un ordre valide pour le théorème 1.10 : pour chaque arête $uw \in E$, il suffit de remarquer que tout sommet $v \in V$ tel que $L(u) < L(v) < L(w)$ appartient forcément au sous-arbre enraciné en u .

1.3 L'algorithme de Banch-and-Price

Les approches de décomposition sont des techniques classiques d'optimisation consistant à reformuler le problème traité en séparant la difficulté du problème en deux problèmes distincts (un problème maître et un ou plusieurs sous-problèmes). Ces techniques offrent aussi comme avantage d'éliminer certaines symétries liées aux problèmes traités. Dans cette section, nous allons nous intéresser plus particulièrement à la reformulation de Dantzig-Wolfe. Pour plus de détails, nous vous conseillons de lire [24, 60].

La technique de reformulation de Dantzig-Wolfe se résume comme suit. Considérons le problème F suivant :

$$[F] \quad \min \quad cx, \quad (1.3a)$$

$$\text{s. t.} \quad Ax \geq a, \quad (1.3b)$$

$$Bx \geq b, \quad (1.3c)$$

$$x \in \mathbb{N}^n. \quad (1.3d)$$

où $c \in \mathbb{R}^n$, $a \in \mathbb{R}^m$, $A \in \mathcal{M}_{m,n}(\mathbb{R})$, $b \in \mathbb{R}^p$ et $B \in \mathcal{M}_{p,n}(\mathbb{R})$. La contrainte (1.3b) est vue comme des contraintes difficiles ou liantes tandis que la contrainte (1.3c) définit les

contraintes associées à un sous-problème. Par exemple, si le problème possède une sous-structure dite diagonale par blocs, alors le système de contrainte associé à cette structure forme les contraintes du sous-problème. Les autres contraintes du problème sont des contraintes qui lient les blocs entre eux et sont appelées contraintes liantes.

Soit $X = \{x \in \mathbb{N}^n : Bx \geq b\}$ le sous-problème, supposé non vide et borné, et soit $\mathcal{Q}$ l'ensemble énuméré des solutions du sous-problèmes. Alors, $X = \{x^q\}_{q \in \mathcal{Q}}$ et le problème reformulé s'écrit :

$$[M] \quad Z^M = \min \quad \sum_{q \in \mathcal{Q}} (cx^q) \lambda_q, \quad (1.4a)$$

$$\text{s. t.} \quad \sum_{q \in \mathcal{Q}} (Ax^q) \lambda_q \geq a, \quad (\pi) \quad (1.4b)$$

$$\sum_{q \in \mathcal{Q}} \lambda_q = 1, \quad (\sigma) \quad (1.4c)$$

$$\lambda_q \in \{0, 1\}, \quad \forall q \in \mathcal{Q}. \quad (1.4d)$$

Le problème M est appelé problème maître. Notons Z_{PL}^M la valeur de l'objectif de la relaxation linéaire ($\lambda_q \in [0, 1], \forall q \in \mathcal{Q}$) de M .

Typiquement, la formulation M possèdent un nombre trop important de variables dû à la taille en général exponentielle de X . L'énumération exhaustive de X paraissant difficilement envisageable, une procédure de génération de colonnes est utilisée pour résoudre la relaxation linéaire du problème M . Par cette procédure, que nous rappelons brièvement en suivant, les colonnes (ou variables) sont ajoutées dynamiquement à la formulation en fonction du coût réduit associé aux variables :

Procédure de génération de colonnes

① Résoudre à l'optimum la relaxation linéaire du problème maître M restreint au sous-ensemble de colonnes générées $\overline{\mathcal{Q}}$ jusqu'à présent. Enregistrer la solution duale associée (π, σ) .

② Résoudre le sous-problème suivant :

$$\zeta(\pi) = \min\{(c - \pi A)x : x \in X\}.$$

③ Vérifier s'il existe des colonnes de coût réduit négatif :

Si $\zeta(\pi) + \sigma < 0$, ajouter la colonne $x^q = \operatorname{argmin} \zeta(\pi)$ à $\overline{\mathcal{Q}}$ et retourner à l'étape ①. Sinon la solution courante est la solution optimale du problème maître relâché.

Remarque L'initialisation de la procédure de génération de colonnes se fait par l'ajout de variables artificielles. ◆

A chaque itération de la procédure de génération de colonnes, la valeur de l'objectif du problème maître relâché restreint ne constitue pas une borne duale valide pour le problème maître entier. Cependant, une borne duale valide $L(\pi)$ peut être obtenue en se basant sur la relaxation lagrangienne du problème maître appliquée à la contrainte (1.4b) :

$$L(\pi) = \min \sum_{q \in \mathcal{Q}} (c x^q) \lambda_q + \pi \left(a - \sum_{q \in \mathcal{Q}} (A x^q) \lambda_q \right), \quad (1.5a)$$

$$\text{s. t. } \sum_{q \in \mathcal{Q}} \lambda_q = 1, \quad (1.5b)$$

$$\lambda_q \geq 0, \quad \forall q \in \mathcal{Q}. \quad (1.5c)$$

En remarquant que $\arg\min \zeta(\pi)$ est la solution du problème, la borne lagrangienne devient :

$$L(\pi) = \zeta(\pi) + \pi a. \quad (1.6)$$

La meilleure borne lagrangienne est notée $LD = \max_{\pi \in \mathbb{R}^m} L(\pi)$. Ainsi, au cours du processus de génération de colonnes, nous enregistrons à chaque étape la meilleure borne lagrangienne trouvée. De plus, un nouveau critère d'arrêt peut alors être utilisé : si la meilleure borne lagrangienne trouvée est supérieure ou égale à la valeur du problème maître relâché restreint, alors la procédure peut s'arrêter.

Remarque En général, la borne obtenue par la relaxation linéaire du problème est meilleure que la borne obtenue par la relaxation linéaire de la formulation originelle (1.3a)-(1.3d). En effet, la reformulation de Dantzig-Wolfe équivaut à une convexification implicite des solutions du sous-problème :

$$Z_{LP}^M = \min \{ cx : Ax \geq a, x \in \text{conv}(X) \},$$

avec $\text{conv}(X) = \{x \in \mathbb{R}^n : x = \sum_{q \in \mathcal{Q}} x^q \lambda_q, \sum_{q \in \mathcal{Q}} \lambda_q = 1, \lambda_q \geq 0 \forall q \in \mathcal{Q}\}$. Seul le cas $\text{conv}(X) = \{x \in \mathbb{R}^n : Bx \geq b\}$ garantit que la borne Z_{LP}^M est égale à la borne obtenue par la relaxation linéaire du problème originel. $\blacklozenge$

1.4 Les différentes modélisations

Nous allons présenter maintenant les différentes approches développées dans la littérature pour formuler et résoudre différentes variantes du problème de placement sur deux dimensions.

1.4.1 Modélisation par la programmation mathématique

Les premières études ont permis de modéliser le problème de sac-à-dos sur deux dimensions et de « bin packing » sur deux dimensions dès 1985.

En effet, la première formulation du problème fut proposé par Beasley [4]. Elle consistait à utiliser une discrétisation de l'espace disponible afin de déterminer la position des objets selon les coordonnées définies par la discrétisation (variables $\Psi_{iab} = 1$ si le coin inférieur-gauche de l'objet i est en position (a, b)). Pour le problème de sac-à-dos, nous avons la formulation suivante :

$$\max \left\{ \sum_i \rho_i \delta_i : \sum_i \sum_{\nu=a-\mathbf{w}_i+1}^a \sum_{\tau=b-\mathbf{h}_i+1}^b \Psi_{i\nu\tau} \leq 1 \forall a, b; \sum_{(a,b)} \Psi_{iab} = \delta_i \forall i; \delta_i, \Psi_{iab} \in \mathbb{B} \forall i, a, b \right\}.$$

D'autres formulations utilisant une approche similaire sont développées par Hadjiconstantinou et Christofides en 1995 [41]; Beasley et Mingozzi en 1996 [5] ou Boschetti, Mingozzi et Hadjiconstantinou en 2002 [12].

Une autre approche fut développée en 1995 par Chen, Lee et Shen [19, 26, 53, 54, 55]. Elle utilisait des variables de positionnement relatifs des objets les uns par rapport aux autres (variables $g_{ij} = 1$ (resp. d_{ij}) si l'objet i est à gauche (resp. en dessous) de l'objet j) :

$$\max \left\{ \sum_i \rho_i \delta_i : \begin{array}{l} x_i + \mathbf{w}_i \leq x_j + \mathbf{W}(1 - g_{ij}) \quad \forall i, j; \quad y_i + \mathbf{w}_i \leq y_j + \mathbf{H}(1 - d_{ij}) \quad \forall i, j; \\ g_{ij} + g_{ji} + d_{ij} + d_{ji} - \delta_i - \delta_j \geq -1 \quad \forall i, j; \quad \delta_i, g_{ij}, d_{ij} \in \mathbb{B} \quad \forall i, j \end{array} \right\}.$$

En plus de l'étude de formulations compactes ou pseudo-polynomiales, des études utilisant les techniques de décomposition sont aussi développées [12, 56, 58]. Ces approches sont généralement résolues à l'aide d'une procédure de génération de colonnes. Dans le cas des problèmes de placement, cette approche est vue comme le remplissage du conteneur par des bandes verticales et horizontales ne dépassant pas la taille du conteneur, où les bandes représentent des colonnes de la procédure de génération de colonnes (cf figure 1.11). Soit $\mathcal{V}$

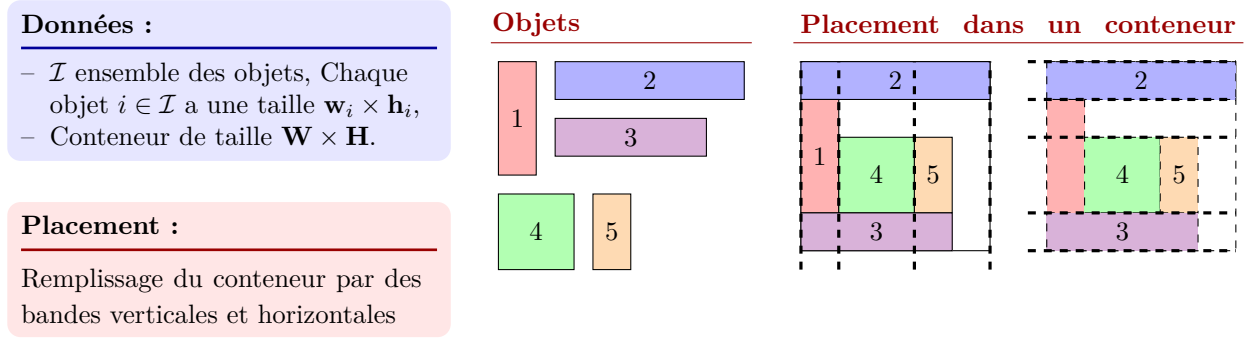


FIGURE 1.11: exemple de remplissage par bandes d'un conteneur

(resp. $\mathcal{H}$) l'ensemble des bandes verticales (resp. horizontales). Des variables sont utilisées pour déterminer la largeur de chaque bande sélectionnée (variables λ_v (resp. μ_h) représentant la largeur (resp. hauteur) de la bande $v \in \mathcal{V}$ (resp. $h \in \mathcal{H}$)). Scheithauer définit la formulation suivante :

$$\max \left\{ \sum_i \rho_i \delta_i : \begin{array}{l} \sum_{v \in \mathcal{V}} x_i^v \lambda_v = \mathbf{w}_i \delta_i \quad \forall i; \quad \sum_{h \in \mathcal{H}} y_i^h \mu_h = \mathbf{h}_i \delta_i \quad \forall i; \quad \sum_{v \in \mathcal{V}} \lambda_v \leq \mathbf{W}; \\ \sum_{h \in \mathcal{H}} \mu_h \leq \mathbf{H}; \quad \delta_i \in \mathbb{B} \quad \forall i; \quad \lambda_v \geq 0 \quad \forall v \in \mathcal{V}; \quad \mu_h \geq 0 \quad \forall h \in \mathcal{H} \end{array} \right\}.$$

Cependant, ces dernières formulations ne sont que des relaxations du problème original et ne permettent d'avoir que des bornes supérieures. A partir de 2007, Belov et al. [8, 9, 10] ont étudié les formulations étendues précédentes en améliorant leur qualité de bornes.

Enfin, une approche par ajout de plans coupants fut proposée par Baldacci et Boschetti [3] en 2007. Ils utilisent la formulation suivante, où $\mathcal{S}$ représente l'ensemble des ensembles d'objets qui ne peuvent pas être rangés dans un même conteneur :

$$\max \left\{ \sum_i \rho_i \delta_i : \sum_{i \in S} \delta_i \leq |S| - 1 \quad \forall S \in \mathcal{S}; \quad \delta_i \in \mathbb{B} \quad \forall i \right\}.$$

L'ensemble $\mathcal{S}$ n'étant pas connu a priori car trop difficile à énumérer, ils remplacent les contraintes par des contraintes classiques de sac-à-dos sur une dimension. Ils utilisent aussi des heuristiques pour avoir des bornes inférieures et des bornes supérieures obtenues par la littérature. L'ensemble de ces contraintes permet alors d'obtenir de nouvelles bornes duales.

Ces différentes formulations sont explicitées plus en détail dans le chapitre II.

1.4.2 Modélisation par la théorie des graphes

Soit $G = (V, E)$ un graphe non orienté. Pour tout sommet $v \in V$, nous associons une valeur $\mathbf{w}_v \in \mathbb{R}$ appelée poids. La valeur maximale du stable pondéré de G , notée $\alpha(G, \vec{\mathbf{w}})$, est la plus grande somme des poids d'un ensemble stable :

$$\max \left\{ \sum_{v \in S} \mathbf{w}_v : S \text{ stable} \right\}.$$

En 1997, Fekete et Schepers [27, 28, 32, 33] proposent une caractérisation basée sur les graphes d'intervalles. En effet, ils ont observé que, pour un placement réalisable donné, la projection des objets sur les différents axes permet d'obtenir des graphes d'intervalles. Le graphe obtenu à partir de la projection des objets sur l'axe des abscisses est noté G_x , celui obtenu par la projection des objets sur l'axe des ordonnées est noté G_y . L'originalité de leur approche repose sur la détermination de conditions nécessaires et suffisantes sur ces graphes pour établir la réciproque (cf théorème 1.15 et figure 1.12).

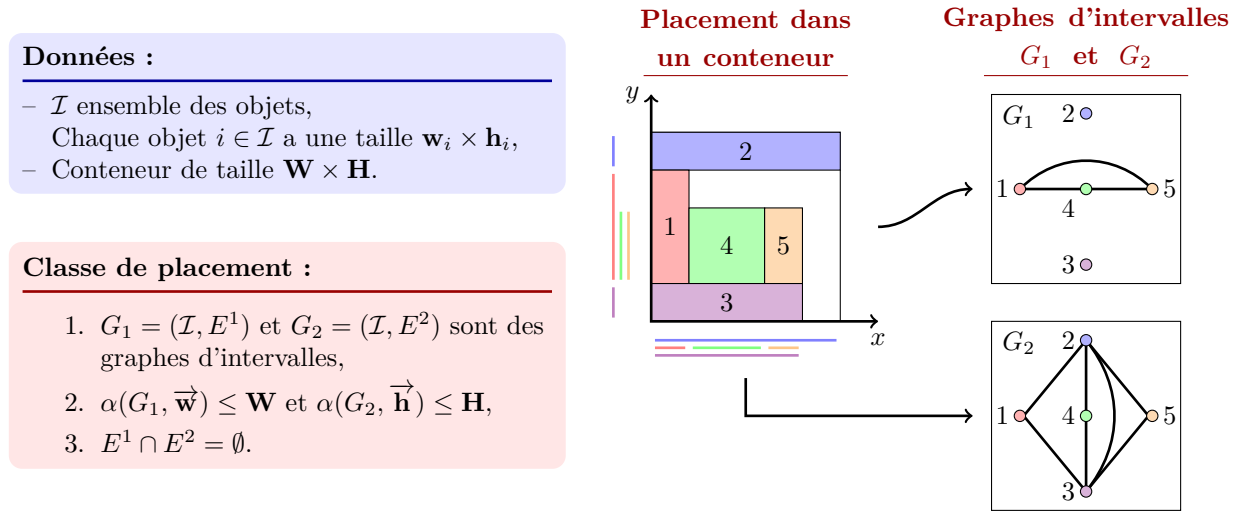


FIGURE 1.12: Classe de placement pour un placement sur deux dimensions

Théorème 1.15 (Fekete et Schepers 1997 [27])

Un ensemble $\mathcal{I}$ d'objets peut être placé dans le conteneur si et seulement s'il existe un ensemble de deux graphes $G_x = (\mathcal{I}, E_x)$ et $G_y = (\mathcal{I}, E_y)$ avec les propriétés suivantes :

$$G_x \text{ et } G_y \text{ sont des graphes d'intervalles,} \quad (1.7a)$$

$$\alpha(G_x, \vec{\mathbf{w}}) \leq \mathbf{W} \text{ et } \alpha(G_y, \vec{\mathbf{h}}) \leq \mathbf{H}, \quad (1.7b)$$

$$E_x \cap E_y = \emptyset. \quad (1.7c)$$

La condition (1.7a) permet donc de définir un positionnement relatif des objets les uns par rapport aux autres. La condition (1.7b) permet de s'assurer que les objets ne dépassent pas du conteneur alors que la condition (1.7c) garantit que les objets ne se chevauchent pas.

Le principal avantage de cette modélisation est que différents placements peuvent avoir la même caractérisation par les graphes (cf. figure 1.13). Ainsi, de nombreuses symétries sont caractérisées dans une même modélisation réduisant ainsi l'espace des solutions à énumérer. Un couple de graphe (G_x, G_y) ayant les propriétés (1.7a)-(1.7c) est alors appelé classe de placement.

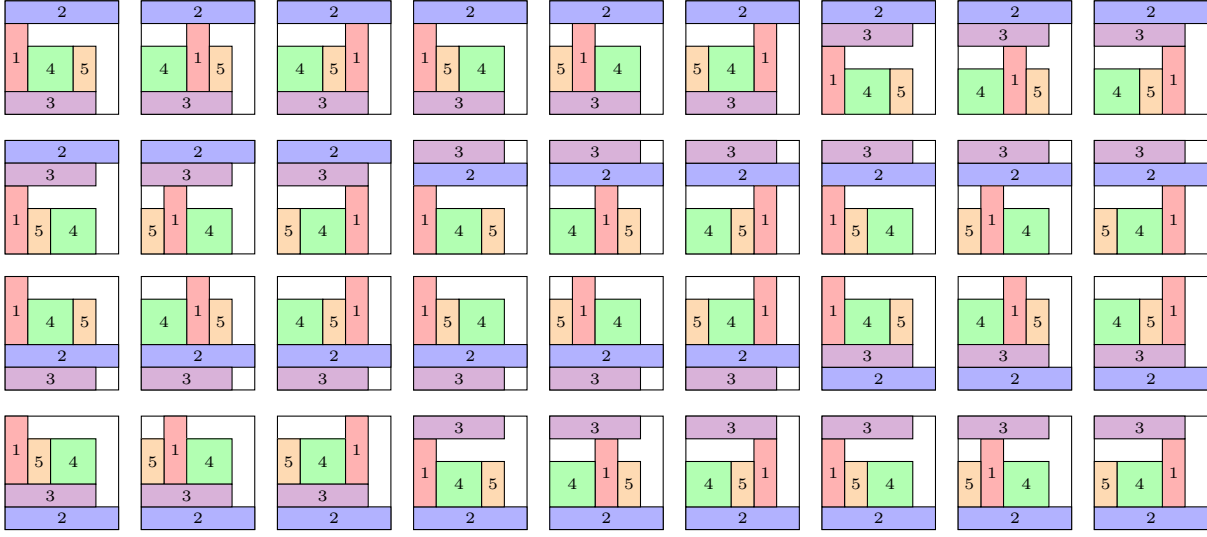


FIGURE 1.13: 36 placements différents avec la même modélisation par les graphes (cf. figure 1.12)

Remarque Pour chaque classe de placement, un placement réalisable peut être obtenu. En effet, les graphes complémentaires de graphes d'intervalles sont des graphes de comparabilité (cf. théorème 1.8). Soit F_x (resp. F_y) une orientation transitive dans le graphe complémentaire de G_x (resp. G_y). Alors, le placement suivant est un placement valide :

$$\begin{aligned} x_i &= \max \left\{ x_j + \mathbf{w}_j : \overrightarrow{(i, j)} \in F_x \right\} \\ y_i &= \max \left\{ y_j + \mathbf{h}_j : \overrightarrow{(i, j)} \in F_y \right\} \end{aligned}, \quad \forall i \in \mathcal{I}.$$

◆

Pour connaître la réalisabilité d'un ensemble d'objets donné, l'algorithme de Fekete et Schepers énumère toutes les classes de placement potentielles jusqu'à trouver une classe valide. Leur algorithme utilise la caractérisation donnée par le théorème 1.9 pour savoir si un graphe est un graphe d'intervalles.

Cependant, de nombreuses symétries ne sont pas éliminées par cette caractérisation (cf. figure 1.14).

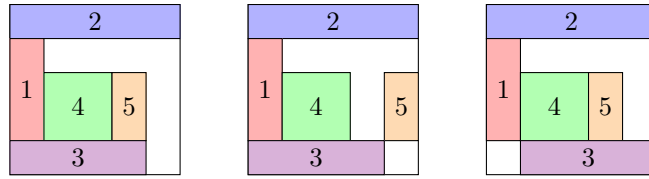


FIGURE 1.14: Différents placements proches ayant des classes de placement différentes

De plus, Ferreira et Oliveira [34] montrent que des phénomènes de dégénérescence existent dans certaines classes de placement. En effet, différentes classes peuvent correspondre au même placement car certaines arêtes inutiles peuvent être rajoutées (cf. figure 1.15).

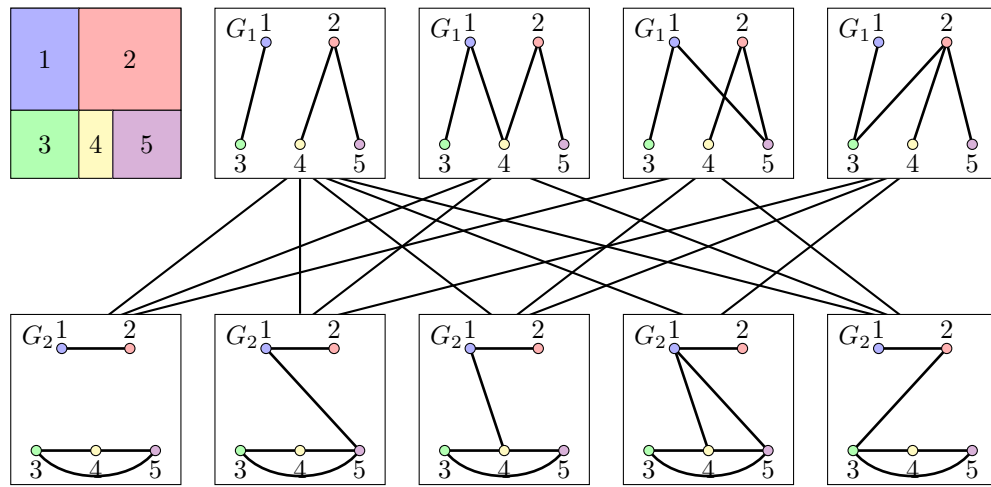


FIGURE 1.15: 14 classes de placement correspondant au même placement par reconstruction

1.4.3 Approches par des algorithmes combinatoires et/ou par la programmation par contraintes

Dès 1995, Hadjiconstantinou et Christofides [41] proposent un algorithme énumératif basé sur une formulation. Il consiste à placer chaque objet sur une coordonnée parmi les positions respectant la condition « calée en bas à gauche » (positions définies par les objets précédemment placés). En 1998, Martello et Vigo [51] et en 2003, Caprara et al. [13, 15] utilisent une approche similaire pour résoudre le problème de sac-à-dos sur deux dimensions. Cependant, de nombreuses symétries subsistent dont celles mises en évidence par Fekete et Schepers. De plus, comme nous le verrons par la suite, il est possible de rajouter des inégalités valides à la formulation pour modéliser la condition « calée en bas à gauche ».

Une autre approche fut développée par Clautiaux et al. [21] en 2007. Contrairement à l'approche de Hadjiconstantinou et Christofides, celle-ci consiste à séparer les décisions entre les abscisses et les ordonnées en générant, dans un premier temps, l'ensemble des

abscisses pour le sous-ensemble testé. Si l'ensemble des objets possède une abscisse, l'algorithme essaye de générer des ordonnées pour chaque objet. Cette approche offre un double avantage. Le premier est que si un sous-ensemble n'est pas réalisable selon les abscisses, il est inutile d'énumérer cette solution sur la deuxième dimension. Ainsi, de nombreux cas irréalisables sont éliminés rapidement. Le deuxième est que certaines symétries de position peuvent être éliminées par des ordres lexicographiques. Cependant, cette approche dépend toujours d'une discrétisation de l'espace. Dans le cas d'un conteneur de grande taille, l'algorithme peut être moins performant.

En plus de ces travaux basées sur des algorithmes combinatoires, différentes études ont été publiées utilisant des techniques de programmation par contraintes pour résoudre le problème de réalisabilité d'un placement. Dès 1994, van Hentenryck [59] proposa d'utiliser la programmation par contraintes pour résoudre le problème de réalisabilité d'un placement en testant les contraintes de non-intersection des objets et de non-dépassement de la taille du conteneur. A partir de 2003, Korf [42, 48, 49] proposa d'améliorer cette approche en utilisant les connaissances sur les problèmes de « bin packing ». Ces approches traitent les deux dimensions en même temps en plaçant les objets selon une coordonnée. Des dominances entre les placements permettaient d'éliminer certaines symétries. Durant ma thèse, Clautiaux et al. [22] a proposé des évolutions dans le domaine de la résolution par la programmation par contraintes des problèmes d'ordonnancement de ressource (« cumulative scheduling problem ») pour améliorer leur approche initiale. En 2008, Simonis et O'Sullivan [57] ont utilisé les nouvelles avancées dans les problèmes d'ordonnancement de ressource apportées par Beldiceanu et Carlsson [7, 6] pour améliorer la résolution du problème de placement faite par Clautiaux et al. Ils effectuent aussi une étude comparative entre les approches consistant à considérer les deux dimensions à la fois et les approches séparant les décisions entre les deux dimensions. Van Hentenryck, Korf, Simonis et O'Sullivan ont testé ces approches pour résoudre le problème consistant à trouver le plus petit conteneur contenant l'ensemble des carrés de taille 1 à n . Toutes ces approches utilisent une discrétisation de l'espace comme base pour leur algorithme. Enfin, en 2007, Pisinger et Sigurd [55] ont utilisé la programmation par contraintes associée à une formulation compacte pour résoudre le problème de sac-à-dos sur deux dimensions.

1.4.4 Bornes duales

Différentes bornes duales ont été développées dans la littérature. La première borne classique pour le problème de sac-à-dos sur deux dimensions consiste à résoudre le problème de sac-à-dos sur une dimension où les poids des objets correspond aux volumes des objets. Notons la formulation F_0 :

$$[F_0] \quad \max \left\{ \sum_{i \in \mathcal{I}} \rho_i \delta_i : \sum_{i \in \mathcal{I}} \mathbf{w}_i \mathbf{h}_i \delta_i \leq \mathbf{WH}, \quad \delta_i \in \{0, 1\} \quad \forall i \in \mathcal{I} \right\}.$$

Il est possible d'obtenir de meilleures bornes duales en utilisant les fonctions duales réalisables. Une fonction $f : [0, 1] \rightarrow [0, 1]$ est dite duale réalisable (DFF) si pour chaque ensemble S de nombres réels positifs, nous avons $\sum_{x \in S} x \leq 1 \Rightarrow \sum_{x \in S} f(x) \leq 1$. Au départ, elles étaient utilisées pour obtenir des bornes duales pour le problème de « cutting stock ».

Par leur caractérisation des placements par les graphes d'intervalle, Fekete et Schepers démontrent qu'elles peuvent aussi être utilisées pour les problèmes de placement

[29, 30, 31]. Soient f et g deux fonctions duales réalisables. Par exemple, pour le problème de « bin packing » sur deux dimensions, une borne duale est donnée par $\left\lceil \sum_{i \in \mathcal{I}} f\left(\frac{\mathbf{w}_i}{\mathbf{W}}\right) g\left(\frac{\mathbf{h}_i}{\mathbf{H}}\right) \right\rceil$.

Carlier et Néron [16, 17, 18] généralisent les DFF et définissent les fonctions discrètes duales réalisables de la façon suivante : une fonction $f : [0, C] \rightarrow [0, C']$ est appelée fonction discrète duale réalisable si pour chaque ensemble S de nombres réels positifs, nous avons $\sum_{x \in S} x \leq C \Rightarrow \sum_{x \in S} f(x) \leq f(C) = C'$. Pour d'autres détails sur les DFF, nous vous invitons à vous reporter sur les papiers de Carlier et al. [20, 23]. Ils introduisent aussi les notions de dominance et de maximalité d'une fonction, notée MDFF. Pour le problème de sac-à-dos sur deux dimensions, la borne duale obtenue en appliquant des MDFF est donnée par :

$$[F_{DFF}] \quad \max \left\{ \sum_{i \in \mathcal{I}} \rho_i \delta_i : \sum_{i \in \mathcal{I}} f(\mathbf{w}_i) g(\mathbf{h}_i) \delta_i \leq f(\mathbf{W}) g(\mathbf{H}); \quad \delta_i \in \mathbb{B} \quad \forall i \in \mathcal{I} \right\}. \quad (1.8)$$

Voici quelques exemples de MDFF :

$$f_1^k : [0, C] \rightarrow [0, kC] \quad \text{avec } k \in \{1, \dots, C\}; \quad f_2^k : [0, C] \rightarrow [0, 2 \lfloor \frac{C}{k} \rfloor] \quad \text{avec } k \in \{1, \dots, C\}$$

$$x \mapsto \begin{cases} xk, & \text{si } \frac{x(k+1)}{C} \in \mathbb{N}, \\ \lfloor \frac{x(k+1)}{C} \rfloor C, & \text{sinon.} \end{cases} \quad x \mapsto \begin{cases} 2 \lfloor \frac{C}{k} \rfloor - 2 \lfloor \frac{C-x}{k} \rfloor, & \text{si } x > \frac{C}{2}, \\ \lfloor \frac{C}{k} \rfloor, & \text{si } x = \frac{C}{2}, \\ 2 \lfloor \frac{x}{k} \rfloor, & \text{sinon.} \end{cases}$$

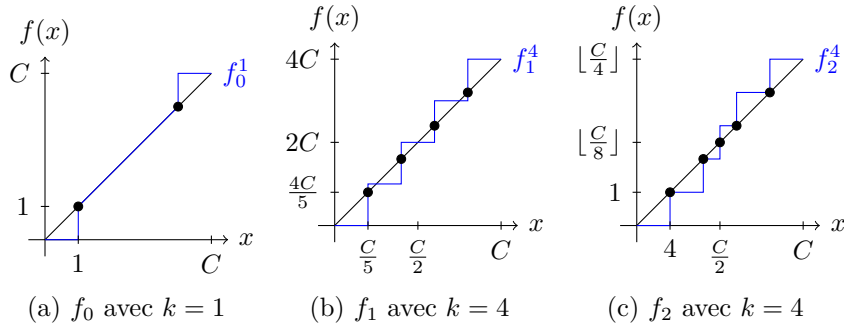


FIGURE 1.16: Exemples de MDFF

Caprara et Monaci [14] utilisèrent une approche bilinéaire pour estimer les valeurs des fonctions duales. Notons $U_{F_{BiL}^1}^\delta$ et $U_{F_{BiL}^2}^\delta$ les bornes obtenues. La deuxième borne donne des résultats de meilleure qualité mais sont plus coûteuses en temps de calcul.

1.4.5 Instances des problèmes de sac-à-dos sur deux dimensions résolues dans ce mémoire

Durant ce mémoire, nous allons nous intéresser au problème de sac-à-dos sur deux dimensions. Pour résoudre ce problème, l'approche la plus performante consiste à décomposer le problème en deux étapes. La première consiste à énumérer des sous-ensembles d'objets. Pour chaque sous-ensemble sélectionné, la deuxième étape consiste à tester la réalisabilité de ce sous-ensemble. Cette approche fut utilisée la première fois par Fekete

et Schepers. La première étape se fait par un algorithme de « Branch-and-Bound » sur le problème de sac-à-dos relâché à une seule dimension (Formulation F_0). L'utilisation des MDFF permet d'améliorer la sélection des sous-ensembles en détectant les cas irréalisables rapidement. Caprara et Monaci [13] proposent une comparaison des algorithmes de « Branch-and-Bound » pouvant être utilisés pour cette première étape. La deuxième consiste alors à utiliser une des procédures de réalisabilité précédemment définies pour savoir si le sous-ensemble rentre dans le conteneur.

Inst	Taille conteneur	Nbre objets	Valeur optimum	Nb objets dans sol	Valeur bornes duales		
					$U_{F_0}^\delta$	$U_{F_{DFF}}^\delta$	$U_{F_{BiL}}^{\delta^2}$ ^a
ngcut1	10×10	10	164	5	201	201	164
ngcut2	10×10	17	230	5	253	253	230
ngcut3	10×10	21	247	7	266	254	249
ngcut4	15×10	7	268	6	275	268	268
ngcut5	15×10	14	358	6	373	358	358
ngcut6	15×10	15	289	7	317	317	289
ngcut8	20×20	13	834	8	938	906	856
ngcut9	20×20	18	924	11	962	924	924
ngcut10	30×30	13	1452	6	1517	1452	1452
ngcut11	30×30	15	1688	9	1864	1766	1726
ngcut12	30×30	22	1865	9	2012	1913	1865
cgcut1	15×10	16	244	8	260	246	244
cgcut2	40×70	32	2892	12	2919	2915	2903
cgcut3	40×70	62	1860	10	2020	1940	1900
wang20	70×40	42	2726	8	2800	2762	2752
gcut1	250×250	10	48368	3	62488	48440	48368
gcut2	250×250	20	59798	6	62500	61420	59798
gcut3	250×250	30	61275	6	62500	61384	61275
gcut4	250×250	50	61380	4	62500	62496	≤ 62497
gcut5	500×500	10	195582	5	249854	230446	195582
gcut6	500×500	20	236305	4	249992	241018	236305
gcut7	500×500	30	240143	4	249998	246057	240143
gcut8	500×500	50	245758	4	250000	249791	≤ 249964
gcut9	1000×1000	10	939600	5	979256	939600	939600
gcut10	1000×1000	20	937349	5	999918	960716	937349
gcut11	1000×1000	30	969709	5	1000000	986854	969709
gcut12	1000×1000	50	979521	5	1000000	993466	$\leq$ 993124
gcut13	3000×3000	32	≥ 8622498 ^b	≥ 13	9000000	8999510	$\leq$ 8917895
okp1	100×100	50	27718	11	29994	29344	27796
okp2	100×100	30	22502	11	24800	24800	23696
okp3	100×100	30	24019	11	26714	25884	24213
okp4	100×100	61	32893	10	33631	32950	32893
okp5	100×100	97	27923	8	29045	28201	27923

a. résultats obtenus par Caprara et Monaci

b. meilleure solution trouvée par Fekete et Schepers

TABEAU 1.17: Instances pour le problème de sac-à-dos de la littérature

Nous avons testé nos approches sur des instances classiques du problème de sac-à-dos sur deux dimensions de la littérature dont nous reportons quelques informations (taille de l'instance et la valeur de la solution optimale) dans le tableau 1.17. Les instances « ngcut », « gcut » et « cgcut » sont définies par Beasley [4], l'instance « wang20 » par Wang [61] et les instances « okp » par Fekete et Schepers [33]. Toutes les instances sont disponibles à cette adresse :

<http://www.ibr.cs.tu-bs.de/alg/packlib/instances.shtml>

De plus, les bornes duales classiques de la littérature sont reportées dans le tableau 1.17. Seule l'instance gcut13 reste non résolu dans l'état actuel des connaissances. Enfin, pour les instances « gcut » et « cgcut », « wang » et « okp », les bornes $U_{F_{DFF}}$ sont obtenues par une partie des coefficients possibles pour les MDFF précédemment définies afin d'éviter un temps de calcul trop important.

D'autres instances sont traitées dans ce mémoire. Il s'agit d'instances de contrôle de la réalisabilité d'un ensemble d'objets dont la cardinalité est donnée dans le tableau 1.18. La taille de la boîte est 20×20 pour chacune de ces instances. Ces instances ont été définies par Clautiaux et al. [21] et sont disponibles à l'adresse suivante :

<http://www2.lifl.fr/~clautiau/pmwiki/pmwiki.php?n=Research.Benchmarks>

Inst.	nombre d'objets	Inst.	nombre d'objets	Inst.	nombre d'objets
E02F17	17	E00N10	10	E08N15	15
E02F20	20	E00N15	15	E10N10	10
E02F22	22	E00N23	23	E10N15	15
E04F15	15	E02N20	20	E13N10	10
E04F17	17	E03N10	10	E13N15	15
E04F19	19	E03N15	15	E15N10	10
E04F20	20	E03N16	16	E15N15	15
E05F15	15	E03N17	17	E00X23	23
E05F18	18	E04N15	15	E05X15	15
E05F20	20	E04N17	17	E07X15	15
E07F15	15	E04N18	18	E10X15	15
E08F15	15	E05N15	15	E13X15	15
E20F15	15	E05N17	17		
E03X18	18	E07N10	10		
E20X15	15	E07N15	15		

(a) Instances réalisables

(b) Instances irréalisables

TABLEAU 1.18: Instances pour le problème de réalisabilité

Dans la suite de notre mémoire, nous fixons 30 minutes le temps limite de résolution pour chaque instance. De plus, pour chaque formulation notée F (où F est une des formulations introduites dans ce chapitre ou dans le chapitre suivant), nous associons la notation U_F^{PL} pour la borne de la relaxation linéaire associée à la formulation F et U_F^δ pour la borne obtenue avec δ_i entier.

*
* *

Sommaire

2.1	Modélisation utilisant une position relative des objets	44
2.2	Modélisations basées sur une discrétisation de l'espace	46
2.3	Modélisation basée sur un découpage par cliques maximales d'un placement	61
2.4	Décomposition des formulations . . .	73
2.5	Comparaison entre les différentes approches	81

CHAPITRE II

Formulations par programmation mathématique

Dans ce chapitre, nous allons exposer les différentes formulations de type programmation mathématique possibles pour le problème de sac-à-dos sur deux dimensions en exposant les avantages et les inconvénients de chacune d'elles. Pour chaque type de modélisation, nous définissons les différentes notations utilisées puis les différentes formulations. Nous présentons ici les formulations de la littérature ainsi qu'une nouvelle formulation compacte. Certaines des formulations exposées dans cette partie sont pseudo-polynomiales, d'autres sont compactes. Certaines formulations non valides au départ sont rendues valides. Inversement, certaines formulations sont relâchées afin de pouvoir effectuer une décomposition de Dantzig-Wolfe.

Dans la seconde partie, nous appliquons la décomposition de Dantzig-Wolfe à certaines de ces formulations afin d'obtenir des formulations de meilleure qualité mais de taille exponentielle. En effet, pour la plupart de ces formulations, des contraintes de sac-à-dos peuvent être vues comme sous-systèmes. Elles modélisent le non-dépassement de taille du conteneur pour chaque dimension à une position ou une coordonnée donnée. Alors que les autres contraintes définissent des contraintes liantes. Certaines de ces formulations ne représentent qu'une relaxation au problème (typiquement, les contraintes de consécutivité des colonnes sont ignorées).

Enfin, une étude comparative théorique et expérimentale des qualités de bornes pour les différentes formulations est effectuée.

2.1 Modélisation utilisant une position relative des objets

Une première approche pour modéliser le problème consiste à définir une position des objets les uns par rapport aux autres. Différents travaux [19, 26, 53, 54, 55] furent effectués pour modéliser le problème de cette manière. Il s'agit, dans la littérature actuelle, de la seule approche menant à des formulations compactes du problème. Elle a comme principal inconvénient d'introduire des « grands M » dans la formulation ainsi que des contraintes faibles de non-intersection des items. Ceci engendre des relaxations linéaires de très mauvaises qualités. En général, la solution optimale de la relaxation linéaire sélectionne tous les objets ($\delta_i = 1, \forall i \in \mathcal{I}$).

2.1.1 Lexique des variables pour les formulations définissant une position relative entre les objets

Nous commençons par définir l'ensemble des variables nécessaires à la modélisation du problème. D'abord, pour ce modèle, les variables x_i et y_i définies en section 1.1 sont utilisées afin de positionner chaque objet sélectionné précisément. De plus, une variable indiquant si un objet se trouve à gauche ou à droite d'un autre est rajoutée. C'est-à-dire, si deux objets ne partagent pas d'abscisse en commun alors forcément l'un est à gauche de l'autre. De même, une variable indiquant si un objet se trouve en dessous ou au dessus d'un autre est utilisée :

$$\left\{ \begin{array}{l} g_{ij} = \begin{cases} 1 & \text{si l'objet } i \text{ est à gauche de l'objet } j, \\ 0 & \text{sinon} \end{cases}, \quad \forall i, j \in \mathcal{I} (i \neq j), \\ d_{ij} = \begin{cases} 1 & \text{si l'objet } i \text{ est au dessous de l'objet } j, \\ 0 & \text{sinon} \end{cases}, \quad \forall i, j \in \mathcal{I} (i \neq j). \end{array} \right.$$

2.1.2 Formulation définissant une position relative entre les objets

Voici la formulation définie par Onodera, Taniguchi et Tamaru [53] :

FORMULATION 1: Formulation par position relative $[F^{c_{ij}d_{ij}}]$

$$[F^{c_{ij}d_{ij}}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.1a)$$

$$\text{s. t. } g_{ij} + g_{ji} + d_{ij} + d_{ji} \geq 1 - (1 - \delta_i) - (1 - \delta_j), \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.1b)$$

$$x_i + \mathbf{w}_i \leq x_j + \mathbf{W}(1 - g_{ij}), \quad \forall i, j \in \mathcal{I} (i \neq j), \quad (2.1c)$$

$$y_i + \mathbf{h}_i \leq y_j + \mathbf{H}(1 - d_{ij}), \quad \forall i, j \in \mathcal{I} (i \neq j), \quad (2.1d)$$

$$0 \leq x_i \leq \mathbf{W} - \mathbf{w}_i, \quad \forall i \in \mathcal{I}, \quad (2.1e)$$

$$0 \leq y_i \leq \mathbf{H} - \mathbf{h}_i, \quad \forall i \in \mathcal{I}, \quad (2.1f)$$

$$\delta_i, l_{ij}, b_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i \neq j). \quad (2.1g)$$

La contrainte (2.1b) impose que chaque paire d'objets (i, j) sélectionnée est soit l'un à gauche de l'autre ; soit l'un en dessous de l'autre. Donc, chaque paire d'objets ne peut s'intersecter dans le placement. Les contraintes (2.1c)-(2.1d) fixe une coordonnée à chaque objet sélectionné en fonction des positions relatives des objets les uns par rapport aux autres. En effet, si l'objet i est à gauche de l'objet j , alors la contrainte (2.1c) impose que l'abscisse de l'objet j se situe après l'abscisse de fin de l'objet i . De même pour les ordonnées. Les contraintes (2.1e)-(2.1f) impose que les coordonnées des objets ne peut dépasser la taille de la boîte. Donc, cette contrainte impose le non-dépassement du conteneur.

Lemme 2.1

Toute solution entière réalisable de la formulation $F^{c_{ij}d_{ij}}$ peut être convertie en une solution réalisable de même valeur selon la définition 1.1.

Preuve. Soit une solution entière de la formulation $F^{c_{ij}d_{ij}}$, notée (δ, c, d, x, y) .

Montrons alors que, $\forall i \in \mathcal{I}' = \{i \in \mathcal{I} : \delta_i = 1\}$, les coordonnées (x_i, y_i) de la formulation $F^{c_{ij}d_{ij}}$ vérifient les conditions (1.1a)-(1.1b).

D'après les contraintes (2.1e)-(2.1f), la condition (1.1a) est vérifiée par (x, y) .

De plus, pour tout $i, j \in \mathcal{I}' (i \neq j)$, nous avons $g_{ij} + g_{ji} + d_{ij} + d_{ji} \geq 1$, d'après la contrainte (2.1b). Comme la solution est entière, nous rencontrons au moins un de ces quatre cas :

- Soit $g_{ij} = 1$, alors $x_i + \mathbf{w}_i \leq x_j$ d'après la contrainte (2.1c),
- Soit $g_{ji} = 1$, alors $x_j + \mathbf{w}_j \leq x_i$ d'après la contrainte (2.1c),
- Soit $d_{ij} = 1$, alors $y_i + \mathbf{h}_i \leq y_j$ d'après la contrainte (2.1d),
- Soit $d_{ji} = 1$, alors $y_j + \mathbf{h}_j \leq y_i$ d'après la contrainte (2.1d).

Donc, la condition (1.1b) est vérifiée pour (x, y) . □

Lemme 2.2

Toute solution réalisable selon la définition 1.1 peut être convertie en une solution entière réalisable de même valeur pour la formulation $F^{c_{ij}d_{ij}}$.

Preuve. Soit une solution vérifiant les conditions (1.1a)-(1.1b).

Donc, pour tout objet $i \in \mathcal{I}'$, notons (x_i, y_i) la coordonnée associée à l'objet i .

Pour tout $i \in \mathcal{I}'$, posons $\delta_i = 1$ et pour tout $i \in \mathcal{I} \setminus \mathcal{I}'$, posons $\delta_i = 0$.

De plus, pour tout $i, j \in \mathcal{I}' (i \neq j)$, si $x_i + \mathbf{w}_i \leq x_j$ alors posons $g_{ij} = 1$, sinon posons $g_{ij} = 0$. Et si $y_i + \mathbf{h}_i \leq y_j$ alors posons $d_{ij} = 1$, sinon posons $d_{ij} = 0$. Si i et/ou $j \in \mathcal{I} \setminus \mathcal{I}'$, posons alors $g_{ij} = d_{ij} = 0$. Les variables x et y ne changent pas.

Montrons alors que les contraintes (2.1a)-(2.1g) sont vérifiées par cette solution.

La contrainte (2.1b) est vérifiée car :

- Si i et $j \in \mathcal{I}'$, alors, d'après la condition (1.1b), $g_{ij} + g_{ji} + d_{ij} + d_{ji} \geq 1 = \delta_i + \delta_j - 2$,
- Sinon, $g_{ij} + g_{ji} + d_{ij} + d_{ji} \geq 0 \geq \delta_i + \delta_j - 2$.

La contrainte (2.1c) est vérifiée car :

- Si i et $j \in \mathcal{I}'$ et $g_{ij} = 1$. Alors, par définition de g_{ij} , nous savons que $x_i + \mathbf{w}_i \leq x_j$,
- Sinon, $g_{ij} = 0$ avec i et $j \in \mathcal{I}$. D'après la condition (1.1a), $x_i + \mathbf{w}_i \leq \mathbf{W}$. Alors, $x_i + \mathbf{w}_i \leq \mathbf{W} \leq x_j + \mathbf{W}$.

Similairement, nous démontrons que la contrainte (2.1d) est vérifiée.

Les contraintes (2.1e)-(2.1f) sont vérifiées d'après la condition (1.1a).

Enfin, comme l'objectif reste inchangé, nous en déduisons que la solution (δ, c, d, x, y) est une solution entière réalisable pour la formulation $F^{c_{ij}d_{ij}}$. □

Corrolaire 2.3

La formulation $F^{c_{ij}d_{ij}}$ est valide.

Les contraintes suivantes sont des contraintes permettant d'éliminer des solutions symétriques pour la formulation $F^{c_{ij}d_{ij}}$. La formulation est alors appelée $F_{renf}^{c_{ij}d_{ij}}$.

$$g_{ij} + g_{ji} \leq \min\{\delta_i, \delta_j\}, \quad \forall i, j \in \mathcal{I}(i < j), \quad (2.2a)$$

$$d_{ij} + d_{ji} \leq \min\{\delta_i, \delta_j\}, \quad \forall i, j \in \mathcal{I}(i < j), \quad (2.2b)$$

$$g_{ij} + g_{ji'} - 1 \leq g_{ii'}, \quad \forall i, i', j \in \mathcal{I}(i \neq j \neq i'), \quad (2.2c)$$

$$d_{ij} + d_{ji'} - 1 \leq d_{ii'}, \quad \forall i, i', j \in \mathcal{I}(i \neq j \neq i'). \quad (2.2d)$$

Lemme 2.4

Les contraintes (2.2a)-(2.2b) sont des contraintes valides pour la formulation $F^{c_{ij}d_{ij}}$.

Preuve. Nous démontrons que les contraintes (2.2a)-(2.2b) sont vérifiées par l'absurde. Supposons qu'il existe une solution entière réalisable de $F^{c_{ij}d_{ij}}$ ne vérifiant pas la contrainte (2.2a). Alors, il existe un couple d'objets $i, j \in \mathcal{I}'$ avec $i < j$ tel que $g_{ij} + g_{ji} > 1$. Donc, $g_{ij} = g_{ji} = 1$. Donc, d'après la contrainte (2.1c), nous avons $x_i < x_j$ et $x_j < x_i$. C'est absurde. Donc, la contrainte (2.2a) est vérifiée.

De manière similaire, nous démontrons la validité de la contrainte (2.2b). $\square$

Lemme 2.5

Les contraintes (2.2c)-(2.2d) sont des contraintes valides pour la formulation $F^{c_{ij}d_{ij}}$.

Preuve. Soit une solution entière réalisable de la formulation $F^{c_{ij}d_{ij}}$. Soit $i, i', j \in \mathcal{I}$ avec $i \neq i' \neq j$.

Si $g_{ij} = 0$ ou $g_{ji'} = 0$ alors la contrainte (2.2c) impose $g_{ii'} \geq 0$. Donc, elle est vérifiée.

Si $g_{ij} = g_{ji'} = 1$, alors la contrainte (2.2c) impose $g_{ii'} = 1$. La solution reste valide car la contrainte (2.1c) pour les couples (i, j) et (j, i') implique les inégalités suivantes :

$x_i + \mathbf{w}_i \leq x_j \leq x_j + \mathbf{w}_j \leq x_{i'}$. Donc, la contrainte (2.1c) est valide pour (i, i') , d'où la validité de la contrainte (2.2c). De même, nous montrons que la contrainte (2.2d) est valide. $\square$

2.2 Modélisations basées sur une discrétisation de l'espace

Historiquement, les premières formulations proposées dans la littérature [4, 5, 41] utilisaient une discrétisation de l'espace afin de déterminer la position exacte des objets dans le conteneur et modéliser la définition d'un placement. Ce sont des formulations de taille pseudo-polynomiale puisque le nombre de variables dépend de la taille du conteneur. Cependant, contrairement aux formulations compactes, la contrainte de non dépassement du conteneur est définie par les domaines des variables. Cela évite d'avoir des « grands M » dans les formulations. Les relaxations linéaires sont alors de meilleure qualité.

Soit $\mathcal{X}$ l'ensemble des abscisses possibles pour le placement des objets. Typiquement, $\mathcal{X} = \{0, \dots, \mathbf{W} - 1\}$. De même, $\mathcal{Y}$ est l'ensemble des ordonnées possibles pour le placement des objets. Typiquement, $\mathcal{Y} = \{0, \dots, \mathbf{H} - 1\}$.

A chaque objet $i \in \mathcal{I}$, nous définissons $\mathcal{X}_i = \{0, \dots, \mathbf{W} - \mathbf{w}_i\}$ comme étant l'ensemble des abscisses possibles pour l'objet i . De manière analogue, nous posons $\mathcal{Y}_i = \{0, \dots, \mathbf{H} - \mathbf{h}_i\}$ comme étant l'ensemble des ordonnées possibles pour l'objet i . Ces espaces de définition permettent de modéliser la contrainte de non-dépassement de la taille du conteneur. En effet, pour chaque objet i , les ensembles $\mathcal{X}_i$ et $\mathcal{Y}_i$ représentent les domaines de définition des variables de positionnement associées aux objets $i \in \mathcal{I}$.

2.2.1 Lexique des variables pour les formulations définissant les coordonnées selon une discrétisation de l'espace

Nous allons commencer par définir l'ensemble des variables utilisées par les formulations utilisant une discrétisation de l'espace. La première variable définie est une variable indiquant la coordonnée de position d'un objet sélectionné :

$$\Psi_{iab} = \begin{cases} 1 & \text{si le coin inférieur-gauche de l'objet } i \text{ se situe} \\ & \text{en coordonnée } (a, b) \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, (a, b) \in \mathcal{X}_i \times \mathcal{Y}_i.$$

Une variable indiquant si une coordonnée est occupée par un objet est aussi utilisée :

$$z_{iab} = \begin{cases} 1 & \text{si la coordonnée } (a, b) \text{ est occupée par l'objet } i \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, (a, b) \in \mathcal{X}_i \times \mathcal{Y}_i.$$

Cependant, le nombre de variables ainsi obtenu est beaucoup trop important. La définition de nouvelles variables séparant les décisions entre les abscisses et les ordonnées est utile :

$$\begin{aligned} \chi_{ia} &= \begin{cases} 1 & \text{si le coin inférieur-gauche de l'objet } i \text{ se situe en abscisse } a \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}_i, \\ \gamma_{ib} &= \begin{cases} 1 & \text{si le coin inférieur-gauche de l'objet } i \text{ se situe en ordonnée } b \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, b \in \mathcal{Y}_i. \end{aligned}$$

Des variables indiquant si l'abscisse (ou l'ordonnée) est occupée par un objet sont aussi ajoutées :

$$\begin{aligned} x_{ia} &= \begin{cases} 1 & \text{si l'abscisse } a \text{ est occupée par l'objet } i \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}_i, \\ y_{ib} &= \begin{cases} 1 & \text{si l'ordonnée } b \text{ est occupée par l'objet } i \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, b \in \mathcal{Y}_i. \end{aligned}$$

Ensuite, une variable indiquant si aucun objet n'occupe une coordonnée est aussi rajoutée :

$$z_{ab} = \begin{cases} 1 & \text{si la coordonnée } (a, b) \text{ est libre} \\ 0 & \text{sinon} \end{cases}, \quad \forall (a, b) \in \mathcal{X} \times \mathcal{Y}.$$

Enfin, les variables x_{ij} et y_{ij} définies en section 1.4 peuvent être utilisées pour savoir si deux objets partagent une coordonnée selon l'une des dimensions.

Remarque Chaque coordonnée est vue comme une bande de largeur un à remplir. ♦

2.2.2 Formulation avec des variables indiquant la coordonnée de positionnement de chaque objet sélectionné

Beasley [4] proposa la première formulation qui modélise la définition d'un placement réalisable entier. En effet, cette formulation utilise une variable choisissant la coordonnée de position de chaque objet sélectionné.

FORMULATION 2: Formulation de Beasley $[F^{\Psi_{iab}}]$

$$[F^{\Psi_{iab}}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.3a)$$

$$\text{s. t.} \quad \sum_{(a,b) \in \mathcal{X}_i \times \mathcal{Y}_i} \Psi_{iab} = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.3b)$$

$$\sum_{i \in \mathcal{I}} \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'} \leq 1, \quad \forall (a, b) \in \mathcal{X} \times \mathcal{Y}, \quad (2.3c)$$

$$\delta_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}, \quad (2.3d)$$

$$\Psi_{iab} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, (a, b) \in \mathcal{X}_i \times \mathcal{Y}_i. \quad (2.3e)$$

La contrainte (2.3b) impose que chaque objet i ne peut être placé qu'à une seule coordonnée au maximum si cet objet est sélectionné. Sinon, aucune position n'est affectée. La contrainte (2.3c) impose que chaque coordonnée (a, b) n'est occupée que par un seul objet au maximum. Donc, les objets ne peuvent pas se chevaucher.

Lemme 2.6

Toute solution entière réalisable de la formulation $F^{\Psi_{iab}}$ peut être convertie en une solution entière réalisable de même valeur selon la définition 1.4.

Preuve. Soit une solution entière, notée (δ, Ψ) , vérifiant les contraintes (2.3b)-(2.3e).

$\forall i \in \mathcal{I}' = \{i \in \mathcal{I} : \delta_i = 1\}$, en utilisant la contrainte (2.3b), il existe une coordonnée $(a, b) \in \mathcal{X}_i \times \mathcal{Y}_i$ telle que $\Psi_{iab} = 1$. Posons alors $(x_i, y_i) = (a, b) \in \mathbb{N}^2$.

Montrons alors que la solution (x, y) est une solution réalisable selon la définition 1.4.

$\forall i \in \mathcal{I}'$, comme $a \in \mathcal{X}_i$, nous avons $x_i + \mathbf{w}_i \leq \mathbf{W}$. De même, $b \in \mathcal{Y}_i$, donc $y_i + \mathbf{h}_i \leq \mathbf{H}$.

Donc, la condition (1.1a) est vérifiée.

De plus, la condition (1.1b) est vérifiée. En effet, nous le démontrons par l'absurde.

Supposons qu'il existe un couple d'objets $i, j \in \mathcal{I}'$ avec $i \neq j$ tel que :

$$[x_i, x_i + \mathbf{w}_i] \cap [x_j, x_j + \mathbf{w}_j] \neq \emptyset \quad \text{et} \quad [y_i, y_i + \mathbf{h}_i] \cap [y_j, y_j + \mathbf{h}_j] \neq \emptyset.$$

Il existe donc une coordonnée (a, b) tel que :

$$a \in [x_i, x_i + \mathbf{w}_i] \cap [x_j, x_j + \mathbf{w}_j] \quad \text{et} \quad b \in [y_i, y_i + \mathbf{h}_i] \cap [y_j, y_j + \mathbf{h}_j].$$

$$\text{Alors, } \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'} = 1 \text{ et } \sum_{\substack{a' \in \mathcal{X}_j: \\ a' = a - \mathbf{w}_j + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_j: \\ b' = b - \mathbf{h}_j + 1}}^b \Psi_{ja'b'} = 1.$$

C'est incompatible avec la contrainte (2.3c) d'où l'absurdité.

D'où, la condition (1.1a) est vérifiée par la solution (x, y) . De plus, $\sum_{i \in \mathcal{I}} \rho_i \delta_i = \sum_{i \in \mathcal{I}'} \rho_i$.

Donc, (x, y) est une solution réalisable de même objectif pour la définition 1.4. $\square$

Lemme 2.7

Toute solution entière réalisable selon la définition 1.4 peut être convertie en une solution entière réalisable de même valeur pour la formulation $F^{\Psi_{iab}}$.

Preuve. Soit une solution entière réalisable vérifiant les conditions (1.1a)-(1.1b).

Donc, pour tout objet $i \in \mathcal{I}'$, notons (x_i, y_i) la coordonnée associée à l'objet i .

Pour tout $i \in \mathcal{I}'$, posons $\delta_i = 1$, $\Psi_{ix_i y_i} = 1$ et $\Psi_{iab} = 0$, $\forall (a, b) \in \mathcal{X}_i \times \mathcal{Y}_i$ tel que $(a, b) \neq (x_i, y_i)$. Et, pour tout $i \in \mathcal{I} \setminus \mathcal{I}'$, posons $\delta_i = 0$ et $\Psi_{iab} = 0$, $\forall (a, b) \in \mathcal{X}_i \times \mathcal{Y}_i$.

Donc, $\sum_{(a,b) \in \mathcal{X}_i \times \mathcal{Y}_i} \Psi_{iab} = \delta_i$, $\forall i \in \mathcal{I}$. La contrainte (2.3b) est donc vérifiée.

Pour démontrer la validité de la contrainte (2.3b), il suffit de le démontrer par l'absurde.

En effet, nous supposons qu'il existe un couple $(a, b) \in \mathcal{X} \times \mathcal{Y}$ tel que :

$$\sum_{i \in \mathcal{I}} \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'} > 1$$

D'où, il existe au moins deux objets i et $j \in \mathcal{I}$ avec $i \neq j$ tel que :

$$\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'} = 1 \quad \text{et} \quad \sum_{\substack{a' \in \mathcal{X}_j: \\ a' = a - \mathbf{w}_j + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_j: \\ b' = b - \mathbf{h}_j + 1}}^b \Psi_{ja'b'} = 1.$$

Soit $(a_i, b_i) \in \mathcal{X}_i \times \mathcal{Y}_i$ tel que $\Psi_{ia_i b_i} = 1$, $a_i \in [a - \mathbf{w}_i + 1; a]$ et $b_i \in [b - \mathbf{h}_i + 1; b]$. Et, soit $(a_j, b_j) \in \mathcal{X}_j \times \mathcal{Y}_j$ tel que $\Psi_{ja_j b_j} = 1$, $a_j \in [a - \mathbf{w}_j + 1; a]$ et $b_j \in [b - \mathbf{h}_j + 1; b]$. Alors, (a_i, b_i) représente la coordonnée de l'objet i et (a_j, b_j) représente la coordonnée de j . Nous savons que $a \in [a_i; a_i + \mathbf{w}_i - 1] \cap [a_j; a_j + \mathbf{w}_j - 1]$ et $b \in [b_i; b_i + \mathbf{h}_i - 1] \cap [b_j; b_j + \mathbf{h}_j - 1]$. D'où, $[a_i; a_i + \mathbf{w}_i - 1] \cap [a_j; a_j + \mathbf{w}_j - 1] \neq \emptyset$ et $[b_i; b_i + \mathbf{h}_i - 1] \cap [b_j; b_j + \mathbf{h}_j - 1] \neq \emptyset$, c'est absurde. Donc, La contrainte (2.3c) est vérifiée. Comme l'objectif reste inchangé, alors la solution (δ, Ψ) trouvée est une solution réalisable pour la formulation $F^{\Psi_{iab}}$. $\square$

Corrolaire 2.8

La formulation $F^{\Psi_{iab}}$ est valide.

Pour cette formulation, nous avons ajouter de nouvelles inégalités valides ayant pour but de forcer la propriété « calée en bas à gauche » :

$$\sum_{j \in \mathcal{I}: j \neq i} \sum_{b \in \mathcal{Y}_j} \Psi_{j, a - \mathbf{w}_j, b} \geq \sum_{b \in \mathcal{Y}_i} \Psi_{iab}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}_i \setminus \{0\}, \quad (2.4a)$$

$$\sum_{j \in \mathcal{I}: j \neq i} \sum_{a \in \mathcal{X}_j} \Psi_{j, a, b - \mathbf{h}_j} \geq \sum_{a \in \mathcal{X}_i} \Psi_{iab}, \quad \forall i \in \mathcal{I}, b \in \mathcal{Y}_i \setminus \{0\}. \quad (2.4b)$$

La contrainte (2.4a) permet de vérifier que chaque objet sélectionné ne touchant pas le bord gauche du conteneur touche forcément un autre objet placé avant. La contrainte (2.4b) est similaire mais pour le bord inférieur de l'objet. L'ajout de telles contraintes permet l'élimination de solutions symétriques renforçant ainsi la formulation. Notons cette formulation renforcée $F_{renf}^{\Psi_{iab}}$.

Lemme 2.9

Les inégalités (2.4a)-(2.4b) sont des inégalités valides pour la formulation $F^{\Psi_{iab}}$.

Preuve. Soit une solution entière de $F^{\Psi_{iab}}$, notée (δ, Ψ) .

Si $i \in \mathcal{I} \setminus \mathcal{I}'$, d'après la contrainte (2.3b), les inégalités sont vérifiées.

Soit $i \in \mathcal{I}'$. D'après la preuve du lemme 2.6, nous contruisons une solution réalisable pour la définition 1.4. De plus, d'après le lemme 1.3, il existe un placement de l'ensemble $\mathcal{I}'$ avec la condition « calée en bas à gauche ». Nous appliquons alors la preuve du lemme 2.7 pour reconstruire une nouvelle solution $(\tilde{\delta}, \tilde{\Psi})$ réalisable pour la formulation $F^{\Psi_{iab}}$. Lors de la phase de reconstruction de la solution, nous rencontrons deux cas :

Si $x_i = 0$, alors $\sum_{b \in \mathcal{Y}_i} \tilde{\Psi}_{ib} = 1$. Et donc, la contrainte (2.4a) est vérifiée d'après la contrainte (2.3b). Sinon, il existe au moins un objet $j \in \mathcal{I}$ avec $j \neq i$ tel que $x_j + \mathbf{w}_j = x_i$. Alors, $\sum_{b \in \mathcal{Y}_i} \tilde{\Psi}_{ix_i b} = 1$ et $\sum_{b \in \mathcal{Y}_j} \tilde{\Psi}_{j, x_i - \mathbf{w}_j, b} = 1$. Donc, la contrainte (2.4a) est vérifiée.

Similairement, nous démontrons la validité de la contrainte (2.4b). $\square$

2.2.3 Formulation avec des variables indépendantes indiquant la coordonnée de positionnement de chaque objet sélectionné

Christophides et Hadjiconstantinou [41] proposent une formulation séparant la prise de décision du choix de l'abscisse et de l'ordonnée de chaque objet sélectionné. Ainsi, ce choix se fait sur deux variables entières distinctes. Une variable de non-occupation d'une position est alors rajoutée. Néanmoins, cette formulation ne modélise qu'une relaxation du problème :

FORMULATION 3: Formulation de Christophides et al. $[F_{relax}^{z_{ab}}]$

$$[F_{relax}^{z_{ab}}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.5a)$$

$$\text{s. t.} \quad \sum_{a \in \mathcal{X}_i} \chi_{ia} = \sum_{b \in \mathcal{Y}_i} \gamma_{ib} = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.5b)$$

$$\sum_{i \in \mathcal{I}} \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \mathbf{h}_i \chi_{ia'} + \sum_{b \in \mathcal{Y}} z_{ab} = \mathbf{H}, \quad \forall a \in \mathcal{X}, \quad (2.5c)$$

$$\sum_{i \in \mathcal{I}} \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \mathbf{w}_i \gamma_{ib'} + \sum_{a \in \mathcal{X}} z_{ab} = \mathbf{W}, \quad \forall b \in \mathcal{Y}, \quad (2.5d)$$

$$\sum_{a' \in \mathcal{X}: a' = a}^{a + \mathbf{w}_i - 1} \sum_{b' \in \mathcal{Y}: b' = b}^{b + \mathbf{h}_i - 1} z_{a'b'} \leq \mathbf{w}_i \mathbf{h}_i (2 - \chi_{ia} - \gamma_{ib}), \quad \forall i \in \mathcal{I}, (a, b) \in \mathcal{X}_i \times \mathcal{Y}_i \quad (2.5e)$$

$$\chi_{ia}, \gamma_{ib} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}_i, b \in \mathcal{Y}_i, \quad (2.5f)$$

$$z_{ab} \in \{0, 1\}, \quad \forall (a, b) \in \mathcal{X} \times \mathcal{Y}. \quad (2.5g)$$

La contrainte (2.5b) impose une abscisse et une ordonnée pour chaque objet i sélectionné. Dans le cas où l'objet i n'est pas sélectionné, alors aucune abscisse, ni d'ordonnée n'est affectée. Les contraintes (2.5c)-(2.5d) permettent de déterminer les coordonnées qui ne sont pas occupées par un objet. En effet, pour chaque abscisse a (resp. chaque ordonnée b), la contrainte (2.5c) (resp. (2.5d)) impose qu'un nombre limité et fixé de positions inoccupées en fonction des objets intersectant cette abscisse (resp. ordonnée) et respectant la hauteur (resp. largeur) de la boîte. La contrainte (2.5e) impose de ne pas affecter le mode inoccupé à une position (a, b) alors que la coordonnée est déjà occupée par un objet.

Cette formulation ne modélise qu'une relaxation du problème : la contrainte de non-intersection des objets n'est pas totalement modélisée car des solutions avec chevauchement des objets sont des solutions valides pour la formulation (par exemple la figure 2.1 représente un placement où les objets 1 et 2 sont positionnés en coordonnée $(0, 0)$).

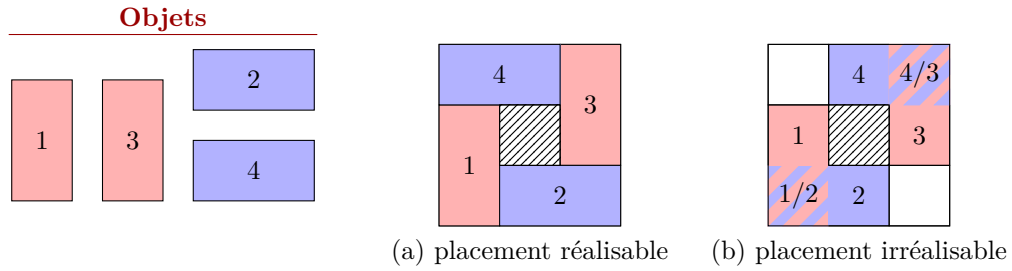


FIGURE 2.1: Exemple de placements réalisable et irréalisable vérifiant les contraintes (2.5b)-(2.5g) (zone hachurée : $z_{ab} = 1$)

Amaral et Letchford [2] ont rajouté des contraintes pour remédier à ce problème. Cependant, elles sont insuffisantes et trop nombreuses. Nous proposons les contraintes suivantes (inspirées de Boschetti et al. [12]) pour corriger les problèmes liés au chevauchement des objets. Ces contraintes permettent d'imposer que deux objets s'intersectent que sur une dimension au maximum en utilisant les variables x_{ij} et y_{ij} définies en section 1.1. La formulation obtenue par l'ajout de ces contraintes est notée $F^{z_{ab}}$:

$$\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'} + \sum_{\substack{a' \in \mathcal{X}_j: \\ a' = a - \mathbf{w}_j + 1}}^a \chi_{ja'} - 1 \leq x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), a \in \mathcal{X}, \quad (2.6a)$$

$$\sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib'} + \sum_{\substack{b' \in \mathcal{Y}_j: \\ b' = b - \mathbf{h}_j + 1}}^b \gamma_{jb'} - 1 \leq y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), b \in \mathcal{Y}, \quad (2.6b)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in \mathcal{I} (i < j). \quad (2.6c)$$

Lemme 2.10

Toute solution réalisable à la relaxation linéaire de la formulation $F^{\Psi_{iab}}$ peut être transposée en une solution réalisable de même valeur pour la relaxation linéaire de la formulation $F^{z_{ab}}$.

Preuve. Soit une solution réelle à la relaxation linéaire de la formulation $F^{\Psi_{iab}}$, notée $(\delta^{beas}, \Psi^{beas})$. Posons :

$$\begin{aligned} - \delta_i^{ch} &= \delta_i^{beas}, & \forall i \in \mathcal{I}, \\ - \chi_{ia}^{ch} &= \sum_{b \in \mathcal{Y}_i} \Psi_{iab}^{beas}, & \forall i \in \mathcal{I}, a \in \mathcal{X}_i, \\ - \gamma_{ib}^{ch} &= \sum_{a \in \mathcal{X}_i} \Psi_{iab}^{beas}, & \forall i \in \mathcal{I}, b \in \mathcal{Y}_i, \end{aligned}$$

$$- z_{ab}^{ch} = 1 - \sum_{i \in \mathcal{I}} \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'}^{beas}, \quad \forall (a, b) \in \mathcal{X} \times \mathcal{Y}.$$

Montrons alors que la solution $(\delta^{ch}, \chi^{ch}, \gamma^{ch}, z^{ch})$ est une solution réelle réalisable pour $F^{z_{ab}}$. Pour tout $i \in \mathcal{I}$, $\sum_{a \in \mathcal{X}_i} \chi_{ia}^{ch} = \sum_{a \in \mathcal{X}_i} \sum_{b \in \mathcal{Y}_i} \Psi_{iab}^{beas} = \sum_{b \in \mathcal{Y}_i} \gamma_{ib}^{ch}$. Donc, d'après la contrainte (2.3b), la contrainte (2.5b) est vérifiée.

Ensuite, par construction de la solution, nous avons, pour tout $a \in \mathcal{X}_i$:

$$\sum_{b \in \mathcal{Y}} z_{ab}^{ch} = \sum_{b \in \mathcal{Y}} \left(1 - \sum_{i \in \mathcal{I}} \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'}^{beas} \right) = H - \sum_{i \in \mathcal{I}} \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{b \in \mathcal{Y}_i} \mathbf{h}_i \Psi_{ia'b}^{beas}.$$

Donc, la contrainte (2.5c) est vérifiée par la solution. Similairement, nous montrons que la contrainte (2.5d) est vérifiée par la solution.

Pour tout $i \in \mathcal{I}$, $a \in \mathcal{X}_i$, $b \in \mathcal{Y}_i$, nous avons en utilisant la contrainte (2.3b) :

$$\chi_{ia}^{ch} + \gamma_{ib}^{ch} = \sum_{b' \in \mathcal{Y}_i} \Psi_{iab'}^{beas} + \sum_{a' \in \mathcal{X}_i} \Psi_{ia'b}^{beas} \leq \Psi_{iab}^{beas} + \sum_{a' \in \mathcal{X}_i} \sum_{b' \in \mathcal{Y}_i} \Psi_{ia'b'}^{beas} \leq \Psi_{iab}^{beas} + 1. \quad (1\star)$$

De plus, par construction, nous avons, pour tout $i \in \mathcal{I}$, $a \in \mathcal{X}$, $b \in \mathcal{Y}$,

$$z_{ab}^{ch} \leq 1 - \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{iab'}^{beas}. \quad (2\star)$$

En utilisant les inégalités (1 $\star$) et (2 $\star$), nous en déduisons pour tout $i \in \mathcal{I}$, $a \in \mathcal{X}$, $b \in \mathcal{Y}$:

$$\sum_{a' \in \mathcal{X}: a' = a}^{a + \mathbf{w}_i - 1} \sum_{b' \in \mathcal{Y}: b' = b}^{b + \mathbf{h}_i - 1} z_{a'b'}^{ch} \leq \mathbf{w}_i \mathbf{h}_i (1 - \Psi_{iab}^{beas}) \leq \mathbf{w}_i \mathbf{h}_i (2 - \chi_{ia}^{ch} - \gamma_{ib}^{ch}).$$

Donc, la contrainte (2.6a) est vérifiée par la solution.

Enfin, comme pour la preuve de l'inégalité (1 $\star$), nous montrons que pour tout $a \in \mathcal{X}$, $b \in \mathcal{Y}$:

$$\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'}^{ch} + \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib'}^{ch} \leq \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{iab'}^{beas} + 1. \quad (3\star)$$

Donc, en utilisant la contrainte (2.3c), nous avons pour tout $i, j \in \mathcal{I}$ ($i < j$), $a \in \mathcal{X}$, $b \in \mathcal{Y}$:

$$\begin{aligned} & \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'}^{ch} + \sum_{\substack{a' \in \mathcal{X}_j: \\ a' = a - \mathbf{w}_j + 1}}^a \chi_{ja'}^{ch} + \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib'}^{ch} + \sum_{\substack{b' \in \mathcal{Y}_j: \\ b' = b - \mathbf{h}_j + 1}}^b \gamma_{jb'}^{ch} \\ & \leq \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{iab'}^{beas} + \sum_{\substack{a' \in \mathcal{X}_j: \\ a' = a - \mathbf{w}_j + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_j: \\ b' = b - \mathbf{h}_j + 1}}^b \Psi_{jab'}^{beas} + 2 \leq 3. \end{aligned}$$

Nous déduisons que les contraintes (2.6a)-(2.6c) sont vérifiées. Nous avons construit une solution réelle pour la relaxation linéaire de la formulation $F^{z_{ab}}$ de même objectif. $\square$

Corrolaire 2.11

La borne duale obtenue par la relaxation linéaire de la formulation $F^{\Psi_{iab}}$ est plus petite ou égale à celle obtenue par la relaxation linéaire de la formulation $F^{z_{ab}}$.

Remarque Comme la formulation $F^{\Psi_{iab}}$ est valide et en utilisant les inégalités (2★) et (3★) de la preuve du lemme 2.10, chaque solution entière construite à partir d'une solution entière $F^{\Psi_{iab}}$ vérifie l'inégalité suivante :

$$z_{ab} \leq \left(2 - \sum_{\substack{a' \in \mathcal{X}: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia} - \sum_{\substack{b' \in \mathcal{Y}: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib} \right), \quad \forall i \in \mathcal{I}, (a, b) \in \mathcal{X} \times \mathcal{Y}. \quad (2.7)$$

C'est donc bien une inégalité valide plus forte que la contrainte (2.5e). $\blacklozenge$

Lemme 2.12

Toute solution entière réalisable de la formulation $F^{z_{ab}}$ peut être transposée en une solution entière réalisable de même valeur pour la formulation $F^{\Psi_{iab}}$.

Preuve. Soit une solution entière de la formulation $F^{z_{ab}}$, notée $(\delta^{ch}, \chi^{ch}, \gamma^{ch}, z^{ch}, x^{ch}, y^{ch})$. Posons :

$$\begin{aligned} - \delta_i^{beas} &= \delta_i^{ch}, & \forall i \in \mathcal{I}, \\ - \Psi_{iab}^{beas} &= \chi_{ia}^{ch} \gamma_{ib}^{ch}, & \forall i \in \mathcal{I}, (a, b) \in \mathcal{X}_i \times \mathcal{Y}_i. \end{aligned}$$

Montrons alors que la solution $(\delta^{beas}, \Psi^{beas})$ est une solution valide de la formulation $F^{\Psi_{iab}}$. Pour tout $i \in \mathcal{I}$, d'après la contrainte (2.5b), nous avons :

$$\sum_{a \in \mathcal{X}_i} \sum_{b \in \mathcal{Y}_i} \Psi_{iab}^{beas} = \left(\sum_{a \in \mathcal{X}_i} \chi_{ia}^{ch} \right) \left(\sum_{b \in \mathcal{Y}_i} \gamma_{ib}^{ch} \right) = (\delta_i^{ch})^2.$$

Comme la solution est entière, $(\delta_i^{ch})^2 = \delta_i^{ch}$, $\forall i \in \mathcal{I}$. Donc, la contrainte (2.3b) est vérifiée. De plus, d'après les contraintes (2.6a)-(2.6c) et comme la solution est entière, nous avons pour tout $i, j \in \mathcal{I} (i < j)$, $a \in \mathcal{X}$, $b \in \mathcal{Y}$:

$$\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \chi_{ia'}^{ch} \gamma_{ib'}^{ch} + \sum_{\substack{a' \in \mathcal{X}_j: \\ a' = a - \mathbf{w}_j + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_j: \\ b' = b - \mathbf{h}_j + 1}}^b \chi_{ja'}^{ch} \gamma_{jb'}^{ch} \leq 1. \quad (\star)$$

Supposons que la contrainte (2.3b) n'est pas vérifiée. Donc, il existe $(a, b) \in \mathcal{X} \times \mathcal{Y}$ avec :

$$\sum_{i \in \mathcal{I}} \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'}^{beas} > 1.$$

Par l'intégralité de la solution $(\delta^{beas}, \Psi^{beas})$, il existe $i, j \in \mathcal{I}$ avec $i < j$ tel que :

$$\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'}^{beas} + \sum_{\substack{a' \in \mathcal{X}_j: \\ a' = a - \mathbf{w}_j + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_j: \\ b' = b - \mathbf{h}_j + 1}}^b \Psi_{ja'b'}^{beas} > 1.$$

C'est absurde avec l'inégalité $(\star)$. Donc, la contrainte est vérifiée. Nous avons donc construit une solution entière réalisable pour la formulation $F^{z_{ab}}$ de même objectif. $\square$

Corrolaire 2.13

La formulation $F^{z_{ab}}$ est valide : les contraintes (2.5a)-(2.5b), (2.5f) et (2.6a)-(2.6c) sont suffisantes pour formuler le problème.

Une autre approche consiste à utiliser une variable d'occupation d'une coordonnée par un objet à la place d'une variable de non occupation. Une première formulation quadratique peut être définie :

FORMULATION 4: Nouvelle formulation quadratique $[FQ^{z_{iab}}]$

$$[FQ^{z_{iab}}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.8a)$$

$$\text{s. t.} \quad \sum_{a \in \mathcal{X}_i} \chi_{ia} = \sum_{b \in \mathcal{Y}_i} \gamma_{ib} = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.8b)$$

$$\left(\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'} \right) \left(\sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib'} \right) = z_{iab}, \quad \forall i \in \mathcal{I}, (a, b) \in \mathcal{X} \times \mathcal{Y}, \quad (2.8c)$$

$$\sum_{i \in \mathcal{I}} z_{iab} \leq 1, \quad \forall (a, b) \in \mathcal{X} \times \mathcal{Y}, \quad (2.8d)$$

$$\chi_{ia}, \gamma_{ib} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}_i, b \in \mathcal{Y}_i, \quad (2.8e)$$

$$z_{iab} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, (a, b) \in \mathcal{X} \times \mathcal{Y}. \quad (2.8f)$$

La contrainte (2.8b) permet d'imposer une seule abscisse et une seule ordonnée pour chaque objet i sélectionné ($\delta_i = 1$). Si l'objet i n'est pas sélectionné ($\delta_i = 0$), alors aucune coordonnée n'est affectée à l'objet. La contrainte (2.8c) permet de fixer les coordonnées (a, b) occupées par un objet. La contrainte (2.8d) permet de vérifier que chaque position (a, b) n'est occupée que par un seul objet à la fois. Donc, les objets ne s'intersectent pas.

Cette formulation quadratique pourrait être résolue par un solveur de type SDP. Cependant, cette étude ne rentre pas dans le cadre de ce mémoire. Donc, cette formulation n'est pas étudiée. Elle servira seulement à démontrer la validité d'autres formulations obtenues par linéarisation de $FQ^{z_{iab}}$. Par exemple, Beasley et Mingozzi [5] proposent une formulation qui est en réalité une linéarisation de la formulation $FQ^{z_{iab}}$.

FORMULATION 5: Formulation de Beasley et Mingozzi $[F^{z_{iab}}]$

$$[F^{z_{iab}}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.9a)$$

$$\text{s. t.} \quad \sum_{a \in \mathcal{X}_i} \chi_{ia} = \sum_{b \in \mathcal{Y}_i} \gamma_{ib} = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.9b)$$

$$\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \mathbf{h}_i \chi_{ia'} = \sum_{b \in \mathcal{Y}} z_{iab}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}, \quad (2.9c)$$

$$\sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \mathbf{w}_i \gamma_{ib'} = \sum_{a \in \mathcal{X}} z_{iab}, \quad \forall i \in \mathcal{I}, b \in \mathcal{Y}, \quad (2.9d)$$

$$\sum_{i \in \mathcal{I}} z_{iab} \leq 1 \quad \forall (a, b) \in \mathcal{X} \times \mathcal{Y}, \quad (2.9e)$$

$$\delta_i, \chi_{ia}, \gamma_{ib} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}_i, b \in \mathcal{Y}_i, \quad (2.9f)$$

$$z_{iab} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, (a, b) \in \mathcal{X} \times \mathcal{Y}. \quad (2.9g)$$

Remarque La contrainte (2.8c) de la formulation $FQ^{z_{iab}}$ peut-être linéarisée de manière classique :

$$\left(\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'} \right) + \left(\sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib'} \right) - 1 \leq z_{iab}, \quad \forall i \in \mathcal{I}, (a, b) \in \mathcal{X} \times \mathcal{Y}, \quad (2.10)$$

Cependant la linéarisation offre une moins bonne relaxation linéaire que la formulation $F^{z_{iab}}$ et génère un nombre de contraintes plus important. Elle n'est donc pas étudiée. ♦

Les contraintes (2.9b) et (2.9e) sont identiques aux contraintes (2.8b)-(2.8d). Les contraintes (2.9c)-(2.9d) permettent de déterminer les coordonnées occupées par les différents objets sélectionnés. En effet, d'après la contrainte (2.9c) (resp. (2.9d)), pour chaque abscisse a (resp. ordonnée b), si un objet intersecte l'abscisse a (resp. l'ordonnée b), la contrainte fixe un nombre d'ordonnées (resp. abscisses) équivalent à la hauteur (largeur) de l'objet.

Lemme 2.14

Toute solution entière réalisable de la formulation $F^{\Psi_{iab}}$ peut être transposée en une solution entière réalisable de même valeur pour la formulation $FQ^{z_{iab}}$.

Preuve. Soit une solution entière de la formulation $F^{\Psi_{iab}}$, notée $(\delta^{beas}, \Psi^{beas})$. Posons :

$$\begin{aligned} - \delta_i^{quad} &= \delta_i^{beas}, & \forall i \in \mathcal{I}, \\ - \chi_{ia}^{quad} &= \sum_{b \in \mathcal{Y}_i} \Psi_{iab}^{beas}, & \forall i \in \mathcal{I}, a \in \mathcal{X}_i, \\ - \gamma_{ib}^{quad} &= \sum_{a \in \mathcal{X}_i} \Psi_{iab}^{beas}, & \forall i \in \mathcal{I}, b \in \mathcal{Y}_i, \\ - z_{iab}^{quad} &= \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'}^{beas}, & \forall i \in \mathcal{I}, (a, b) \in \mathcal{X} \times \mathcal{Y}. \end{aligned}$$

Montrons alors que la solution $(\delta^{quad}, \chi^{quad}, \gamma^{quad}, z^{quad})$ est une solution entière réalisable pour la formulation $FQ^{z_{iab}}$.

D'abord, pour tout $i \in \mathcal{I}$, $\sum_{a \in \mathcal{X}_i} \chi_{ia}^{quad} = \sum_{a \in \mathcal{X}_i} \sum_{b \in \mathcal{Y}_i} \Psi_{iab}^{beas} = \sum_{b \in \mathcal{Y}_i} \gamma_{ib}^{quad}$. Donc, d'après la contrainte (2.3b), la contrainte (2.8b) est vérifiée. De plus, comme la solution de $F^{\Psi_{iab}}$ est entière, nous en déduisons que pour tout $i \in \mathcal{I}, (a, b) \in \mathcal{X} \times \mathcal{Y}$, $\Psi_{iab} = \chi_{ia} \gamma_{ib}$ et donc :

$$\left(\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'}^{quad} \right) \left(\sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib'}^{quad} \right) = \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'}^{beas} = z_{iab}^{quad}$$

Donc, la contrainte (2.8c) est vérifiée. En utilisant la contrainte (2.3c), nous déduisons que la contrainte (2.8d) est vérifiée. Enfin, les bornes des variables sont vérifiées d'après la construction de la solution et la valeur de l'objectif reste identique. Donc, toute solution entière de la formulation $F^{\Psi_{iab}}$ peut être convertie en une solution entière réalisable de la formulation $FQ^{z_{iab}}$. □

Lemme 2.15

Toute solution entière réalisable de la formulation $FQ^{z_{iab}}$ peut être transposée en une solution entière réalisable de même valeur pour la formulation $F^{\Psi_{iab}}$.

Preuve. Soit une solution entière de la formulation $FQ^{z_{iab}}$, notée $(\delta^{quad}, \chi^{quad}, \gamma^{quad}, z^{quad})$.

Posons :

$$\begin{aligned} - \delta_i^{beas} &= \delta_i^{quad}, & \forall i \in \mathcal{I}, \\ - \Psi_{iab}^{beas} &= \chi_{ia}^{quad} \gamma_{ib}^{quad}, & \forall i \in \mathcal{I}, (a, b) \in \mathcal{X}_i \times \mathcal{Y}_i. \end{aligned}$$

Montrons alors que la solution $(\delta^{beas}, \Psi^{beas})$ est une solution réalisable de la formulation $F\Psi_{iab}$.

Pour tout $i \in \mathcal{I}$, d'après la contrainte (2.8b), nous avons :

$$\sum_{a \in \mathcal{X}_i} \sum_{b \in \mathcal{Y}_i} \Psi_{iab}^{beas} = \left(\sum_{a \in \mathcal{X}_i} \chi_{ia}^{quad} \right) \left(\sum_{b \in \mathcal{Y}_i} \gamma_{ib}^{quad} \right) = (\delta_i^{quad})^2.$$

Comme la solution est entière, $(\delta_i^{quad})^2 = \delta_i^{quad}$, $\forall i \in \mathcal{I}$. Donc, la contrainte (2.3b) est vérifiée.

De plus, pour tout $(a, b) \in \mathcal{X} \times \mathcal{Y}$,

$$\sum_{i \in \mathcal{I}} \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'}^{beas} = \sum_{i \in \mathcal{I}} \left(\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'}^{quad} \right) \left(\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \gamma_{ib'}^{quad} \right)$$

Or, en utilisant les contraintes (2.8c) et (2.8d), nous retrouvons la contrainte (2.3c).

Enfin, les bornes des variables sont vérifiées d'après la construction de la solution et la valeur de l'objectif reste la même. Donc, toute solution entière de la formulation $FQ^{z_{iab}}$ peut être convertie en une solution entière réalisable de la formulation $F\Psi_{iab}$. $\square$

Lemme 2.16

Toute solution réalisable de la relaxation linéaire de la formulation $FQ^{z_{iab}}$ peut être transposée en une solution réalisable de même valeur pour la relaxation linéaire de la formulation $F^{z_{iab}}$.

Preuve. La formulation $F^{z_{iab}}$ peut être obtenue par des combinaisons linéaires des contraintes de la formulation $FQ^{z_{iab}}$: les contraintes (2.9b) et (2.9e) sont identiques aux contraintes (2.8b) et (2.8d). De plus, les contraintes (2.9c)-(2.9d) peuvent être obtenues à partir de la contrainte (2.8c). En effet, si nous sommions la contrainte (2.8c) sur l'ensemble des ordonnées $b \in \mathcal{Y}$, alors nous obtenons la contrainte (2.9c) :

$$\begin{aligned} \sum_{b \in \mathcal{Y}} \left(\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'} \right) \left(\sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib'} \right) &= \left(\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'} \right) \left(\mathbf{h}_i \sum_{b \in \mathcal{Y}_i} \gamma_{ib} \right) \\ &= \mathbf{h}_i \delta_i \left(\sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'} \right) = \sum_{b \in \mathcal{Y}} z_{iab}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}. \end{aligned}$$

De même, en sommant sur les indices $a \in \mathcal{X}$, nous obtenons la contrainte (2.9d). Donc, toute solution de la formulation $FQ^{z_{iab}}$ est une solution de la formulation $F^{z_{iab}}$ de même valeur objectif. $\square$

Lemme 2.17

Toute solution entière réalisable de la formulation $F^{z_{iab}}$ peut être transposée en une solution entière réalisable de même valeur pour la formulation $FQ^{z_{iab}}$.

Preuve. Soit une solution entière de la formulation $F^{z_{iab}}$, notée $(\delta, \chi, \gamma, z)$.

Les contraintes (2.8b) et (2.8d) sont vérifiées d'après les contraintes (2.9b) et (2.9e).

Il faut donc vérifier que la contrainte (2.8c) est satisfaite par la solution entière $(\delta, \chi, \gamma, z)$. Pour tout $i \in \mathcal{I}$, soit $\delta_i = 0$ et dans ce cas la contrainte (2.8c) est forcément vérifiée; soit $\delta_i = 1$ et alors, il existe $a_i \in \mathcal{X}_i$ et $b_i \in \mathcal{Y}_i$ tel que $\chi_{ia_i} = 1$ et $\gamma_{ib_i} = 1$. Alors, pour tout $(a, b) \in \mathcal{X} \times \mathcal{Y}$, nous rencontrons l'un des quatre cas suivants (cf. les différentes zones de la figure 2.2) :

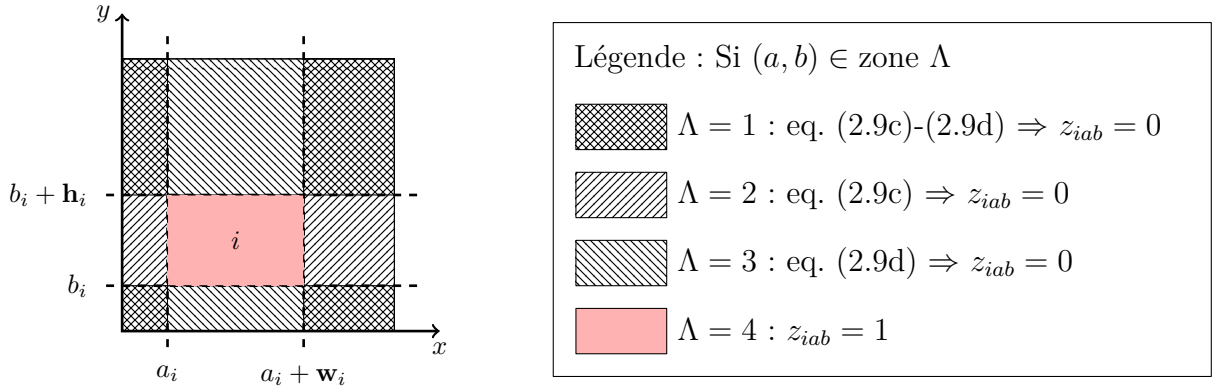


FIGURE 2.2: Différents cas rencontrés pour un objet sélectionné i placé en coordonnée (a_i, b_i) ($\chi_{ia_i} = 1$ et $\gamma_{ib_i} = 1$)

Zone 1: Si $a_i \notin [a - w_i + 1; a]$ et $b_i \notin [b - h_i + 1; b]$; alors, d'après les contraintes (2.9c)-(2.9d), nous en déduisons que $z_{iab} = 0$. Donc, la contrainte (2.8c) est vérifiée par la solution.

Zone 2: Si $a_i \notin [a - w_i + 1; a]$ et $b_i \in [b - h_i + 1; b]$; alors, d'après la contrainte (2.9c), nous en déduisons que $z_{iab} = 0$. Donc, la contrainte (2.8c) est vérifiée par la solution.

Zone 3: Si $a_i \in [a - w_i + 1; a]$ et $b_i \notin [b - h_i + 1; b]$; alors, d'après la contrainte (2.9d), nous en déduisons que $z_{iab} = 0$. Donc, la contrainte (2.8c) est vérifiée par la solution.

Zone 4: Si $a_i \in [a - w_i + 1; a]$ et $b_i \in [b - h_i + 1; b]$; alors, il suffit de montrer que $z_{iab} = 1$. En effet, d'après la contrainte (2.9c) :

$$\sum_{b' \in \mathcal{Y}} z_{iab'} = h_i. \quad (\star)$$

De plus, d'après la contrainte (2.9b), pour tout $b' \in \mathcal{Y}$ avec $b' \neq b_i$, nous avons $\gamma_{ib'} = 0$. En particulier, en utilisant la contrainte (2.9d) pour tout $b' \in \mathcal{Y}$ tel que $b' \in [0; b_i] \cup [b_i + h_i - 1; H]$, nous en déduisons que $\sum_{a' \in \mathcal{X}} z_{ia'b'} = 0$.

Donc, pour tout $b' \in \mathcal{Y}$ tel que $b' \notin [b_i; b_i + h_i - 1]$, nous avons $z_{iab'} = 0$.

Ainsi, nous remplaçons l'égalité $(\star)$ par :

$$\sum_{\substack{b' \in \mathcal{Y}: \\ b' \in [b_i; b_i + h_i - 1]}} z_{iab'} = h_i.$$

Alors, $z_{iab} = 1$ comme $b \in [b_i; b_i + h_i + 1]$. La contrainte (2.8c) est donc vérifiée.

Donc, toute solution entière vérifiant les contraintes (2.9b)-(2.9g) est une solution entière réalisable pour les contraintes (2.8b)-(2.8f) de même valeur objectif. $\square$

Corrolaire 2.18

Les formulations $FQ^{z_{iab}}$ et $F^{z_{iab}}$ sont valides.

Lemme 2.19

La borne duale obtenue par la relaxation linéaire de la formulation $F^{\Psi_{iab}}$ est plus petite ou égale à celle obtenue par la relaxation linéaire de la formulation $F^{z_{iab}}$.

Preuve. Soit une solution réelle réalisable de la relaxation linéaire pour la formulation $F^{\Psi_{iab}}$, notée $(\delta^{beas}, \Psi^{beas})$. Posons :

$$\begin{aligned} - \delta_i^{bm} &= \delta_i^{beas}, & \forall i \in \mathcal{I}, \\ - \chi_{ia}^{bm} &= \sum_{b \in \mathcal{Y}_i} \Psi_{iab}^{beas}, & \forall i \in \mathcal{I}, a \in \mathcal{X}_i, \\ - \gamma_{ib}^{bm} &= \sum_{a \in \mathcal{X}_i} \Psi_{iab}^{beas}, & \forall i \in \mathcal{I}, b \in \mathcal{Y}_i, \\ - z_{iab}^{bm} &= \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'}^{beas}, & \forall i \in \mathcal{I}, (a, b) \in \mathcal{X} \times \mathcal{Y}. \end{aligned}$$

En réutilisant la preuve du lemme 2.14, nous montrons rapidement que les contraintes (2.9b) et (2.9e) sont vérifiées pour toutes solutions réelles de la relaxation linéaire.

Il suffit de montrer que les contraintes (2.9c)-(2.9d) sont vérifiées. Par construction de la solution, nous avons, pour tout $i \in \mathcal{I}, a \in \mathcal{X}_i$:

$$\sum_{b \in \mathcal{Y}} z_{iab} = \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{b \in \mathcal{Y}} \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \Psi_{ia'b'}^{beas} = \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \sum_{b \in \mathcal{Y}_i} \mathbf{h}_i \Psi_{ia'b}^{beas} = \mathbf{h}_i \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'}.$$

Donc, la contrainte (2.9c) est vérifiée par la solution. Similairement, nous montrons que la contrainte (2.9d) est vérifiée. Donc, nous avons construit une solution réalisable pour la formulation $F^{z_{iab}}$ de même objectif. $\square$

Remarque Sans perte de généralité, l'intégralité de la variable z_{iab} peut être relâché pour les formulations $FQ^{z_{iab}}$ et $F^{z_{iab}}$, $\forall i \in \mathcal{I}, (a, b) \in \mathcal{X} \times \mathcal{Y}$. $\blacklozenge$

Pour les formulations $F^{z_{ab}}$ et $F^{z_{iab}}$, nous ajoutons de nouvelles inégalités valides similaires à celles développées pour la formulation $F^{\Psi_{iab}}$. Elles permettent de modéliser la condition « calée en bas à gauche ». Ces contraintes permettent l'élimination de solutions symétriques. Notons les formulations obtenues $F_{renf}^{z_{ab}}$ et $F_{renf}^{z_{iab}}$. Des contraintes similaires mais offrant de moins bonnes relaxations linéaires ont été introduites par [8] :

$$\sum_{j \in \mathcal{I}: j \neq i} \chi_{j, a - \mathbf{w}_j} \geq \chi_{ia}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}_i \setminus \{0\}, \quad (2.11a)$$

$$\sum_{j \in \mathcal{I}: j \neq i} \gamma_{j, b - \mathbf{h}_j} \geq \gamma_{ia}, \quad \forall i \in \mathcal{I}, b \in \mathcal{Y}_i \setminus \{0\}. \quad (2.11b)$$

Lemme 2.20

Les inégalités (2.11a)-(2.11b) sont valides pour la formulation $F^{z_{ab}}$.

Preuve. Pour toute solution de $F^{\Psi_{iab}}$ vérifiant les inégalités (2.4a)-(2.4b) et en utilisant la même construction d'une solution que dans la preuve du lemme 2.10, nous construisons une solution pour $F^{z_{ab}}$ vérifiant les inégalités (2.11a)-(2.11b). $\square$

2.2.4 Formulation avec des variables séparant les décisions sur les abscisses et les ordonnées de chaque objet sélectionné

Le but est de séparer toutes les décisions entre les coordonnées en utilisant des variables de décision différentes pour les abscisses et pour les ordonnées. Boschetti et al. [12] proposent donc une autre formulation utilisant ces variables de décisions indépendantes.

FORMULATION 6: Formulation de Boschetti et al. $[F^{x_{ia}y_{ib}}]$

$$[F^{x_{ia}y_{ib}}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.12a)$$

$$\text{s. t.} \quad \sum_{a \in \mathcal{X}_i} \chi_{ia} = \sum_{b \in \mathcal{Y}_i} \gamma_{ib} = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.12b)$$

$$x_{ia} = \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}} \chi_{ia'}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}, \quad (2.12c)$$

$$y_{ib} = \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}} \gamma_{ib'}, \quad \forall i \in \mathcal{I}, b \in \mathcal{Y}, \quad (2.12d)$$

$$x_{ia} + x_{ja} - 1 \leq x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), a \in \mathcal{X}, \quad (2.12e)$$

$$x_{ij} \leq \sum_{a \in \mathcal{X}} x_{ia}, \quad x_{ij} \sum_{a \in \mathcal{X}} \leq x_{ja}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.12f)$$

$$y_{ib} + y_{jb} - 1 \leq y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), b \in \mathcal{Y}, \quad (2.12g)$$

$$y_{ij} \leq \sum_{b \in \mathcal{Y}} y_{ib}, \quad y_{ij} \leq \sum_{b \in \mathcal{Y}} y_{jb}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.12h)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.12i)$$

$$\sum_{i \in \mathcal{I}} \mathbf{h}_i x_{ia} \leq \mathbf{H}, \quad \forall a \in \mathcal{X}, \quad (2.12j)$$

$$\sum_{i \in \mathcal{I}} \mathbf{w}_i y_{ib} \leq \mathbf{W}, \quad \forall b \in \mathcal{Y}, \quad (2.12k)$$

$$\delta_i, x_{ij}, y_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.12l)$$

$$\chi_{ia}, \gamma_{ib} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}_i, b \in \mathcal{Y}_i, \quad (2.12m)$$

$$x_{ia}, y_{ib} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}, b \in \mathcal{Y}. \quad (2.12n)$$

La contrainte (2.12b) est identique à la contrainte (2.8b). La contrainte (2.12c) (resp. (2.12d)) permet de déterminer les abscisses (resp. ordonnées) occupées par un objet donné. En effet, d'après la contrainte (2.12c) (resp. (2.12d)), pour chaque abscisse a (resp. ordonnée b), si un objet i intersecte l'abscisse a (resp. l'ordonnée b), alors les variables x_{ia} (resp. y_{ib}) correspondantes prennent leurs valeurs à un; à zero dans le cas contraire. La contrainte (2.12e) (resp. (2.12g)) permet de déterminer les couples d'objets (i, j) partageant une abscisse (resp. ordonnée). La contrainte (2.12i) permet d'imposer que toute paire d'objets (i, j) ne peut partager à la fois une abscisse et une ordonnée. Donc, cette contrainte impose le non-chevauchement des objets. Les contraintes (2.12j)-(2.12k) modélisent le fait que chaque bande verticale et horizontale ne peut dépasser la hauteur et la largeur du

conteneur. En réalité, elles sont facultatives.

Remarque Nous remarquons que la formulation $F^{x_{ia}y_{ib}}$ est une relaxation de la formulation $F^{z_{ab}}$ en enlevant les variables de non-occupation d'une position z_{ab} et la contrainte (2.5d). D'après le corrolaire 2.13, nous montrons que la formulation $F^{x_{ia}y_{ib}}$ est valide. Enfin, de la même manière que la formulation $F^{z_{ab}}$, cette formulation peut être améliorée par les contraintes (2.11a)-(2.11b). Notons $F_{renf}^{x_{ia}y_{ib}}$ la formulation renforcée de la formulation $F^{x_{ia}y_{ib}}$. $\blacklozenge$

Corrolaire 2.21

La borne duale obtenue par la relaxation linéaire de la formulation $F_{renf}^{z_{ab}}$ est plus petite ou égale à celle obtenue par la relaxation linéaire de la formulation $F_{renf}^{x_{ia}y_{ib}}$.

Enfin, il est possible de relâcher la formulation $F^{x_{ia}y_{ib}}$ afin de réduire le nombre de contraintes et la dépendance des abscisses (ordonnées) entre elles aux niveaux des contraintes en vue d'une future décomposition. En sommant sur les contraintes (2.12c)-(2.12g) sur les abscisses et les ordonnées, nous obtenons la formulation suivante :

FORMULATION 7: Formulation relâchée de $F^{x_{ia}y_{ib}}$ [$F_{relax}^{x_{ia}y_{ib}}$]

$$[F_{relax}^{x_{ia}y_{ib}}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.13a)$$

$$\text{s. t.} \quad \sum_{a \in \mathcal{X}} x_{ia} = \mathbf{w}_i \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.13b)$$

$$\sum_{b \in \mathcal{Y}} y_{ib} = \mathbf{h}_i \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.13c)$$

$$\sum_{a \in \mathcal{X}} (x_{ia} + x_{ja} - 1) \leq \mathbf{W} x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.13d)$$

$$x_{ij} \leq \sum_{a \in \mathcal{X}} x_{ia}, \quad x_{ij} \leq \sum_{a \in \mathcal{X}} x_{ja}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.13e)$$

$$\sum_{b \in \mathcal{Y}} (y_{ib} + y_{jb} - 1) \leq \mathbf{H} y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.13f)$$

$$y_{ij} \leq \sum_{b \in \mathcal{Y}} y_{ib}, \quad y_{ij} \leq \sum_{b \in \mathcal{Y}} y_{jb}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.13g)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.13h)$$

$$\sum_{i \in \mathcal{I}} \mathbf{h}_i x_{ia} \leq \mathbf{H}, \quad \forall a \in \mathcal{X}, \quad (2.13i)$$

$$\sum_{i \in \mathcal{I}} \mathbf{w}_i y_{ib} \leq \mathbf{W}, \quad \forall b \in \mathcal{Y}, \quad (2.13j)$$

$$\delta_i, x_{ij}, y_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.13k)$$

$$x_{ia}, y_{ib} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}, b \in \mathcal{Y}. \quad (2.13l)$$

Les contraintes (2.13d)-(2.13f) peuvent être améliorées par les contraintes suivantes :

$$\sum_{a \in \mathcal{X}} (x_{ia} + x_{ja} - 1) \leq \min\{\mathbf{w}_i, \mathbf{w}_j\} x_{ij} \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.14a)$$

$$\sum_{b \in \mathcal{Y}} (y_{ib} + y_{jb} - 1) \leq \min\{\mathbf{h}_i, \mathbf{h}_j\} y_{ij} \quad \forall i, j \in \mathcal{I} (i < j). \quad (2.14b)$$

2.3 Modélisation basée sur un découpage par cliques maximales d'un placement

Une nouvelle approche consiste à utiliser l'approche de Fekete et Schepers (théorème 1.15) et la caractérisation des graphes d'intervalles par les cliques maximales (théorème 1.11) pour en déduire une nouvelle formulation. En effet, chaque clique est vue comme une bande remplissant le conteneur sur une largeur ou hauteur variable. La formulation pouvant être obtenue est compacte et souffre des mêmes inconvénients que les modèles utilisant une position relative des objets, à savoir l'utilisation de « grands M » et de contraintes de sac-à-dos (modélisant le non-dépassement du conteneur) rendant la relaxation linéaire faible. En revanche, il est possible de définir de nombreuses contraintes permettant l'élimination de symétries de placement.

2.3.1 Lexique des variables pour les formulations définissant un découpage par des cliques maximales d'un placement

D'après la caractérisation de Fekete et Schepers, nous savons qu'à chaque placement sur deux dimensions, nous pouvons associer une paire de graphes d'intervalles. Nous formulons donc ce problème en utilisant cette représentation. Or, d'après la caractérisation de Fulkerson et Gross, les cliques maximales d'un graphe d'intervalles peuvent être classées consécutivement. Pour utiliser cette caractérisation, nous définissons de nouvelles quantité calculable à partir des données.

Soient K^x une borne supérieure sur le nombre maximum de cliques maximales selon l'axe des abscisses et K^y une borne supérieure sur le nombre maximum de cliques maximales selon l'axe des ordonnées.

Par exemple, nous prenons $K^x = K^y = K$ où :

$$K = \max \left\{ \sum_{i \in \mathcal{I}} \delta_i : \sum_{i \in \mathcal{I}} \mathbf{w}_i \mathbf{h}_i \delta_i \leq \mathbf{W} \mathbf{H}, \quad \delta_i \in \{0, 1\} \quad \forall i \in \mathcal{I} \right\}.$$

En effet, K représente une borne supérieure sur le nombre maximum d'objets qui peuvent entrer dans le conteneur. Or, nous savons que le nombre maximum de cliques maximales pour un graphe d'intervalles est inférieur à son nombre de sommets. Donc, K représente une borne supérieure sur le nombre maximal de cliques maximales pour les différents graphes d'intervalles possibles.

Remarque Nous pouvons utiliser les fonctions duales réalisables (présentées à la section 1.4) pour améliorer la valeur de K . ♦

Soit $\mathcal{Q}_x = \{1, \dots, K^x\}$ l'ensemble des positions/classements possibles des cliques maximales selon l'axe des x . De même, $\mathcal{Q}_y = \{1, \dots, K^y\}$ est l'ensemble des positions/classements possibles des cliques maximales selon l'axe des y .

Remarque Comme nous le verrons par la suite, une clique maximale peut être vue comme une bande. En effet, chaque clique maximale pour le graphe d'intervalles associée à l'axe

des abscisses forme une bande verticale dans le placement associé. De même, chaque clique maximale pour le graphe d'intervalles associé à l'axe des ordonnées forme une bande horizontale dans le placement associé. Nous parlons donc indifféremment de cliques ou de bandes. ♦

Ainsi, nous définissons alors des variables indiquant si un objet est présent dans une clique :

$$\left\{ \begin{array}{l} x_i^k = \begin{cases} 1 & \text{si l'objet } i \text{ appartient à la clique en position } k \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \\ y_i^l = \begin{cases} 1 & \text{si l'objet } i \text{ appartient à la clique en position } l \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y. \end{array} \right.$$

Nous associons aussi à chaque clique une variable définissant sa largeur :

$$\left\{ \begin{array}{l} w^k \in [0, \mathbf{W}] \text{ est la largeur de bande verticale en position } k, \quad \forall k \in \mathcal{Q}_x, \\ h^l \in [0, \mathbf{H}] \text{ est la largeur de bande horizontale en position } l, \quad \forall l \in \mathcal{Q}_y. \end{array} \right.$$

D'autres variables sont aussi introduites comme la fraction d'un objet dans une clique :

$$\left\{ \begin{array}{l} \phi_i^k \in [0, 1] \text{ est la fraction de l'objet } i \text{ dans la bande verticale en position } k, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \\ \varphi_i^l \in [0, 1] \text{ est la fraction de l'objet } i \text{ dans la bande horizontale en position } l, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y. \end{array} \right.$$

De plus, nous rajoutons des variables indiquant la première clique où commence un objet. Par exemple, pour les cliques verticales :

$$\left\{ \begin{array}{l} d_i^k = \begin{cases} 1 & \text{si l'objet } i \text{ débute à partir de la clique verticale} \\ & \text{en position } k \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \\ f_i^k = \begin{cases} 1 & \text{si l'objet } i \text{ fini à partir de la clique verticale} \\ & \text{en position } k \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x. \end{array} \right.$$

De même pour les cliques horizontales :

$$\left\{ \begin{array}{l} e_i^l = \begin{cases} 1 & \text{si l'objet } i \text{ débute à partir de la clique horizontale} \\ & \text{en position } l \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \\ g_i^l = \begin{cases} 1 & \text{si l'objet } i \text{ fini à partir de la clique horizontale} \\ & \text{en position } l \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y. \end{array} \right.$$

Enfin, nous avons aussi besoin d'une variable permettant de savoir si une clique est vide ou non :

$$\left\{ \begin{array}{l} x_0^k = \begin{cases} 1 & \text{si la clique verticale en position } k \text{ est vide} \\ 0 & \text{sinon} \end{cases}, \\ y_0^l = \begin{cases} 1 & \text{si la clique horizontale en position } l \text{ est vide} \\ 0 & \text{sinon} \end{cases}, \end{array} \right. \quad \begin{array}{l} \forall k \in \mathcal{Q}_x, \\ \forall l \in \mathcal{Q}_y. \end{array}$$

2.3.2 Formulation définissant un découpage par cliques d'un placement

La formulation compacte quadratique suivante est déduite des théorèmes précédemment cités :

FORMULATION 8: Nouvelle formulation compacte quadratique $[\text{FQ}^{w^k h^l}]$

$$[\text{FQ}^{w^k h^l}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.15a)$$

$$\text{s. t.} \quad \sum_{k \in \mathcal{Q}_x} w^k x_i^k = \mathbf{w}_i \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.15b)$$

$$\sum_{l \in \mathcal{Q}_y} h^l y_i^l = \mathbf{h}_i \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.15c)$$

$$\sum_{k \in \mathcal{Q}_x} w^k \leq \mathbf{W}, \quad (2.15d)$$

$$\sum_{l \in \mathcal{Q}_y} h^l \leq \mathbf{H}, \quad (2.15e)$$

$$x_i^k x_i^{k''} \leq x_i^{k'}, \quad \forall i \in \mathcal{I}, k, k', k'' \in \mathcal{Q}_x (k < k' < k''), \quad (2.15f)$$

$$y_i^l y_i^{l''} \leq y_i^{l'}, \quad \forall i \in \mathcal{I}, l, l', l'' \in \mathcal{Q}_y (l < l' < l''), \quad (2.15g)$$

$$x_i^k x_j^k \leq x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), k \in \mathcal{Q}_x, \quad (2.15h)$$

$$y_i^l y_j^l \leq y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), l \in \mathcal{Q}_y, \quad (2.15i)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.15j)$$

$$x_{ij}, y_{ij}, \delta_i \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.15k)$$

$$x_i^k \in \{0, 1\}, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.15l)$$

$$y_i^l \in \{0, 1\}, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.15m)$$

$$0 \leq w^k \leq \mathbf{W}, \quad \forall k \in \mathcal{Q}_x, \quad (2.15n)$$

$$0 \leq h^l \leq \mathbf{H}, \quad \forall l \in \mathcal{Q}_y. \quad (2.15o)$$

Les contraintes (2.15b)-(2.15c) permettent de vérifier que, pour chaque objet sélectionné, les bandes contenant celui-ci le couvrent selon sa largeur et sa hauteur. Les contraintes (2.15d)-(2.15e) imposent que le placement n'excède pas la taille de la boîte en vérifiant que la somme des largeurs des bandes est inférieure à la taille du conteneur. Les contraintes (2.15f)-(2.15g) imposent que les objets apparaissent dans des bandes classées consécutivement selon la règle de classement des cliques (théorème 1.11). Enfin, les contraintes (2.15h)-(2.15j) imposent que les objets ne se chevauchent pas.

Lemme 2.22

Toute solution entière réalisable de la formulation $FQ^{w^k h^l}$ peut être transformée en une solution réalisable de même valeur selon la définition 1.1.

Preuve. Soit une solution entière de la formulation $FQ^{w^k h^l}$.

Pour tout objet $i \in \mathcal{I}' = \{i \in \mathcal{I} : \delta_i = 1\}$, soit $k_i \in \mathcal{Q}_x$ (resp. $l_i \in \mathcal{Q}_y$) le plus petit indice de position tel que $x_i^{k_i} = 0$ et $x_i^{k_i+1} = 1$ (resp. $y_i^{l_i} = 0$ et $y_i^{l_i+1} = 1$). Posons alors, $\forall i \in \mathcal{I}'$,

$$x_i = \sum_{k \in \mathcal{Q}_x : k \leq k_i} w^k \quad \text{et} \quad y_i = \sum_{l \in \mathcal{Q}_y : l \leq l_i} h^l.$$

Montrons alors que cette solution est valide.

En effet, la condition (1.1a) est vérifiée car, d'après les contraintes (2.15b) et (2.15d), $\forall i \in \mathcal{I}'$, nous avons : $x_i + \mathbf{w}_i = \sum_{k \in \mathcal{Q}_x : k \leq k_i^1} w^k + \sum_{k \in \mathcal{Q}_x} w^k x_i^k \leq \sum_{k \in \mathcal{Q}_x} w^k \leq \mathbf{W}$. De même,

la condition (1.1a) est vérifiée pour les variables y en utilisant les contraintes (2.15c) et (2.15e).

De plus, d'après les contraintes (2.15h)-(2.15j), nous rencontrons quatre cas :

- si $x_{ij} = 0$ et $x_i < x_j$ alors $x_i + \mathbf{w}_i \leq \sum_{\substack{k \in \mathcal{Q}_x \\ k \leq k_i}} w^k + \sum_{k \in \mathcal{Q}_x} w^k x_i^k \leq \sum_{\substack{k \in \mathcal{Q}_x \\ k \leq k_j}} w^k$, d'après (2.15f).
- si $x_{ij} = 0$ et $x_j < x_i$ alors $x_j + \mathbf{w}_j \leq \sum_{\substack{k \in \mathcal{Q}_x \\ k \leq k_j}} w^k + \sum_{k \in \mathcal{Q}_x} w^k x_j^k \leq \sum_{\substack{k \in \mathcal{Q}_x \\ k \leq k_i}} w^k$, d'après (2.15f).
- si $y_{ij} = 0$ et $y_i < y_j$ alors $y_i + \mathbf{h}_i \leq \sum_{\substack{l \in \mathcal{Q}_y \\ l \leq l_i}} h^l + \sum_{l \in \mathcal{Q}_y} h^l y_i^l \leq \sum_{\substack{l \in \mathcal{Q}_y \\ l \leq l_j}} h^l$, d'après (2.15g).
- si $y_{ij} = 0$ et $y_j < y_i$ alors $y_j + \mathbf{h}_j \leq \sum_{\substack{l \in \mathcal{Q}_y \\ l \leq l_j}} h^l + \sum_{l \in \mathcal{Q}_y} h^l y_j^l \leq \sum_{\substack{l \in \mathcal{Q}_y \\ l \leq l_i}} h^l$, d'après (2.15g).

Donc, la condition (1.1b) est valide. $\square$

Lemme 2.23

Toute solution réalisable selon la définition 1.1 peut être transformée en une solution entière réalisable de même valeur pour la formulation $FQ^{w^k h^l}$.

Preuve. Soit une solution vérifiant les conditions (1.1a)-(1.1b). D'après le lemme 1.3, il existe une solution avec la condition « calée en bas à gauche ».

Pour tout objet $i \in \mathcal{I}'$, notons (x_i, y_i) la coordonnée associée à l'objet i .

Soient, $X = \{x \in \mathbb{R} : \exists i \in \mathcal{I}', x = x_i + \mathbf{w}_i\}$ et $Y = \{y \in \mathbb{R} : \exists i \in \mathcal{I}', y = y_i + \mathbf{h}_i\}$ l'ensemble des abscisses et des ordonnées tel qu'il existe au moins un objet terminant en cette coordonnée. Posons $X = \{X_1, \dots, X_{m^x}\}$ avec $X_1 < \dots < X_{m^x}$ et $m^x = |X|$; $Y = \{Y_1, \dots, Y_{m^y}\}$ avec $Y_1 < \dots < Y_{m^y}$ et $m^y = |Y|$. D'après la condition (1.1a), $X_{m^x} \leq \mathbf{W}$ et $Y_{m^y} \leq \mathbf{H}$. Prenons $X_0 = 0 < X_1$ et $Y_0 = 0 < Y_1$. Pour tout $i \in \mathcal{I}'$, posons $\delta_i = 1$ et pour tout $i \in \mathcal{I} \setminus \mathcal{I}'$, posons $\delta_i = 0$. Par définition de X et Y , nous avons $m_x \leq |\mathcal{I}'| \leq K_x$ et $m_y \leq |\mathcal{I}'| \leq K_y$. De plus, comme la solution est « calée en bas à gauche », nous avons qu'il existe $0 \leq k \leq m_x - 1$ tel que $x_i = X_k$ et qu'il existe $0 \leq l \leq m_y - 1$ tel que $y_i = Y_l$, pour chaque item $i \in \mathcal{I}'$.

Alors, pour tout $i \in \mathcal{I} \setminus \mathcal{I}'$, posons les variables associées à zéro et pour tout $i \in \mathcal{I}'$, posons :

- $x_i^k = 1$ si $x_i \leq X_{k-1} < x_i + \mathbf{w}_i$; 0 sinon, pour tout $1 \leq k \leq m^x$,
- $y_i^l = 1$ si $y_i \leq Y_{l-1} < y_i + \mathbf{h}_i$; 0 sinon, pour tout $1 \leq l \leq m^y$.

Enfin, posons, $w^k = X^k - X^{k-1}$, pour $1 \leq k \leq m^x$, et $h^l = Y^l - Y^{l-1}$, pour $1 \leq l \leq m^y$. Montrons que la solution ainsi obtenue est valide.

Les contraintes (2.15b)-(2.15c) sont vérifiées par la solution. En effet, pour chaque objet $i \in \mathcal{I}'$, grâce à la condition « calée en bas à gauche », nous avons :

$$\mathbf{w}_i = \sum_{\substack{k \in \mathcal{Q}_x: \\ x_i \leq X_{k-1} < x_i + \mathbf{w}_i}} (X_k - X_{k-1}) = \sum_{k \in \mathcal{Q}_x} w^k x_i^k.$$

Nous démontrons de manière analogue la véracité de la contrainte (2.15c).

Les contraintes (2.15d)-(2.15e) sont aussi vérifiées car

$$\sum_{k \in \mathcal{Q}_x} w^k = X_{m^x} - X_0 \leq \mathbf{W} \text{ et } \sum_{l \in \mathcal{Q}_y} h^l = Y_{m^y} - Y_0 \leq \mathbf{H}.$$

Pour $i \in \mathcal{I} \setminus \mathcal{I}'$, les contraintes (2.15f)-(2.15g) sont vérifiées. Pour $i \in \mathcal{I}'$, supposons que $x_i^k = 1$ et $x_i^{k''} = 1$ (sinon les contraintes (2.15f)-(2.15g) sont trivialement vérifiées) avec $1 \leq k < k'' \leq m^x$. D'où, $x_i \leq X_{k-1} \leq X_{k'-1} \leq X_{k''-1} < x_i + \mathbf{w}_i$ pour tout $k < k' < k''$. Donc, $x_i^{k'} = 1$ pour tout $k < k' < k''$. De même pour les variables y . Donc, les contraintes (2.15f)-(2.15g) sont vérifiées.

Enfin, d'après la condition (1.1b), nous rencontrons quatre cas possibles, pour toute paire d'objets $i, j \in \mathcal{I}$ avec $i < j$:

- Soit $x_i + \mathbf{w}_i \leq x_j$, alors $X_{k-1} < x_i + \mathbf{w}_i \leq x_j \leq X_{k'-1}$ pour tout $1 \leq k < k' \leq m^x$ tel que $x_i^k = 1$ et $x_j^{k'} = 1$. Ainsi, $x_i^k x_j^k = 0$, pour tout $k \in \mathcal{Q}_x$. x_{ij} prend la valeur zéro et alors les contraintes (2.15h)-(2.15j) sont vérifiées.
- Soit $x_j + \mathbf{w}_j \leq x_i$, alors $X_{k-1} < x_j + \mathbf{w}_j \leq x_i \leq X_{k'-1}$ pour tout $1 \leq k < k' \leq m^x$ tel que $x_j^k = 1$ et $x_i^{k'} = 1$. Ainsi, $x_i^k x_j^k = 0$, pour tout $k \in \mathcal{Q}_x$. x_{ij} prend la valeur zéro et alors les contraintes (2.15h)-(2.15j) sont vérifiées.
- Soit $y_i + \mathbf{h}_i \leq y_j$ ou $y_j + \mathbf{h}_j \leq y_i$, alors, de manière similaire aux cas précédents, la variable y_{ij} prend la valeur zéro et donc les contraintes (2.15h)-(2.15j) sont vérifiées.

Nous avons donc réussi à contruire une solution valide pour la formulation $\text{FQ}^{w^k h^l}$. $\square$

Corrolaire 2.24

La formulation $\text{FQ}^{w^k h^l}$ est valide.

Remarque Nous remarquons qu'en réalité les contraintes (2.15f)-(2.15g) ne sont pas toutes nécessaires. En effet, il suffit juste de vérifier ces contraintes que sur des triplets de cliques $k, k+1, k' \in \mathcal{Q}_x$ tel que $k+1 < k'$ pour la contrainte (2.15f) et $l, l+1, l' \in \mathcal{Q}_y$ tel que $l+1 < l'$ pour la contrainte (2.15g). Ceci réduit ainsi grandement le nombre de contraintes. $\blacklozenge$

Il est possible de linéariser la formulation $\text{FQ}^{w^k h^l}$ en procédant de la façon suivante. Pour les contraintes quadratiques (2.15b)-(2.15c), en remarquant que $w^k x_i^k = \max_{j \in \mathcal{I}} \mathbf{w}_j \phi_j^k$ pour $i \in \mathcal{I}$ et $k \in \mathcal{Q}_x$ et $h^l y_i^l = \max_{j \in \mathcal{I}} \mathbf{h}_j \varphi_j^l$ pour $i \in \mathcal{I}$ et $l \in \mathcal{Q}_y$, nous linéarisons de la façon suivante :

$$\sum_{k \in \mathcal{Q}_x} \phi_i^k = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.16a)$$

$$\sum_{l \in \mathcal{Q}_y} \varphi_i^l = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.16b)$$

$$\phi_i^k \leq x_i^k, \quad \mathbf{w}_i \phi_i^k \leq w^k, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.16c)$$

$$\varphi_i^l \leq y_i^l, \quad \mathbf{h}_i \varphi_i^l \leq h^l, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y. \quad (2.16d)$$

Les contraintes (2.16a)-(2.16b) remplacent les contraintes de respect de la largeur des objets (2.15b)-(2.15c). Les contraintes (2.16c)-(2.16d) imposent que seuls les objets contenus dans une bande donnée peuvent avoir une fraction positive strictement. Les contraintes (2.16c)-(2.16d) permettent aussi d'imposer une largeur de bande au moins aussi grande que la plus grande fraction d'objet contenue dans la bande.

Lemme 2.25

Les contraintes (2.16a)-(2.16d) sont une linéarisation valide des contraintes (2.15b)-(2.15c).

Preuve. Toute solution entière valide de la formulation $\text{FQ}^{w^k h^l}$ vérifient les contraintes (2.16a)-(2.16d) en posant $\mathbf{w}_i \phi_i^k = w^k x_i^k$ pour $i \in \mathcal{I}$ et $k \in \mathcal{Q}_x$ et $\mathbf{h}_i \varphi_i^l = h^l y_i^l$ pour $i \in \mathcal{I}$ et $l \in \mathcal{Q}_y$.

Montrons alors que la formulation obtenue en remplaçant les contraintes (2.15b)-(2.15c) par (2.16a)-(2.16d) est encore valide. Soit une solution entière d'une telle formulation. Il suffit d'adapter les preuves des lemmes 2.22 et 2.23 à la formulation linéarisée.

Pour la preuve du premier lemme, en remarquant que, $\forall i \in \mathcal{I}'$,

$$x_i + \mathbf{w}_i = \sum_{k \in \mathcal{Q}_x : k \leq k_i^1} w^k + \sum_{k \in \mathcal{Q}_x} w_i \phi_i^k \leq \sum_{k \in \mathcal{Q}_x} w^k \leq \mathbf{W},$$

nous démontrons que le lemme reste vrai pour la version linéarisée de $\text{FQ}^{w^k h^l}$.

Pour le deuxième lemme, posons :

- $\phi_i^k = (w^k x_i^k) / \mathbf{w}_i$, pour $i \in \mathcal{I}, k \in \mathcal{Q}_x$; 0 sinon,
- $\varphi_i^l = (h^l y_i^l) / \mathbf{h}_i$, pour $i \in \mathcal{I}, l \in \mathcal{Q}_y$; 0 sinon.

Les contraintes (2.16a)-(2.16d) sont vérifiées par cette solution. $\square$

La linéarisation des contraintes (2.15f)-(2.15g) peut se faire en les remplaçant par :

$$x_i^k - x_i^{k-1} \leq d_i^k, \quad x_i^k - x_i^{k+1} \leq f_i^k, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.17a)$$

$$y_i^l - y_i^{l-1} \leq e_i^l, \quad y_i^l - y_i^{l+1} \leq g_i^l, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.17b)$$

$$\sum_{k \in \mathcal{Q}_x} d_i^k = \delta_i, \quad \sum_{k \in \mathcal{Q}_x} f_i^k = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.17c)$$

$$\sum_{l \in \mathcal{Q}_y} e_i^l = \delta_i, \quad \sum_{l \in \mathcal{Q}_y} g_i^l = \delta_i, \quad \forall i \in \mathcal{I}. \quad (2.17d)$$

Pour chaque dimension, les contraintes (2.17a)-(2.17b) permettent l'initialisation des variables indiquant, pour un objet donné, la première clique et la dernière clique contenant celui-ci. Pour chaque dimension, les contraintes (2.17c)-(2.17d) imposent que, pour chaque objet sélectionné, il n'existe qu'une première clique et qu'une dernière clique.

Lemme 2.26

Les contraintes (2.17a)-(2.17d) sont une linéarisation valide des contraintes (2.15f)-(2.15g).

Preuve. Toute solution valide de la formulation $\text{FQ}^{w^k h^l}$ vérifient les contraintes (2.17a)-(2.17d) en posant :

- $d_i^k = 1$ si $x_i^k = 1$ et $x_i^{k-1} = 0$; 0 sinon. Et $f_i^k = 1$ si $x_i^k = 1$ et $x_i^{k+1} = 0$; 0 sinon pour tout $i \in \mathcal{I}, k \in \mathcal{Q}_x$.

– $e_i^l = 1$ si $y_i^l = 1$ et $y_i^{l-1} = 0$; 0 sinon. Et $g_i^l = 1$ si $y_i^l = 1$ et $y_i^{l+1} = 0$; 0 sinon pour tout $i \in \mathcal{I}, l \in \mathcal{Q}_y$.

En effet, les contraintes (2.17a)-(2.17b) sont trivialement vérifiées. De plus, nous déduisons que $\sum_{k \in \mathcal{Q}_x} d_i^k \geq \delta_i$ et $\sum_{k \in \mathcal{Q}_x} f_i^k \geq \delta_i$ pour tout $i \in \mathcal{I}$. Enfin, nous montrons par l'absurde que $\sum_{k \in \mathcal{Q}_x} d_i^k \leq \delta_i$ et $\sum_{k \in \mathcal{Q}_x} f_i^k \leq \delta_i$. Par exemple, supposons qu'il existe un objet $i \in \mathcal{I}$ tel que $\sum_{k \in \mathcal{Q}_x} d_i^k > \delta_i$. Alors, il existe au moins deux cliques $k', k'' \in \mathcal{Q}_x$ tel que $d_i^{k'} = d_i^{k''} = 1$ et $k' < k''$. Or, d'après la contrainte (2.17c), nous en déduisons que $x_i^{k'} = x_i^{k''} = 1$ et $x_i^{k''-1} = 0$. Ce résultat est incompatible avec la contrainte (2.15f). Donc, la contrainte (2.17c) est vérifiée. Nous démontrons de manière similaire que la contrainte (2.17d) est vérifiée.

Il nous reste donc à vérifier que la formulation obtenue en remplaçant les contraintes (2.15f)-(2.15g) par (2.17a)-(2.17d) est valide. Soit une solution entière d'une telle formulation. Par l'absurde, supposons qu'il existe une contrainte (2.15f) violée. Alors, il existe $k, k', k'' \in \mathcal{Q}_x$ les plus petits indices tel que $x_i^{k'} < x_i^k x_i^{k''}$ avec $k < k' < k''$. Nous en déduisons alors que $x_i^{k-1} = x_i^{k'} = x_i^{k''-1} = 0$ et $x_i^k = x_i^{k''} = 1$. D'où, $d_i^k = d_i^{k''} = 1$ ce qui contredit la contrainte (2.17c). Donc, la contrainte (2.15f) est vérifiée par la solution. Nous démontrons de manière similaire que la contrainte (2.15g) est vérifiée. $\square$

La linéarisation des contraintes (2.15h)-(2.15i) est classique :

$$x_i^k + x_j^k - 1 \leq x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), k \in \mathcal{Q}_x, \quad (2.18a)$$

$$y_i^l + y_j^l - 1 \leq y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), l \in \mathcal{Q}_y. \quad (2.18b)$$

Nous en déduisons alors une nouvelle formulation compacte linéaire notée $F^{w^k h^l}$:

FORMULATION 9: Nouvelle formulation compacte $[F^{w^k h^l}]$

$$[F^{w^k h^l}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.19a)$$

$$\text{s. t.} \quad \sum_{k \in \mathcal{Q}_x} \phi_i^k = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.19b)$$

$$\sum_{l \in \mathcal{Q}_y} \varphi_i^l = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.19c)$$

$$\phi_i^k \leq x_i^k, \quad \mathbf{w}_i \phi_i^k \leq w^k, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.19d)$$

$$\varphi_i^l \leq y_i^l, \quad \mathbf{h}_i \varphi_i^l \leq h^l, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.19e)$$

$$\sum_{k \in \mathcal{Q}_x} w^k \leq \mathbf{W}, \quad \sum_{l \in \mathcal{Q}_y} h^l \leq \mathbf{H}, \quad (2.19f)$$

$$x_i^k - x_i^{k-1} \leq d_i^k, \quad x_i^k - x_i^{k+1} \leq f_i^k, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.19g)$$

$$y_i^l - y_i^{l-1} \leq e_i^l, \quad y_i^l - y_i^{l+1} \leq g_i^l, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.19h)$$

$$\sum_{k \in \mathcal{Q}_x} d_i^k = \delta_i, \quad \sum_{k \in \mathcal{Q}_x} f_i^k = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.19i)$$

$$\sum_{l \in \mathcal{Q}_y} e_i^l = \delta_i, \quad \sum_{l \in \mathcal{Q}_y} g_i^l = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.19j)$$

$$x_i^k + x_j^k - 1 \leq x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), k \in \mathcal{Q}_x, \quad (2.19k)$$

$$y_i^l + y_j^l - 1 \leq y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), l \in \mathcal{Q}_y, \quad (2.19l)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.19m)$$

$$x_{ij}, y_{ij}, \delta_i \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.19n)$$

$$x_i^k \in \{0, 1\}, \quad 0 \leq w^k \leq \mathbf{W}, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.19o)$$

$$y_i^l \in \{0, 1\}, \quad 0 \leq h^l \leq \mathbf{H}, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y. \quad (2.19p)$$

En plus de ces contraintes de linéarisation, les contraintes suivantes sont vérifiées et peuvent être rajoutées aux formulations. La contrainte (2.20a) permet de vérifier que chaque bande verticale ne peut dépasser la taille du conteneur selon sa hauteur. Similairement pour la contrainte (2.20b) :

$$\sum_{i \in \mathcal{I}} \mathbf{h}_i x_i^k \leq \mathbf{H}, \quad \forall k \in \mathcal{Q}_x, \quad (2.20a)$$

$$\sum_{i \in \mathcal{I}} \mathbf{w}_i y_i^l \leq \mathbf{W}, \quad \forall l \in \mathcal{Q}_y. \quad (2.20b)$$

Lemme 2.27

Les contraintes (2.20a)-(2.20b) sont valides.

Preuve. Soit une solution entière de $F^{w^k h^l}$. Démontrons que la contrainte (2.20a) est vérifiée. Soit $k \in \mathcal{Q}_x$. Posons $\mathcal{I}_k = \{i \in \mathcal{I} : x_i^k = 1\}$. Alors, $\sum_{i \in \mathcal{I}} \mathbf{h}_i x_i^k = \sum_{i \in \mathcal{I}_k} \mathbf{h}_i$. Or, en utilisant les contraintes (2.19k)-(2.19m), nous déduisons que $x_{ij} = 1$ et $y_{ij} = 0$, $\forall i, j \in \mathcal{I}_k$ avec $i < j$. Alors, nous en déduisons que $\sum_{i \in \mathcal{I}_k} y_i^l \leq 1$, $\forall l \in \mathcal{Q}_y$.

De plus, d'après les contraintes (2.19c) et (2.19e), nous avons, $\sum_{l \in \mathcal{Q}_y} \mathbf{h}_i \varphi_i^l = \mathbf{h}_i$, pour tout $i \in \mathcal{I}_k$ et $\sum_{i \in \mathcal{I}_k} \mathbf{h}_i \varphi_i^l \leq h^l$, pour tout $l \in \mathcal{Q}_y$.

Donc, $\sum_{i \in \mathcal{I}_k} \mathbf{h}_i \leq \sum_{i \in \mathcal{I}_k} \sum_{l \in \mathcal{Q}_y} \mathbf{h}_i \varphi_i^l \leq \sum_{l \in \mathcal{Q}_y} h^l \leq \mathbf{H}$, d'après la contrainte (2.19f).

De manière similaire, nous démontrons que la contrainte (2.20b) est vérifiée. $\square$

Les contraintes suivantes forcent le remplissage en priorité des premières cliques :

$$\sum_{i \in \mathcal{I}} x_i^k + x_0^k \geq 1, \quad \forall k \in \mathcal{Q}_x, \quad (2.21a)$$

$$\sum_{i \in \mathcal{I}} y_i^l + y_0^l \geq 1, \quad \forall l \in \mathcal{Q}_y, \quad (2.21b)$$

$$x_i^k + x_0^k \leq 1, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.21c)$$

$$y_i^l + y_0^l \leq 1, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.21d)$$

$$x_0^k \leq x_0^{k+1}, \quad \forall k \in \mathcal{Q}_x, \quad (2.21e)$$

$$y_0^l \leq y_0^{l+1}, \quad \forall l \in \mathcal{Q}_y, \quad (2.21f)$$

$$K_x - \sum_{k \in \mathcal{Q}_x} x_0^k \leq \sum_{i \in \mathcal{I}} \delta_i, \quad (2.21g)$$

$$K_y - \sum_{l \in \mathcal{Q}_y} y_0^l \leq \sum_{i \in \mathcal{I}} \delta_i. \quad (2.21h)$$

Les contraintes (2.21a)-(2.21d) permettent d'initialiser les variables déterminant si une clique ne contient pas d'objets. Les contraintes (2.21e)-(2.21f) imposent que les premières cliques doivent être remplies avant les dernières cliques. Enfin, les contraintes (2.21g)-(2.21h) imposent qu'au plus $\sum_{i \in \mathcal{I}} \delta_i$ bandes verticales et horizontales sont sélectionnées.

Lemme 2.28

Les contraintes (2.21a)-(2.21f) sont valides.

Preuve. Les contraintes (2.21a)-(2.21d) sont trivialement valides.

Montrons que les contraintes (2.21e)-(2.21f) sont vérifiées. Supposons qu'il existe une solution valide de $F^{w^k h^l}$, notée $(\tilde{w}, \tilde{h}, \tilde{x}, \tilde{y}, \tilde{\phi}, \tilde{\varphi}, \tilde{d}, \tilde{e}, \tilde{f}, \tilde{g})$, tel que la contrainte (2.21e) est violée. Soit $k^* = \operatorname{argmin}_{k \in \mathcal{Q}_x} \{\tilde{x}_0^k > \tilde{x}_0^{k+1}\}$. Donc, $\tilde{x}_0^{k^*} = 1$ et $\tilde{x}_0^{k^*+1} = 0$. D'après les contraintes (2.21a)

et (2.21c), pour tout $i \in \mathcal{I}$, $\tilde{x}_i^{k^*} = 0$ et il existe $i \in \mathcal{I}$ tel que $\tilde{x}_i^{k^*+1} = 1$. Prenons la solution suivante, pour tout $k \in \mathcal{Q}_x$:

- $w^{k-1} = \tilde{w}^k$, $x_i^{k-1} = \tilde{x}_i^k$, $\phi_i^{k-1} = \tilde{\phi}^k$, $d_i^{k-1} = \tilde{d}_i^k$ et $f_i^{k-1} = \tilde{f}_i^k$, $\forall i \in \mathcal{I}$ si $k > k^*$,
- $w^k = \tilde{w}^k$, $x_i^k = \tilde{x}_i^k$, $\phi_i^k = \tilde{\phi}^k$, $d_i^k = \tilde{d}_i^k$ et $f_i^k = \tilde{f}_i^k$, $\forall i \in \mathcal{I}$, si $k < k^*$,
- $w^{K^x} = \tilde{w}^{k^*}$, $x_i^{K^x} = \tilde{x}_i^{k^*}$, $\phi_i^{K^x} = \tilde{\phi}^{k^*}$, $d_i^{K^x} = \tilde{d}_i^{k^*}$ et $f_i^{K^x} = \tilde{f}_i^{k^*}$, $\forall i \in \mathcal{I}$, si $k = k^*$.

Les autres variables prennent les mêmes valeurs.

Alors, les contraintes (2.19b)-(2.19f) et (2.19k)-(2.19m) restent forcément valides. Les contraintes (2.19g)-(2.19j) restent vérifiées car $\tilde{x}_i^{k^*} = \tilde{d}_i^{k^*} = \tilde{f}_i^{k^*} = 0$, $\forall i \in \mathcal{I}$. De plus, nous remarquons que les contraintes supplémentaires (2.20a)-(2.20b) restent encore valides. Enfin, nous avons $\tilde{x}_0^{k^*+1} = 0 = x_0^{k^*} \leq x_0^{k^*+1}$, d'où la contrainte (2.21e) est vérifiée pour k^* . En appliquant itérativement sur les contraintes violées, nous construisons alors une solution valide vérifiant les contraintes (2.21e).

De manière similaire, nous démontrons la validité de la contrainte (2.21f). $\square$

Lemme 2.29

Les contraintes (2.21g)-(2.21h) sont valides.

Preuve. Soit une solution entière valide de $F^{w^k h^l}$. D'après les lemmes 2.22, 2.23 et 2.25 et leurs preuves respectives, nous pouvons reconstruire pour chaque placement réalisable une solution réalisable pour la formulation $F^{w^k h^l}$. En reprenant la solution reconstruite à partir de la preuve du lemme 2.23, nous montrons facilement que les nouvelles inégalités valides (2.20a)-(2.20b) et (2.21a)-(2.21f) sont vérifiées en posant simplement $x_0^k = 1$ si $k > m^x$; 0 sinon. Et $y_0^l = 1$ si $l > m^y$; 0 sinon.

Pour cette solution, nous déduisons que $m^x + \sum_{k \in \mathcal{Q}_x} x_0^k = K_x$ et $m^y + \sum_{l \in \mathcal{Q}_y} y_0^l = K_y$. De plus, par construction, nous savons que $m^x \leq \sum_{i \in \mathcal{I}} \delta_i$ et $m^y \leq \sum_{i \in \mathcal{I}} \delta_i$.

Donc, les contraintes sont (2.21g)-(2.21h) vérifiées par la solution reconstruite. $\square$

De plus, à partir de toutes ces contraintes définies précédemment, nous en déduisons de nouvelles contraintes valides. Les contraintes suivantes permettent de remplacer les contraintes (2.21a)-(2.21b). En effet, en plus de fixer les valeurs des variables x_0^k et y_0^l , ces contraintes garantissent que la solution ne contient que des cliques maximales mais aussi que les objets sont « calés en bas à gauche » :

$$\sum_{j \in \mathcal{I}} d_j^k + x_0^k \geq 1, \quad d_i^k + x_0^k \leq 1, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.22a)$$

$$\sum_{j \in \mathcal{I}} f_j^k + x_0^k \geq 1, \quad f_i^k + x_0^k \leq 1, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.22b)$$

$$\sum_{j \in \mathcal{I}} e_j^l + y_0^l \geq 1, \quad e_i^l + y_0^l \leq 1, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.22c)$$

$$\sum_{j \in \mathcal{I}} g_j^l + y_0^l \geq 1, \quad g_i^l + y_0^l \leq 1, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y. \quad (2.22d)$$

Lemme 2.30

Les contraintes (2.22a)-(2.22d) sont valides.

Preuve. Soit une solution entière réalisable de $F^{w^k h^l}$ vérifiant les inégalités valides précédentes. D'après les différentes preuves des lemmes précédents, nous savons que cette solution peut être explicitée en un placement valide pour la définition 1.1. Il reste donc à montrer que toute solution réalisable selon la définition 1.1 peut être transposée en une

solution de $F^{w^k h^l}$ vérifiant les nouvelles contraintes et les contraintes (2.22a)-(2.22d). Or, d'après le lemme 1.3, pour chaque placement valide, nous pouvons contruire un placement avec la condition « calée en bas à gauche ». Soit $(x_i, y_i)_{i \in \mathcal{I}'}$ un tel placement. En utilisant une preuve similaire au lemme 2.23 pour ce placement mais en procédant comme suit nous démontrons la validité des contraintes :

Soient, $X = \{x \in \mathbb{R} : \exists i \in \mathcal{I}', x = x_i\}$ et $Y = \{y \in \mathbb{R} : \exists i \in \mathcal{I}', y = y_i\}$ l'ensemble des abscisses et des ordonnées tel qu'il existe au moins un objet commençant en cette coordonnée. Posons $X = \{X_1, \dots, X_{m^x}\}$ avec $X_1 < \dots < X_{m^x}$ et $m^x = |X|$; $Y = \{Y_1, \dots, Y_{m^y}\}$ avec $Y_1 < \dots < Y_{m^y}$ et $m^y = |Y|$. Prenons $X_{m^x+1} = \max\{x_i + \mathbf{w}_i : i \in \mathcal{I}'\}$ et $Y_{m^y+1} = \max\{y_i + \mathbf{h}_i : i \in \mathcal{I}'\}$. Pour chaque $i \in \mathcal{I}'$, soit k_i le plus grand indice $1 \leq k \leq m^x$ tel que $x_i \leq X_{k_i} < x_i + \mathbf{w}_i$ et l_i le plus grand indice $1 \leq l \leq m^y$ tel que $y_i \leq Y_{l_i} < y_i + \mathbf{h}_i$. Alors, pour tout $i \in \mathcal{I} \setminus \mathcal{I}'$, posons les variables associées à zéro et pour tout $i \in \mathcal{I}'$, posons :

- $x_i^k = 1$ si $x_i \leq X_k < x_i + \mathbf{w}_i$; 0 sinon. Et, $w^k = X_{k+1} - X_k$, pour tout $1 \leq k \leq m^x$,
- $y_i^l = 1$ si $y_i \leq Y_l < y_i + \mathbf{h}_i$; 0 sinon. Et, $h^l = Y_{l+1} - Y_l$, pour tout $1 \leq l \leq m^y$,
- $\phi_i^k = (w^k x_i^k) / \mathbf{w}_i$ si $k \neq k_i$; $\phi_i^{k_i} = 1 - \sum_{k < k_i} \phi_i^k \leq (w^{k_i} x_i^{k_i}) / \mathbf{w}_i$ pour $k \in \mathcal{Q}_x$,
- $\varphi_i^l = (h^l y_i^l) / \mathbf{h}_i$ si $l \neq l_i$; $\varphi_i^{l_i} = 1 - \sum_{l < l_i} \varphi_i^l \leq (h^{l_i} y_i^{l_i}) / \mathbf{h}_i$ pour $l \in \mathcal{Q}_y$,
- $x_0^k = 1$ si $k > m^x$ et $y_0^l = 1$ si $l > m^y$, pour $k \in \mathcal{Q}_x$, $l \in \mathcal{Q}_y$,
- $d_i^k = 1$ si $x_i^k = 1$, $x_i^{k-1} = 0$; 0 sinon. Et $f_i^k = 1$ si $x_i^k = 1$, $x_i^{k+1} = 0$; 0 sinon pour tout $k \in \mathcal{Q}_x$. Donc, $f_i^{k_i} = 1$.
- $e_i^l = 1$ si $y_i^l = 1$, $y_i^{l-1} = 0$; 0 sinon. Et $g_i^l = 1$ si $y_i^l = 1$, $y_i^{l+1} = 0$; 0 sinon pour tout $l \in \mathcal{Q}_y$. Donc, $g_i^{l_i} = 1$.

Nous démontrons que cette solution est valide pour la formulation $F^{w^k h^l}$ en utilisant de manière quasiment similaire que la preuve du lemme 2.23 en utilisant les preuves des lemmes 2.25 et 2.26. De même, les contraintes (2.20a)-(2.20b) et (2.21a)-(2.21h) restent vérifiées par cette solution. Il nous reste à démontrer la validité des contraintes (2.22a)-(2.22d).

Si $k > m^x$, alors $x_0^k = 1$ et $d_i^k = f_i^k = 0$ pour tout $i \in \mathcal{I}$; et dans ce cas, les contraintes (2.22a)-(2.22b) sont vérifiées. Soit $k \leq m^x$, alors, par construction et par la condition « calée en bas à gauche », il existe $i \in \mathcal{I}$ tel que $x_i^k = 1$ et $x_i^{k-1} = 0$. D'où, $d_i^k = 1$. De plus, il existe $i \in \mathcal{I}$ tel que $x_i^k = 1$ et $x_i^{k+1} = 0$. D'où $f_i^k = 1$. De plus, $x_0^k = 0$. Donc, les contraintes (2.22a)-(2.22b) sont vérifiées. Similairement, nous démontrons que les contraintes (2.22c)-(2.22d) sont vérifiées. $\square$

Remarque Nous supprimons les contraintes (2.21a)-(2.21b) car elles sont dominées par les contraintes (2.22a)-(2.22d) d'après les contraintes (2.19g)-(2.19j). $\blacklozenge$

Nous utilisons aussi les contraintes valides suivantes :

$$\sum_{k' \leq k} d_i^{k'} + \sum_{k' \geq k} f_i^{k'} = \delta_i + x_i^k, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.23a)$$

$$\sum_{l' \leq l} e_i^{l'} + \sum_{l' \geq l} g_i^{l'} = \delta_i + y_i^l, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y. \quad (2.23b)$$

Lemme 2.31

Les contraintes (2.23a)-(2.23b) sont valides.

Preuve. Soit une solution entière de $F^{x_{ia} \gamma_{ib}}$ vérifiant les contraintes linéaires précédemment citées. Soit $i \in \mathcal{I}$ et $k \in \mathcal{Q}_x$. Nous rencontrons trois cas :

- Si $\delta_i = 0$ et $x_i^k = 0$, alors $d_i^{k'} = f_i^{k'} = 0$ pour tout $k' \in \mathcal{Q}_x$ et les contraintes sont vérifiées.

- Si $\delta_i = 1$ et $x_i^k = 0$, alors, d'après la contrainte (2.19i), il existe $k' \in \mathcal{Q}_x$ et $k'' \in \mathcal{Q}_x$ tel que $d_i^{k'} = f_i^{k''} = 1$. Si $k' \leq k \leq k''$, alors, par la contrainte (2.19g), $x_i^k = 1$. Ceci est absurde. Si $k' > k$ ou $k'' < k$, alors, $\sum_{k' \leq k} d_i^{k'} + \sum_{k' \geq k} f_i^{k'} = 1$. D'où, la contrainte est vérifiée.
- Si $\delta_i = 1$ et $x_i^k = 1$, alors, d'après la contrainte (2.19g), il existe $k' \in \mathcal{Q}_x$ et $k'' \in \mathcal{Q}_x$ tel que $d_i^{k'} = f_i^{k''} = 1$. Si $k < k'$, alors c'est absurde par la contrainte (2.19i). Si $k > k''$, c'est aussi absurde par la contrainte (2.19g). D'où, $\sum_{k' \leq k} d_i^{k'} + \sum_{k' \geq k} f_i^{k'} = 2$. Donc, la contrainte (2.23a) est vérifiée. Similairement, la contrainte (2.23a) est vérifiée. $\square$

Ces dernières contraintes permettent de remplacer l'ensemble des variables x_i^k et y_i^l dans la formulation. Il suffit alors de rajouter les contraintes suivantes :

$$\sum_{k' \leq k} d_i^{k'} + \sum_{k' \geq k} f_i^{k'} \geq \delta_i, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.24a)$$

$$\sum_{l' \leq l} e_i^{l'} + \sum_{l' \geq l} g_i^{l'} \geq \delta_i, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y. \quad (2.24b)$$

D'après toutes les contraintes définies précédemment, nous obtenons une nouvelle formulation compacte améliorée notée $F_{renf}^{w^k h^l}$:

FORMULATION 10: Nouvelle formulation compacte améliorée $[F_{renf}^{w^k h^l}]$

$$[F_{renf}^{w^k h^l}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.25a)$$

$$\text{s. t.} \quad \sum_{k \in \mathcal{Q}_x} \phi_i^k = \delta_i, \quad \sum_{k \in \mathcal{Q}_x} d_i^k = \delta_i, \quad \sum_{k \in \mathcal{Q}_x} f_i^k = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.25b)$$

$$\sum_{l \in \mathcal{Q}_y} \varphi_i^l = \delta_i, \quad \sum_{l \in \mathcal{Q}_y} e_i^l = \delta_i, \quad \sum_{l \in \mathcal{Q}_y} g_i^l = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.25c)$$

$$\mathbf{w}_i \phi_i^k \leq w^k, \quad \sum_{k' \leq k} d_i^{k'} + \sum_{k' \geq k} f_i^{k'} \geq \delta_i, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.25d)$$

$$\mathbf{h}_i \varphi_i^l \leq h^l, \quad \sum_{l' \leq l} e_i^{l'} + \sum_{l' \geq l} g_i^{l'} \geq \delta_i, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.25e)$$

$$\phi_i^k \leq \sum_{k' \leq k} d_i^{k'} + \sum_{k' \geq k} f_i^{k'} - \delta_i, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.25f)$$

$$\varphi_i^l \leq \sum_{l' \leq l} e_i^{l'} + \sum_{l' \geq l} g_i^{l'} - \delta_i, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.25g)$$

$$\sum_{k \in \mathcal{Q}_x} w^k \leq \mathbf{W}, \quad \sum_{l \in \mathcal{Q}_y} h^l \leq \mathbf{H}, \quad (2.25h)$$

$$\sum_{i \in \mathcal{I}} \mathbf{h}_i \left(\sum_{k' \leq k} d_i^{k'} + \sum_{k' \geq k} f_i^{k'} - \delta_i \right) \leq \mathbf{H}, \quad \forall k \in \mathcal{Q}_x, \quad (2.25i)$$

$$\sum_{i \in \mathcal{I}} \mathbf{w}_i \left(\sum_{l' \leq l} e_i^{l'} + \sum_{l' \geq l} g_i^{l'} - \delta_i \right) \leq \mathbf{W}, \quad \forall l \in \mathcal{Q}_y, \quad (2.25j)$$

$$x_{ij} \geq \sum_{k' \leq k} d_i^{k'} + \sum_{k' \geq k} f_i^{k'} - \delta_i + \sum_{k' \leq k} d_j^{k'} + \sum_{k' \geq k} f_j^{k'} - \delta_j - 1, \quad \forall i, j \in \mathcal{I} (i < j), k \in \mathcal{Q}_x, \quad (2.25k)$$

$$y_{ij} \geq \sum_{l' \leq l} e_i^{l'} + \sum_{l' \geq l} g_i^{l'} - \delta_i + \sum_{l' \leq l} e_j^{l'} + \sum_{l' \geq l} g_j^{l'} - \delta_j - 1, \quad \forall i, j \in \mathcal{I} (i < j), l \in \mathcal{Q}_y, \quad (2.25l)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.25m)$$

$$\sum_{i \in \mathcal{I}} d_i^k + x_0^k \geq 1, \quad \sum_{i \in \mathcal{I}} f_i^k + x_0^k \geq 1, \quad \forall k \in \mathcal{Q}_x, \quad (2.25n)$$

$$\sum_{i \in \mathcal{I}} e_i^l + y_0^l \geq 1, \quad \sum_{i \in \mathcal{I}} g_i^l + y_0^l \geq 1, \quad \forall l \in \mathcal{Q}_y, \quad (2.25o)$$

$$d_i^k + x_0^k \leq 1, \quad x_0^k \leq x_0^{k+1}, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.25p)$$

$$e_i^l + y_0^l \leq 1, \quad y_0^l \leq y_0^{l+1}, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.25q)$$

$$K_x - \sum_{k \in \mathcal{Q}_x} x_0^k \leq \sum_{i \in \mathcal{I}} \delta_i, \quad (2.25r)$$

$$K_y - \sum_{l \in \mathcal{Q}_y} y_0^l \leq \sum_{i \in \mathcal{I}} \delta_i, \quad (2.25s)$$

$$x_{ij}, y_{ij}, \delta_i \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.25t)$$

$$d_i^k, f_i^k \in \{0, 1\}, \quad \phi_i^k \geq 0, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.25u)$$

$$e_i^l, g_i^l \in \{0, 1\}, \quad \varphi_i^k \geq 0, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.25v)$$

$$0 \leq w^k \leq \mathbf{W}, \quad x_0^k \in \{0, 1\}, \quad \forall k \in \mathcal{Q}_x, \quad (2.25w)$$

$$0 \leq h^l \leq \mathbf{H}, \quad y_0^l \in \{0, 1\}, \quad \forall l \in \mathcal{Q}_y. \quad (2.25x)$$

Il est aussi possible de formuler un modèle relâché basé sur la suppression des dépendances entre les cliques en vue d'une future décomposition.

FORMULATION 11: Nouvelle formulation compacte relâchée $[F_{relax}^{w^k h^l}]$

$$[F_{relax}^{w^k h^l}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.26a)$$

$$\text{s. t.} \quad \sum_{k \in \mathcal{Q}_x} \phi_i^k \geq \delta_i, \quad \sum_{l \in \mathcal{Q}_y} \varphi_i^l \geq \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.26b)$$

$$\phi_i^k \leq x_i^k, \quad \mathbf{w}_i \phi_i^k \leq w^k, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.26c)$$

$$\varphi_i^l \leq y_i^l, \quad \mathbf{h}_i \varphi_i^l \leq h^l, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.26d)$$

$$\sum_{i \in \mathcal{I}} \mathbf{h}_i x_i^k \leq \mathbf{H}, \quad \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i^l \leq \mathbf{W}, \quad \forall k \in \mathcal{Q}_x, \quad (2.26e)$$

$$\sum_{k \in \mathcal{Q}_x} w^k \leq \mathbf{W}, \quad \sum_{l \in \mathcal{Q}_y} h^l \leq \mathbf{H}, \quad (2.26f)$$

$$\sum_{k \in \mathcal{Q}_x} (x_i^k + x_j^k - 1) \leq K_x x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.26g)$$

$$\sum_{l \in \mathcal{Q}_y} (y_i^l + y_j^l - 1) \leq K_y y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.26h)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.26i)$$

$$x_{ij}, y_{ij}, \delta_i \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.26j)$$

$$x_i^k \in \{0, 1\}, \quad \phi_i^k \geq 0, \quad \forall i \in \mathcal{I}, k \in \mathcal{Q}_x, \quad (2.26k)$$

$$y_i^l \in \{0, 1\}, \quad \varphi_i^l \geq 0, \quad \forall i \in \mathcal{I}, l \in \mathcal{Q}_y, \quad (2.26l)$$

$$0 \leq w^k \leq \mathbf{W}, \quad \forall k \in \mathcal{Q}_x, \quad (2.26m)$$

$$0 \leq h^l \leq \mathbf{H}, \quad \forall l \in \mathcal{Q}_y. \quad (2.26n)$$

2.4 Décomposition des formulations

Certaines des formulations précédemment définies peuvent être décomposées. Nous allons exposer les décompositions les plus intéressantes.

Par exemple, pour les formulations utilisant une discrétisation de l'espace, seules celles séparant totalement les décisions sur les abscisses et sur les ordonnées sont les formulations facilement décomposables. En effet, pour ces formulations, nous observons des systèmes de matrices blocs pour chaque abscisse et chaque ordonnée. Donc, seules les formulations $F_{relax}^{x_{ia} y_{ib}}$ et $F_{relax}^{x_{ia} y_{ib}}$ sont décomposées. Pour les formulations compactes, seules celles utilisant un décompage par cliques maximales sont intéressantes à étudier.

2.4.1 Lexique des variables pour les décompositions des formulations utilisant une discrétisation de l'espace

Un placement peut être vu comme un remplissage du conteneur par des bandes verticales (resp. horizontales) de largeur un à chaque abscisse (resp. ordonnée). Dans ce cas, de nouvelles formulations sont possibles impliquant de nouvelles variables.

Nous définissons l'ensemble des bandes verticales réalisables, noté $\mathcal{V}$, comme étant l'ensemble des sous-ensembles d'objets ne dépassant la hauteur du conteneur :

$$\mathcal{V} = \left\{ x \in \{0, 1\}^n : \sum_{i \in \mathcal{I}} \mathbf{h}_i x_i \leq \mathbf{H} \right\}.$$

Nous définissons de manière analogue l'ensemble des bandes horizontales possibles, noté $\mathcal{H}$, comme étant l'ensemble des sous-ensemble d'objets ne dépassant la largeur du conteneur :

$$\mathcal{H} = \left\{ y \in \{0, 1\}^n : \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i \leq \mathbf{W} \right\}.$$

Interprétation des éléments composants une bande :

Toute bande verticale $v \in \mathcal{V}$ est définie par un vecteur caractéristique :

$$x_i^v = \begin{cases} 1 & \text{si l'objet } i \text{ est présent dans la bande verticale } v \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I},$$

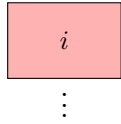
Toute bande horizontale $h \in \mathcal{H}$ est définie par un vecteur caractéristique :

$$y_i^h = \begin{cases} 1 & \text{si l'objet } i \text{ est présent dans la bande horizontale } h \\ 0 & \text{sinon} \end{cases}, \quad \forall i \in \mathcal{I}.$$

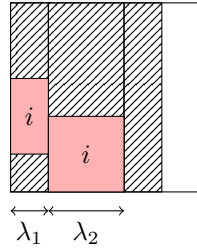
◆

Chaque bande $v \in \mathcal{V}$ (resp. $h \in \mathcal{H}$) est vue comme un tronçon servant à remplir la boîte sur toute sa verticale (resp. son horizontal). Le nombre de bandes à sélectionner est alors représenté par des variables (cf. figure 2.3).

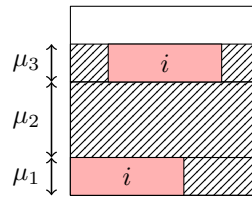
Objets



⋮



(a) bandes verticales



(b) bandes horizontales

FIGURE 2.3: Exemple de remplissage d'un conteneur par des bandes verticales et horizontales

Nous associons donc à chaque bande $v \in \mathcal{V}$ une variable qui représente la quantité de bandes verticales de type v sélectionnée. Similairement, nous définissons à chaque bande $h \in \mathcal{H}$ une variable représentant la quantité de bandes horizontales de type h sélectionnée :

$$\begin{cases} \lambda_v \geq 0 \text{ représente le nombre de bandes verticales de type } v, & \forall v \in \mathcal{V}, \\ \mu_h \geq 0 \text{ représente le nombre de bandes horizontales de type } h, & \forall h \in \mathcal{H}. \end{cases}$$

Remarque Soit $v \in \mathcal{V}$ (resp. $h \in \mathcal{H}$). Si $\lambda_v = 0$ (resp. $\mu_h = 0$), alors la bande v (resp. h) est dite non-sélectionnée. Sinon, la bande v (resp. h) est dite sélectionnée. ◆

De plus, une abscisse (ou ordonnée) peut être associée à chaque bande verticale (ou horizontale) sélectionnée. Soient, $\mathcal{V}^a$ (resp. $\mathcal{H}^b$) l'ensemble des bandes verticales (resp. horizontales) possibles en abscisse a (resp. en coordonnée b). Nous associons alors les variables suivantes :

$$\left\{ \begin{array}{l} \lambda_v^a \geq 0 \text{ est la portion de la bande verticale de type } v \text{ sélectionnée en abscisse } a \\ \mu_h^b \geq 0 \text{ est la portion de la bande horizontale de type } h \text{ sélectionnée en ordonnée } b \end{array} \right. \quad \begin{array}{l} \forall a \in \mathcal{X}, v \in \mathcal{V}^a, \\ \forall b \in \mathcal{Y}, h \in \mathcal{H}^b. \end{array}$$

Il est également possible d'enrichir les sous-problèmes en introduisant les variables de chevauchement des objets x_{ij} et y_{ij} . Nous obtenons alors les sous-ensembles suivants, où $\mathcal{V}_{enr}$ représente l'ensemble des bandes verticales avec des variables de chevauchement des objets selon l'axe des abscisses et $\mathcal{H}_{enr}$ représente l'ensemble des bandes horizontales avec des variables de chevauchement des objets selon l'axe des ordonnées :

$$\mathcal{V}_{enr} = \left\{ x : \begin{array}{l} \sum_{i \in \mathcal{I}} \mathbf{h}_i x_i \leq \mathbf{H}; \quad x_{ij} \leq x_i \text{ et } x_{ij} \leq x_j \quad \forall i, j \in \mathcal{I}(i < j); \\ x_{ij} \geq x_i + x_j - 1 \quad \forall i, j \in \mathcal{I}(i < j); \quad x_i, x_{ij} \in \{0, 1\} \quad \forall i, j \in \mathcal{I}(i < j) \end{array} \right\},$$

$$\mathcal{H}_{enr} = \left\{ y : \begin{array}{l} \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i \leq \mathbf{W}; \quad y_{ij} \leq y_i \text{ et } y_{ij} \leq y_j \quad \forall i, j \in \mathcal{I}(i < j); \\ y_{ij} \geq y_i + y_j - 1 \quad \forall i, j \in \mathcal{I}(i < j); \quad y_i, y_{ij} \in \{0, 1\} \quad \forall i, j \in \mathcal{I}(i < j) \end{array} \right\}.$$

Interprétation des éléments composants une bande :

Toute bande verticale $v \in \mathcal{V}_{enr}$ est définie par un vecteur caractéristique :

$$\begin{aligned} x_i^v &= \begin{cases} 1 & \text{si l'objet } i \text{ est dans la bande verticale } v \\ 0 & \text{sinon} \end{cases}, & \forall i \in \mathcal{I}, \\ x_{ij}^v &= \begin{cases} 1 & \text{si les objets } i \text{ et } j \text{ sont présents dans la bande verticale } v \\ 0 & \text{sinon} \end{cases}, & \forall i, j \in \mathcal{I}, \end{aligned}$$

Toute bande horizontale $h \in \mathcal{H}_{enr}$ est définie par un vecteur caractéristique :

$$\begin{aligned} y_i^h &= \begin{cases} 1 & \text{si l'objet } i \text{ est dans la bande horizontale } h \\ 0 & \text{sinon} \end{cases}, & \forall i \in \mathcal{I}, \\ y_{ij}^h &= \begin{cases} 1 & \text{si les objets } i \text{ et } j \text{ sont présents dans la bande horizontale } h \\ 0 & \text{sinon} \end{cases}, & \forall i, j \in \mathcal{I}. \end{aligned}$$

◆

Les variables associées à chaque bande sont identiques aux variables précédemment définies.

2.4.2 Décomposition de la formulation $\mathbf{F}^{x_{ia}y_{ib}}$

La première solution consiste à décomposer la formulation $\mathbf{F}^{x_{ia}y_{ib}}$ de la page 59 en utilisant les contraintes (2.12j)-(2.12k) comme des contraintes de définition du sous-problème. Nous obtenons alors une première formulation étendue.

Les espaces de solution pour les sous-problèmes sont définis par les ensembles $\mathcal{V}^a$ et $\mathcal{H}^b$ définis précédemment. Les autres contraintes sont dualisées. Cette formulation a pour la première fois été introduite par Boschetti, Mingozzi et Hadjiconstantinou [12] et est définie

de la manière suivante :

FORMULATION 12: Problème maître $[\text{FE}^{\lambda_v^a \mu_h^b}]$ issu de la formulation $\text{F}^{x_{ia} y_{ib}}$

$$[\text{FE}^{\lambda_v^a \mu_h^b}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.27a)$$

$$\text{s. t.} \quad \sum_{a \in \mathcal{X}_i} \chi_{ia} = \sum_{b \in \mathcal{Y}_i} \gamma_{ib} = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.27b)$$

$$\sum_{v \in \mathcal{V}^a} x_i^v \lambda_v^a = \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = b - \mathbf{w}_i + 1}}^a \chi_{ia'}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}, \quad (\nu_{ia}) \quad (2.27c)$$

$$\sum_{h \in \mathcal{H}^b} y_i^h \mu_h^b = \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib'}, \quad \forall i \in \mathcal{I}, b \in \mathcal{Y}, \quad (\tau_{ib}) \quad (2.27d)$$

$$\sum_{v \in \mathcal{V}^a} (x_i^v + x_j^v - 1) \lambda_v^a \leq x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), a \in \mathcal{X}, \quad (\pi_{ija}) \quad (2.27e)$$

$$\sum_{h \in \mathcal{H}^b} (y_i^h + y_j^h - 1) \mu_h^b \leq y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), b \in \mathcal{Y}, \quad (\varrho_{ijb}) \quad (2.27f)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.27g)$$

$$\sum_{v \in \mathcal{V}^a} \lambda_v^a \leq 1, \quad \forall a \in \mathcal{X}, \quad (\sigma_a) \quad (2.27h)$$

$$\sum_{h \in \mathcal{H}^b} \mu_h^b \leq 1, \quad \forall b \in \mathcal{Y}, \quad (\psi_b) \quad (2.27i)$$

$$\delta_i, x_{ij}, y_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.27j)$$

$$\chi_{ia}, \gamma_{ib} \in \{0, 1\}, \quad \forall i \in \mathcal{I} (i < j), a \in \mathcal{X}_i, b \in \mathcal{Y}_i, \quad (2.27k)$$

$$\lambda_v^a \geq 0, \quad \forall a \in \mathcal{X}, v \in \mathcal{V}^a, \quad (2.27l)$$

$$\mu_h^b \geq 0, \quad \forall b \in \mathcal{Y}, h \in \mathcal{H}^b. \quad (2.27m)$$

Les sous-problèmes sont alors des problèmes de sac-à-dos définis de la manière suivante pour chaque abscisse et chaque ordonnée :

FORMULATION 13: Sous-problèmes $[\mathcal{V}^a \ \& \ \mathcal{H}^b]$

$$\forall a \in \mathcal{X} : \quad [\mathcal{V}^a] \quad \max \sum_{i \in \mathcal{I}} \nu_{ia} x_i^a - \sum_{i, j \in \mathcal{I}: i < j} \pi_{ija} (x_i^a + x_j^a) - \sigma_a, \quad (2.28a)$$

$$\text{s. t.} \quad \sum_{i \in \mathcal{I}} \mathbf{h}_i x_i^a \leq \mathbf{H}, \quad (2.28b)$$

$$x_i^a \in \{0, 1\}, \quad \forall i \in \mathcal{I}, \quad (2.28c)$$

$$\forall b \in \mathcal{Y} : \quad [\mathcal{H}^b] \quad \max \sum_{i \in \mathcal{I}} \tau_{ib} y_i^b - \sum_{i, j \in \mathcal{I}: i < j} \varrho_{ijb} (y_i^b + y_j^b) - \psi_b, \quad (2.28d)$$

$$\text{s. t.} \quad \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i^b \leq \mathbf{W}, \quad (2.28e)$$

$$y_i^b \in \{0, 1\}, \quad \forall i \in \mathcal{I}. \quad (2.28f)$$

Une autre approche consiste à prendre les contraintes (2.12e)-(2.12g) et (2.12j)-(2.12k)

comme des contraintes définissant le sous-problème. Nous déduisons une nouvelle décomposition. Cependant, pour chacune de ces décompositions, un sous-problème est défini pour chaque abscisse et chaque ordonnée. Le nombre de sous-problème à résoudre est pseudo-polynômial. Donc, de telles formulations paraissent difficilement envisageable. Il paraît donc plus réalisable d'utiliser la formulation $F_{relax}^{x_{ia}y_{ib}}$ pour modèle.

2.4.3 Décomposition de la formulation $F_{relax}^{x_{ia}y_{ib}}$

La première approche consiste à décomposer la formulation $F_{relax}^{x_{ia}y_{ib}}$ de la page 60 en considérant les contraintes (2.13i)-(2.13j) comme contraintes de définition du sous-problème. Cette formulation permet d'obtenir une borne duale valide pour le problème d'origine. Les sous-problèmes obtenus sont alors des problèmes de sac-à-dos similaires aux sous-problèmes développés précédemment (sous problèmes associés à $\mathcal{V}$ et $\mathcal{H}$).

FORMULATION 14: Problème maître $[FE^{\lambda_v\mu_h}]$ issu de la formulation $F_{relax}^{x_{ia}y_{ib}}$

$$\begin{aligned}
 [FE^{\lambda_v\mu_h}] \quad & \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, & (2.29a) \\
 \text{s. t.} \quad & \sum_{v \in \mathcal{V}} x_i^v \lambda_v = \mathbf{w}_i \delta_i, & \forall i \in \mathcal{I}, \quad (\nu_i) \quad (2.29b) \\
 & \sum_{h \in \mathcal{H}} y_i^h \mu_h = \mathbf{h}_i \delta_i, & \forall i \in \mathcal{I}, \quad (\tau_i) \quad (2.29c) \\
 & \sum_{v \in \mathcal{V}} (x_i^v + x_j^v - 1) \lambda_v \leq \min\{\mathbf{w}_i, \mathbf{w}_j\} x_{ij}, & \forall i, j \in \mathcal{I} (i < j), \quad (\pi_{ij}) \quad (2.29d) \\
 & \sum_{h \in \mathcal{H}} (y_i^h + y_j^h - 1) \mu_h \leq \min\{\mathbf{h}_i, \mathbf{h}_j\} y_{ij}, & \forall i, j \in \mathcal{I} (i < j), \quad (\varrho_{ij}) \quad (2.29e) \\
 & x_{ij} + y_{ij} \leq 1, & \forall i, j \in \mathcal{I} (i < j), \quad (2.29f) \\
 & \sum_{v \in \mathcal{V}} \lambda_v \leq \mathbf{W}, & (\sigma) \quad (2.29g) \\
 & \sum_{h \in \mathcal{H}} \mu_h \leq \mathbf{H}, & (\psi) \quad (2.29h) \\
 & \delta_i, x_{ij}, y_{ij} \in \{0, 1\}, & \forall i, j \in \mathcal{I} (i < j), \quad (2.29i) \\
 & \lambda_v \geq 0, & \forall v \in \mathcal{V}, \quad (2.29j) \\
 & \mu_h \geq 0, & \forall h \in \mathcal{H}. \quad (2.29k)
 \end{aligned}$$

La suppression des contraintes (2.29d)-(2.29f) permet d'obtenir la formulation développée par Scheithauer [56], notée dans ce mémoire $FE_{relax}^{\lambda_v\mu_h}$. La formulation donnée par Tsai, Malstorm et Meeks [58] est obtenue en supprimant aussi les contraintes (2.29c), (2.29h) et (2.29j).

Alors, voici les sous-problèmes associés à la formulation. Il s'agit de résoudre deux

problèmes de sac-à-dos :

FORMULATION 15: Sous-problèmes $[\mathcal{V} \ \& \ \mathcal{H}]$

$$[\mathcal{V}] \quad \max \sum_{i \in \mathcal{I}} \nu_i x_i - \sum_{i,j \in \mathcal{I}: i < j} \pi_{ij} (x_i + x_j) - \sigma, \quad (2.30a)$$

$$\text{s. t. } \sum_{i \in \mathcal{I}} \mathbf{h}_i x_i \leq \mathbf{H}, \quad (2.30b)$$

$$x_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}, \quad (2.30c)$$

$$[\mathcal{H}] \quad \max \sum_{i \in \mathcal{I}} \tau_i y_i - \sum_{i,j \in \mathcal{I}: i < j} \varrho_{ij} (y_i + y_j) - \psi, \quad (2.30d)$$

$$\text{s. t. } \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i \leq \mathbf{W}, \quad (2.30e)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}. \quad (2.30f)$$

La deuxième approche consiste à considérer les contraintes (2.13d)-(2.13f) et (2.13i)-(2.13j) comme des contraintes du sous-problème. Les autres contraintes sont dualisées. Cette formulation a été étudié durant ma thèse par Belov et Rohling [9] et est définie de la manière suivante :

FORMULATION 16: Problème maître $[\text{FE}_{enr}^{\lambda_v \mu_h}]$ issu de la formulation $\text{F}_{relax}^{x_{ia} y_{ib}}$

$$[\text{FE}_{enr}^{\lambda_v \mu_h}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.31a)$$

$$\text{s. t. } \sum_{v \in \mathcal{V}_{enr}} x_i^v \lambda_v = \mathbf{w}_i \delta_i, \quad \forall i \in \mathcal{I}, \quad (\nu_i) \quad (2.31b)$$

$$\sum_{h \in \mathcal{H}_{enr}} y_i^h \mu_h = \mathbf{h}_i \delta_i, \quad \forall i \in \mathcal{I}, \quad (\tau_i) \quad (2.31c)$$

$$\frac{1}{\min\{\mathbf{w}_i, \mathbf{w}_j\}} \left(\sum_{v \in \mathcal{V}_{enr}} x_{ij}^v \lambda_v \right) + \frac{1}{\min\{\mathbf{h}_i, \mathbf{h}_j\}} \left(\sum_{h \in \mathcal{H}_{enr}} y_{ij}^h \mu_h \right) \leq 1, \quad \forall i, j \in \mathcal{I} (i < j), \quad (\omega_{ij}) \quad (2.31d)$$

$$\sum_{v \in \mathcal{V}_{enr}} \lambda_v \leq \mathbf{W}, \quad (\sigma) \quad (2.31e)$$

$$\sum_{h \in \mathcal{H}_{enr}} \mu_h \leq \mathbf{H}, \quad (\psi) \quad (2.31f)$$

$$\delta_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}, \quad (2.31g)$$

$$\lambda_v \geq 0, \quad \forall v \in \mathcal{V}_{enr}, \quad (2.31h)$$

$$\mu_h \geq 0, \quad \forall h \in \mathcal{H}_{enr}. \quad (2.31i)$$

La définition des sous-problèmes est alors donné par les ensembles $\mathcal{V}_{enr}$ et $\mathcal{H}_{enr}$. Il s'agit de problèmes de sac-à-dos avec une variable de liaison entre les objets, donc plus difficile à

résoudre :

FORMULATION 17: Sous-problèmes $[\mathcal{V}_{enr}]$ & $[\mathcal{H}_{enr}]$

$$[\mathcal{V}_{enr}] \quad \max \sum_{i \in \mathcal{I}} \nu_i x_i - \sum_{i,j \in \mathcal{I}: i < j} \omega_{ij} x_{ij} - \sigma, \quad (2.32a)$$

$$\text{s. t. } \sum_{i \in \mathcal{I}} \mathbf{h}_i x_i \leq \mathbf{H}, \quad (2.32b)$$

$$x_i + x_j - 1 \leq x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.32c)$$

$$x_i \in \{0, 1\}, x_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.32d)$$

$$[\mathcal{H}_{enr}] \quad \max \sum_{i \in \mathcal{I}} \tau_i y_i - \sum_{i,j \in \mathcal{I}: i < j} \omega_{ij} y_{ij} - \psi, \quad (2.32e)$$

$$\text{s. t. } \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i \leq \mathbf{W}, \quad (2.32f)$$

$$y_i + y_j - 1 \leq y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.32g)$$

$$y_i \in \{0, 1\}, y_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j). \quad (2.32h)$$

2.4.4 Lexique des variables pour les décompositions des formulations compactes

La formulation $F^{c_{ij}d_{ij}}$ de la page 44 offre peu de possibilités de décomposition car aucune structure par blocs impliquant des variables entières n'apparaît clairement identifiable.

Concernant la formulation $F^{w^k h^l}$ de la page 73, il est possible d'effectuer une décomposition. En effet, un placement peut être vu comme un remplissage du conteneur par des bandes verticales (resp. horizontales) de largeurs variables associées à une position selon l'axe des abscisses (resp. ordonnées). Nous définissons donc de nouveaux ensembles de bandes verticales et horizontales de largeur variable. Soient, $\mathcal{V}_{var}$ (resp. $\mathcal{H}_{var}$) l'ensemble des bandes verticales (horizontales) de largeurs variables possibles.

$$\mathcal{V}_{var} = \left\{ x \in \{0, 1\}^n, \phi \in \mathbb{R}^n, w \in \mathbb{R} : \sum_{i \in \mathcal{I}} \mathbf{h}_i x_i \leq \mathbf{H}, \right. \\ \left. \phi_i \leq x_i \forall i \in \mathcal{I}, \mathbf{w}_i \phi_i \leq w \forall i \in \mathcal{I} \right\},$$

$$\mathcal{H}_{var} = \left\{ y \in \{0, 1\}^n, \varphi \in \mathbb{R}^n, h \in \mathbb{R} : \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i \leq \mathbf{W}, \right. \\ \left. \varphi_i \leq y_i \forall i \in \mathcal{I}, \mathbf{h}_i \varphi_i \leq h \forall i \in \mathcal{I} \right\}.$$

Nous associons alors des variables définissant le nombre de bandes sélectionné :

$$\begin{cases} \lambda_v \geq 0 \text{ représente le nombre de bandes verticales de type } v, & \forall v \in \mathcal{V}, \\ \mu_h \geq 0 \text{ représente le nombre de bandes horizontales de type } h, & \forall h \in \mathcal{H}. \end{cases}$$

2.4.5 Décomposition de la formulation $F_{relax}^{w^k h^l}$

En considérant la formulation $F_{relax}^{w^k h^l}$, nous obtenons la décomposition suivante en prenant les contraintes (2.26e) comme contraintes définissant le sous-problème :

FORMULATION 18: Problème maître $[FE_{var}^{\lambda_v \mu_h}]$ issu de la formulation $F_{relax}^{w^k h^l}$

$$[FE_{var}^{\lambda_v \mu_h}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.33a)$$

$$\text{s. t.} \quad \sum_{v \in \mathcal{V}_{var}} \phi_i^v \lambda_v \geq \delta_i, \quad \forall i \in \mathcal{I}, \quad (\nu_i) \quad (2.33b)$$

$$\sum_{h \in \mathcal{H}_{var}} \varphi_i^h \mu_h \geq \delta_i, \quad \forall i \in \mathcal{I}, \quad (\tau_i) \quad (2.33c)$$

$$\sum_{v \in \mathcal{V}_{var}} w^v \lambda_v \leq \mathbf{W}, \quad (\eta) \quad (2.33d)$$

$$\sum_{h \in \mathcal{H}_{var}} h^h \mu_h \leq \mathbf{H}, \quad (\theta) \quad (2.33e)$$

$$\sum_{v \in \mathcal{V}_{var}} (x_i^v + x_j^v - 1) \lambda_v \leq K_x x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (\pi_{ij}) \quad (2.33f)$$

$$\sum_{h \in \mathcal{H}_{var}} (y_i^h + y_j^h - 1) \mu_h \leq K_y y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (\varrho_{ij}) \quad (2.33g)$$

$$x_{ij} + y_{ij} \leq 1, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.33h)$$

$$\sum_{v \in \mathcal{V}_{var}} \lambda_v \leq \sum_{i \in \mathcal{I}} \delta_i \leq K_x, \quad (\sigma) \quad (2.33i)$$

$$\sum_{h \in \mathcal{H}_{var}} \mu_h \leq \sum_{i \in \mathcal{I}} \delta_i \leq K_y, \quad (\psi) \quad (2.33j)$$

$$\delta_i, x_{ij}, y_{ij} \in \{0, 1\}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (2.33k)$$

$$0 \leq \lambda_v \leq 1, \quad 0 \leq \mu_h \leq 1, \quad \forall v \in \mathcal{V}_{var}, h \in \mathcal{H}_{var}. \quad (2.33l)$$

Les sous problèmes sont alors :

FORMULATION 19: Sous-problèmes $[\mathcal{V}_{var} \ \& \ \mathcal{H}_{var}]$

$$[\mathcal{V}_{var}] \quad \max \sum_{i \in \mathcal{I}} \nu_i \phi_i - \eta w - \sum_{i, j \in \mathcal{I}: i < j} \pi_{ij} x_{ij} - \sigma, \quad (2.34a)$$

$$\text{s. t.} \quad \sum_{i \in \mathcal{I}} \mathbf{h}_i x_i \leq \mathbf{H}, \quad (2.34b)$$

$$\mathbf{w}_i \phi_i \leq \mathbf{W} x_i, \quad \forall i \in \mathcal{I}, \quad (2.34c)$$

$$\mathbf{w}_i \phi_i \leq w x_i, \quad \forall i \in \mathcal{I}, \quad (2.34d)$$

$$x_i \in \{0, 1\}, \phi_i \in \geq 0, w \in \geq 0 \quad \forall i \in \mathcal{I}, \quad (2.34e)$$

$$[\mathcal{H}_{var}] \quad \max \sum_{i \in \mathcal{I}} \tau_i \varphi_i - \theta h - \sum_{i, j \in \mathcal{I}: i < j} \varrho_{ij} y_{ij} - \psi, \quad (2.34f)$$

$$\text{s. t.} \quad \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i \leq \mathbf{W}, \quad (2.34g)$$

$$\mathbf{h}_i \varphi_i \leq \mathbf{H} y_i, \quad \forall i \in \mathcal{I}, \quad (2.34h)$$

$$\mathbf{h}_i \varphi_i \leq h y_i, \quad \forall i \in \mathcal{I}, \quad (2.34i)$$

$$y_i \in \{0, 1\}, \varphi_i \in \geq 0, h \in \geq 0 \quad \forall i \in \mathcal{I}. \quad (2.34j)$$

Il est possible de relâcher les contraintes (2.33f)-(2.33h). La formulation obtenue est notée $FE_{var,relax}^{\lambda_v \mu_h}$.

Lemme 2.32

La borne duale obtenue par la relaxation linéaire de la formulation $FE_{var,relax}^{\lambda_v \mu_h}$ est plus petite ou égale à celle obtenue par la relaxation linéaire de la formulation F_0 .

Preuve. D'après la preuve du lemme 2.25, nous savons que, pour tout $v \in \mathcal{V}_{var}$:

$$\phi_i^v \leq w^v x_i^v, \quad \forall i \in \mathcal{I}.$$

Donc, en utilisant la contrainte (2.33b) puis la contrainte (2.34b), nous avons :

$$\sum_{i \in \mathcal{I}} \mathbf{w}_i \mathbf{h}_i \delta_i \leq \sum_{i \in \mathcal{I}} \sum_{v \in \mathcal{V}_{var}} \phi_i^v \lambda_v \leq \sum_{v \in \mathcal{V}_{var}} \sum_{i \in \mathcal{I}} x_i^v w^v \lambda_v \leq \mathbf{H} \sum_{v \in \mathcal{V}_{var}} w^v \lambda_v.$$

D'après la contrainte (2.33d), nous avons vérifier que $\sum_{i \in \mathcal{I}} \mathbf{w}_i \mathbf{h}_i \delta_i \leq \mathbf{W}\mathbf{H}$ pour toute solution de $FE_{var,relax}^{\lambda_v \mu_h}$. $\square$

2.5 Comparaison entre les différentes approches

Nous effectuons maintenant une étude théorique comparative entre les différentes formulations. En particulier, nous étudions les différentes relaxations possibles du problème afin d'étudier leurs qualités. Les résultats qui suivent sont obtenus à l'aide de CPLEX 11.

D'après les corollaires 2.11 et 2.21, et le lemme 2.19, nous savons que :

$$U_{F_{renf}^{z_{iab}}} \geq U_{F_{renf}^{\Psi_{iab}}} \text{ et } U_{F_{renf}^{x_{ia}y_{ib}}} \geq U_{F_{renf}^{z_{ab}}} \geq U_{F_{renf}^{\Psi_{iab}}}.$$

Lors de nos expérimentations numériques, nous avons constaté que les bornes des formulations $F^{z_{ab}}$, $F^{z_{iab}}$ et $F^{x_{ia}y_{ib}}$ sont identiques. En réalité, la relaxation linéaire de la formulation suivante, notée $F^{\chi_{ia}\gamma_{ib}}$, est à l'origine de la qualité des bornes obtenues :

FORMULATION 20: Formulation relâché du problème $[F^{\chi_{ia}\gamma_{ib}}]$

$$[F^{\chi_{ia}\gamma_{ib}}] \quad \max \sum_{i \in \mathcal{I}} \rho_i \delta_i, \quad (2.35a)$$

$$\text{s. t. } \sum_{a \in \mathcal{X}_i} \chi_{ia} = \sum_{b \in \mathcal{Y}_i} \gamma_{ib} = \delta_i, \quad \forall i \in \mathcal{I}, \quad (2.35b)$$

$$\sum_{i \in \mathcal{I}} \mathbf{h}_i \sum_{\substack{a' \in \mathcal{X}_i: \\ a' = a - \mathbf{w}_i + 1}}^a \chi_{ia'} \leq \mathbf{H}, \quad \forall a \in \mathcal{X}, \quad (2.35c)$$

$$\sum_{i \in \mathcal{I}} \mathbf{w}_i \sum_{\substack{b' \in \mathcal{Y}_i: \\ b' = b - \mathbf{h}_i + 1}}^b \gamma_{ib'} \leq \mathbf{W}, \quad \forall b \in \mathcal{Y}, \quad (2.35d)$$

$$\delta_i, \chi_{ia}, \gamma_{ib} \in \{0, 1\}, \quad \forall i \in \mathcal{I}, a \in \mathcal{X}_i, b \in \mathcal{Y}_i. \quad (2.35e)$$

La formulation $F^{Xia\gamma ib}$ est une relaxation des formulations F^{zab} , F^{ziab} et $F^{xia\gamma ib}$: les contraintes de $F^{Xia\gamma ib}$ peuvent être retrouvées facilement dans les différentes formulations. La formulation $F^{Xia\gamma ib}$ est une formulation relaxant la contrainte de non-intersection des objets. Notons $F_{renf}^{Xia\gamma ib}$ la formulation obtenue en renforçant la formulation $F^{Xia\gamma ib}$ par les contraintes (2.11a)-(2.11b).

De plus, en sommant la contrainte (2.35c) selon $a \in \mathcal{X}$, nous obtenons que la contrainte suivante est vérifiée par toutes les solutions de $F^{Xia\gamma ib}$: $\sum_{i \in \mathcal{I}} \mathbf{w}_i \mathbf{h}_i \delta_i \leq \mathbf{WH}$. Donc, les bornes duales $U_{F^{zab}}$, $U_{F^{ziab}}$, $U_{F^{xia\gamma ib}}$ et $U_{F^{Xia\gamma ib}}$ sont au moins aussi bonne que la borne duale U_{F_0} . Nous avons donc les relations suivantes :

$$\begin{aligned} U_{F_0} &\geq U_{F_{renf}^{Xia\gamma ib}} \geq U_{F_{renf}^{xia\gamma ib}} \geq U_{F_{renf}^{zab}} \geq U_{F_{renf}^{\Psi iab}} \geq OPT \\ U_{F_0} &\geq U_{F_{renf}^{Xia\gamma ib}} \geq U_{F_{renf}^{ziab}} \geq U_{F_{renf}^{\Psi iab}} \geq OPT \end{aligned}$$

Pour les formulations étendues, en utilisant les papiers de Boschetti et al. [12] et de Below et Rohling [9], nous avons les relations suivantes :

$$U_{F_0} \geq U_{FE_{relax}^{\lambda_v}} \geq U_{FE_{relax}^{\lambda_v \mu_h}} \geq U_{FE^{\lambda_v \mu_h}} \geq U_{FE_{env}^{\lambda_v \mu_h}} \geq OPT$$

D'après le lemme 2.32, nous avons :

$$U_{F_0} \geq U_{FE_{var,relax}^{\lambda_v \mu_h}} \geq U_{FE_{var}^{\lambda_v \mu_h}} \geq OPT$$

Les résultats qui suivent sont effectuées sur une partie des formulations exposées dans ce chapitre pour les instances « ngcut », « gcut », « cgcut », « wang » et « okp ». Pour étudier les qualités de bornes, deux types de relaxations sont considérées : la première est la relaxation linéaire et la deuxième est la relaxation avec des variables δ_i entier pour tout $i \in \mathcal{I}$. Les différents résultats obtenus sont reportés dans les tableaux 2.4 et 2.5 et ont été obtenus sur un PC avec un processeur 2.66GHz Pentium Xeon.

Pour les différentes formulations pseudo-polynômiales, seules les résultats pour les formulations $F^{\Psi iab}$, $F_{renf}^{\Psi iab}$ et $F_{renf}^{Xia\gamma ib}$ sont reportées. En effet, pour les instances « ngcut », les bornes des relaxations linéaires ou des relaxations avec δ entier pour les formulations F_{renf}^{zab} , F_{renf}^{ziab} et $F_{renf}^{xia\gamma ib}$ sont identiques aux résultats obtenues pour $F_{renf}^{Xia\gamma ib}$. Seul le temps de calcul est plus important. Par exemple, pour les relaxations avec δ entier sur les instances « ngcut », le temps moyen de calcul est de 79 secondes pour la formulation F_{renf}^{zab} , de 103,24 secondes pour F_{renf}^{ziab} et 33,84 secondes pour $F_{renf}^{xia\gamma ib}$.

Le tableau 2.4 montre que les relaxations linéaires et les relaxations considérant les variables δ comme entier des différentes formulations pseudo-polynômiales offrent des qualités de bornes équivalentes à certaines bornes de la littérature (par exemple $U_{F_{BiL}^1}$). Nous remarquons aussi que les formulations renforcées peuvent améliorer légèrement les bornes obtenues. Les temps de résolution pour les formulations $F^{\Psi iab}$ et $F_{renf}^{\Psi iab}$ sont cependant beaucoup plus important. Les temps de calcul est trop important pour pouvoir tester cette

formulation sur les instances « gcut », « cgcut », « wang » et « okp ». Ceci est dû à l'explosion du nombre de variables causée par des tailles de conteneur plus importantes. Dans le tableau 2.5, les formulations $F^{Xia\gamma ib}_{renf}$ et $F^{Xia\gamma ib}_{renf}$ permettent d'obtenir des bornes duales de qualités équivalentes que $U_{F_{BiL}^1}^\delta$ et meilleurs que $U_{F_{DF}^\delta}^\delta$. Les temps de calcul restent aussi compétitifs comparés à $U_{F_{BiL}^1}^\delta$ et $U_{F_{BiL}^2}^\delta$. Cependant, certaines instances restent trop difficiles pour ces bornes.

Inst.	OPT	Borne duale par relaxation linéaire						
		$U_{F^{\Psi iab}}^{PL}$	$U_{F^{\Psi iab}_{renf}}^{PL}$	$U_{F^{Xia\gamma ib}_{renf}}^{PL}$	$U_{F_{BiL}^1}^{PL\ a}$	$U_{FE_{relax}^{\lambda_v}}^{PL}$	$U_{FE_{relax}^{\lambda_v\mu_h}}^{PL}$	$U_{FE_{enr}^{\lambda_v\mu_h}}^{PL}$
ngcut1	164	205	205	206	206	189	186	186
ngcut2	230	257	257	257	257	257	245	245
ngcut3	247	255	255	260	260	266	257	256
ngcut4	268	271	271	271	271	281	270	270
ngcut5	358	359	359	359	359	383	359	359
ngcut6	289	317	316	316	317	304	302	302
ngcut8	834	906	906	906	906	906	906	905
ngcut9	924	934	934	945	945	961	941	939
ngcut10	1452	1501	1484	1510	1511	1510	1504	1504
ngcut11	1688	1776	1776	1779	1779	1872	1752	1751
ngcut12	1865	1945	1945	1954	1953	1962	1939	1939
gap moy.	-	0.0666	0.0659	0.0713	0.0716	0.0767	0,0467	0,04

(a) Etude des relaxations linéaires pour les différentes formulations

Inst.	OPT	Borne duale avec δ_i entier						
		$U_{F^{\Psi iab}_{renf}}^\delta$	$U_{F^{Xia\gamma ib}}^\delta$	$U_{F^{Xia\gamma ib}_{renf}}^\delta$	$U_{F_{BiL}^1}^\delta\ a$	$U_{F_{BiL}^2}^\delta\ a$	$U_{FE_{relax}^{\lambda_v\mu_h}}^\delta$	$U_{FE_{enr}^{\lambda_v\mu_h}}^\delta$
ngcut1	164	201	201	201	201	164	164	164
ngcut2	230	253	253	253	253	230	230	230
ngcut3	247	254	254	254	254	249	249	249
ngcut4	268	268	268	268	268	268	268	268
ngcut5	358	358	358	358	358	358	358	358
ngcut6	289	298	317	314	317	289	289	289
ngcut8	834	906	906	906	906	856	862	862
ngcut9	924	924	924	924	924	924	924	924
ngcut10	1452	1452	1452	1452	1452	1452	1452	1452
ngcut11	1688	1726	1760	1760	1726	1726	1688	1688
ngcut12	1865	1913	1913	1913	1913	1865	1865	1865
gap moy.	-	0.0472	0.055	0.0541	0.0532	0.0052	0.0038	0.0038
tps moy.	-	69.25	0.06	0.08	0.04	0.1	3.48	31.05

(b) Etude des bornes obtenues avec δ_i entier avec pour les différentes formulations

a. résultats obtenus par Caprara et Monaci

TABLEAU 2.4: Etude des bornes duales pour les différentes modélisations

Concernant les formulations compactes $F^{c_{ij}d_{ij}}$ et $F^{x_{ia}y_{ib}}_{relax}$, elles n'offrent que des relaxations de très faibles qualités puisque, pour chaque instance « ngcut », la sélection de tous

les objets ($\delta_i = 1$ pour tout $i \in \mathcal{I}$) est une solution valide aux problèmes relâchés. Donc, elles ne sont pas reportées dans les tableaux 2.4a et 2.4b car elles ne représentent pas un intérêt en terme de relaxation. La nouvelle formulation compacte proposé $F^{w^k h^l}$ arrive au même constat.

Inst.	OPT	Bornes duales					
		$U_{F^{Xia\gamma ib}}^\delta$	$U_{F_{renf}^{Xia\gamma ib}}^\delta$	$U_{F_{DFP}}^\delta$	$U_{F_{BiL}^1}^\delta$ ^a	$U_{F_{BiL}^2}^\delta$ ^a	$U_{FE_{relax}^{\lambda_v \mu_h}}^\delta$
cgcut1	244	244	244	246	244	244	244
cgcut2	2892	2903	2903	2915	2903	2900	2903
cgcut3	1860	1940	1940	1940	1940	1900	1920
wang20	2726	2756	2756	2762	2752	2752	2735
gap moy.	-	0,0144	0.0144	0.018	0.014	0.009	0.01
tps moy.	-	0.51	16.5	2.14	5.63	5.69	174.09
gcut1	48368	50057	48368	48440	48440	48368	48368
gcut2	59798	60662	60562	61420	60662	59798	59798
gcut3	61275	61275	61275	61384	61275	61275	61275
gcut4	61380	61946	61821	62496	≤ 62497		≤ 62482
gcut5	195582	230446	215322	230446	223327	195582	215322
gcut6	236305	237682	236305	241018	237682	236305	236305
gcut7	240143	240143	240143	246057	240143	240143	240143
gcut8	245758	*	246059	249791	≤ 249964		≤ 249747
gcut9	939600	939600	939600	939600	939600	939600	939600
gcut10	937349	952192	947385	960716	952192	937349	937349
gcut11	969709	974427	974427	986854	974427	969709	969709
gcut12	979521	≤ 997761	979521	993466	≤ 993124		≤ 997102
gcut13	≥ 8622498	≤ 8918301	*	8999510	≤ 8917895		≤ 8905395
gap moy. ^b	-	0.0282	0.0144	0.0329	0.0205	0	0.01
tps moy. ^b	-	247.07	15.94	4.73	139.89	146.79	89.6
okp1	27718	29507	29507	29344	28431	27796	≤ 28717
okp2	22502	24738	24738	24800	24738	23696	≤ 24319
okp3	24019	25884	25884	25884	25884	24213	≤ 26120
okp4	32893	32915	32915	32950	32893	32893	≤ 33239
okp5	27923	27963	*	28201	27934	27923	≤ 28544
gap moy.	-	0.0956	*	0.0966	0.0858	0.0569	*
tps moy.	-	8.74	*	1.05	4.5	239	*

a. résultats obtenus par Caprara et Monaci

b. temps moyen calculé sur 9 instances résolus par Caprara et Monaci

TABLEAU 2.5: Etude des bornes obtenues avec δ_i entier avec pour les différentes formulations

Pour les formulations étendues, seuls les résultats des formulation $FE_{relax}^{\lambda_v \mu_h}$ et $FE_{enr}^{\lambda_v \mu_h}$ sont reportés. Les bornes obtenues par les formulations $FE_{var,relax}^{\lambda_v \mu_h}$ et $FE_{var}^{\lambda_v \mu_h}$ sont identiques à la formulation $FE_{relax}^{\lambda_v \mu_h}$. Cependant, les temps de résolution sont plus important pour $FE_{var}^{\lambda_v \mu_h}$ et $FE_{var,relax}^{\lambda_v \mu_h}$. Ceci est causé par des sous-problèmes plus difficiles à résoudre.

D'après les tableaux 2.4a et 2.4b, les bornes duales obtenues par les formulations $FE_{relax}^{\lambda_v\mu_h}$ et $FE_{enr}^{\lambda_v\mu_h}$ sont de très bonne qualité, comparable à $U_{F_{BiL}^2}^\delta$. Nous remarquons que la borne duale donnée par la formulation $FE_{enr}^{\lambda_v\mu_h}$ n'améliore pas les résultats des bornes données par la formulation $FE_{relax}^{\lambda_v\mu_h}$. Seul un branchement sur les variables x_{ij} et y_{ij} permettrait une légère amélioration des qualités de bornes car la formulation $FE_{enr}^{\lambda_v\mu_h}$ propose une modélisation partielle de la contrainte de non-intersection des objets. Ces résultats se confirment dans le tableau 2.5. Cependant, les temps de calcul sont plus importants pour les formulations étendues comparé au calcul de la borne $U_{F_{BiL}^2}^\delta$. En particulier, quand le nombre d'objets est trop important, la résolution n'est pas possible en 1800 secondes.

Ainsi, grâce à cette étude, nous remarquons que la formulation relâchée $F_{renf}^{\chi_{ia}\gamma_{ib}}$ offre d'aussi bonnes bornes que les autres formulations. Cette formulation est obtenue en relâchant la contrainte de non-intersection des objets. Une meilleure modélisation de ces contraintes est un des principaux enjeux pour espérer améliorer les bornes duales.

De même, pour les formulations étendues $FE_{enr}^{\lambda_v\mu_h}$ et $FE_{var}^{\lambda_v\mu_h}$, les contraintes de non-intersection des objets ne sont que partiellement modélisées. Un branchement sur les variables x_{ij} et y_{ij} n'améliorent que légèrement les bornes obtenues. Il est donc primordial d'améliorer la modélisation de cette contrainte. Le chapitre suivant présente un branchement pour implémenter cette contrainte de non-intersection des objets pour les formulations étendues.

*
* *

Sommaire

3.1	Couplage des bandes horizontales et verticales	88
3.2	Résultats numériques	99

CHAPITRE III

Implémentation de contraintes disjonctives par voie de branchement

D'après la section 2.4, nous savons que les formulations basées sur des approches de décomposition (génération de colonnes) offrent des bornes duales de bonne qualité. Néanmoins, elles ne formulent qu'une relaxation du problème où les contraintes disjonctives interdisant la contrainte de non-intersection des objets et de consécutivité des bandes sont ignorées. Afin de renforcer ces bornes duales, nous étudions dans ce chapitre l'implémentation de ces contraintes disjonctives par voie d'une procédure de branchement qui soit compatible avec la génération de colonnes.

En 2009, Belov et Rohling [9] ont proposé un branchement sur les variables x_{ij} et y_{ij} basée sur la formulation étendue $FE_{enr}^{\lambda_v \mu_h}$ associée à l'approche de Fekete et Schepers. Nous proposons un branchement différent pour améliorer les résultats obtenus par les formulations étendues en modélisant une partie des propriétés manquantes. En 2010, Vanderbeck [36] propose un branchement générique dans le cadre d'approches par génération de colonnes. Ce branchement a la particularité de ne pas modifier la structure du sous-problème et est à l'origine du branchement que nous présentons par la suite.

Ce chapitre est donc consacré à la mise en place de branchements spécifiques pour les approches de décomposition qui implémentent des « coupes de réalisabilité » de type contraintes disjonctives. Après avoir résolu la relaxation linéaire de la formulation issue de l'approche de décomposition par une méthode de génération de colonnes, nous vérifions la contrainte de non-intersection des objets. Si une contrainte est violée, nous implémentons un branchement. Nous prendrons comme décomposition de référence celle définie par Scheithauer (Formulation étendue $FE_{relax}^{\lambda_v \mu_h}$) car elle s'avère la plus abordable en terme de temps de calcul sans perte de qualité de solution et le sous-problème associée est le plus simple à résoudre. Cependant, nous pouvons utiliser ces idées pour les autres approches.

3.1 Couplage des bandes horizontales et verticales

Avant d'appliquer une procédure de séparation par ajout de coupes de réalisabilité, nous projetons la solution (λ, μ) dans l'espace des variables (x, y) . La figure 3.1 montre un exemple de solution irréalizable trouvée par l'algorithme de génération de colonnes sur la formulation $FE_{relax}^{\lambda_v \mu_h}$.

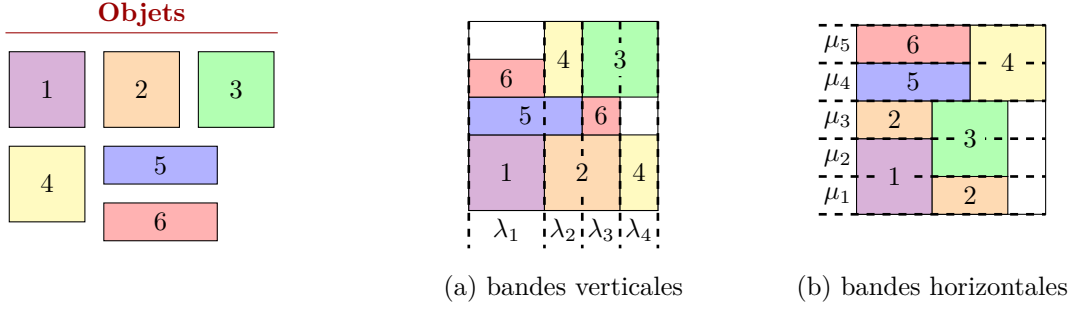


FIGURE 3.1: Solution invalide trouvée pour la formulation étendue $FE_{relax}^{\lambda_v \mu_h}$

Pour pouvoir vérifier que les solutions obtenues sont réalisables, nous montrons que les propriétés décrites par Fekete et Schepers (cf. section 1.4) sont vérifiées par la solution (x, y) obtenue en projetant (λ, μ) . Dans le cas contraire, nous branchons pour renforcer la formulation. La borne duale après évaluation des nœuds résultant du branchement en sera améliorée.

Grâce aux bandes générées par les approches de décomposition, nous construisons des graphes où une arête entre i et j est créée si une bande contenant ces deux objets est sélectionnée. À partir des graphes ainsi construits, nous démontrons facilement que la propriété de non dépassement de la boîte donnée par Fekete et Schepers (condition (1.7b)) est alors forcément vérifiée pour toute solution obtenue par l'une des décompositions développées en section 2.4. Il suffit de remarquer que :

$$\alpha(G_x, \vec{w}) \leq \sum_{v \in \mathcal{V}} \lambda_v \leq \mathbf{W},$$

$$\alpha(G_y, \vec{h}) \leq \sum_{h \in \mathcal{H}} \mu_h \leq \mathbf{H}.$$

En effet, par la contrainte (2.29b) et en constatant que, pour tout ensemble stable S ,

$$\sum_{\substack{v \in \mathcal{V}: \\ \exists i \in S \text{ avec } x_i^v = 1}} \lambda_v \leq \sum_{v \in \mathcal{V}} \lambda_v,$$

nous démontrons la première inégalité. De même, pour la deuxième inégalité.

En revanche, les contraintes de consécutivité des bandes (condition (1.7a)) et de non-intersection des objets (condition (1.7c)) ne sont pas forcément vérifiées. Pour renforcer les solutions, nous allons donc définir des coupes disjonctives pour vérifier la contrainte de non-intersection des objets.

3.1.1 Propriété de non intersection des objets

La contrainte de non intersection des objets consiste à vérifier que pour toute paire d'objets, il existe au moins une dimension où ils ne se chevauchent pas. Sur les graphes associés à chacune des dimensions, cela consiste à vérifier que pour toute paire de nœuds i et j , l'arête (i, j) n'est présente que sur une dimension au maximum. Si nous utilisons les variables indiquant si deux objets s'intersectent selon la verticale ou sur l'horizontale (variables x_{ij} et y_{ij} définies dans la section 1.1), alors il suffit d'ajouter la contrainte suivante dans notre décomposition :

$$x_{ij}y_{ij} = 0, \quad \forall i, j \in \mathcal{I}(i < j). \quad (3.1)$$

La modélisation d'une telle contrainte engendre des problèmes quadratiques. En général, cette contrainte est modélisée par la contrainte linéaire $x_{ij} + y_{ij} \leq 1, \forall i, j \in \mathcal{I}(i < j)$. L'implémentation d'une telle contrainte n'est pas évidente. En effet, il faut pouvoir lier les variables x_{ij}, y_{ij} aux colonnes. L'ajout de contraintes complexifie les sous-problèmes.

Par exemple, la contrainte de non-intersection des objets est partiellement implémentée par les contraintes (2.29d)-(2.29f) dans la formulation $FE^{\lambda_v \mu_h}$ et les contraintes (2.31d) et (2.32c)-(2.32g) dans la formulation $FE_{enr}^{\lambda_v \mu_h}$. Cette dernière formulation est celle employée par Belov et Rohling, un travail fait en parallèle de notre étude. Pour la formulation $FE_{enr}^{\lambda_v \mu_h}$, le sous-problème est plus complexe car il s'agit de résoudre des problèmes de sac-à-dos avec conflit. Après plusieurs branchements, le sous-problème est alors le suivant :

FORMULATION 21: Sous-problèmes après branchements $[\mathcal{V}_{br} \& \mathcal{H}_{br}]$

$$[\mathcal{V}_{br}] \quad \max \sum_{i \in \mathcal{I}} \alpha_i x_i - \gamma, \quad (3.2a)$$

$$\text{s. t. } \sum_{i \in \mathcal{I}} \mathbf{h}_i x_i \leq \mathbf{H}, \quad (3.2b)$$

$$x_i + x_j \leq 1, \quad \forall i, j \in \mathcal{I}(i < j) \text{ tel que } x_{ij} = 0, \quad (3.2c)$$

$$x_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}, \quad (3.2d)$$

$$[\mathcal{H}_{br}] \quad \max \sum_{i \in \mathcal{I}} \beta_i y_i - \sigma, \quad (3.2e)$$

$$\text{s. t. } \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i \leq \mathbf{W}, \quad (3.2f)$$

$$y_i + y_j \leq 1, \quad \forall i, j \in \mathcal{I}(i < j) \text{ tel que } y_{ij} = 0, \quad (3.2g)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}. \quad (3.2h)$$

Nous avons réfléchi à d'autres solutions pour définir les bonnes valeurs associées à ces variables en ne modifiant pas la complexité des sous-problèmes. Ainsi, nous implémentons un branchement ajoutant une contrainte disjonctive $x_{ij} = 0$ ou $y_{ij} = 0$ et en scindant les sous-problèmes pour ne générer que des bandes vérifiant les branchements.

3.1.1.1 Stratégie 1 : Vérification de la contrainte non-intersection pour l'ensemble des bandes

La première stratégie consiste à définir un branchement évitant qu'une paire d'objets ne se chevauchent sur les deux dimensions pour l'ensemble des bandes dans le placement.

Le schéma de branchement

Nous allons donc nous assurer que, pour chaque paire d'objets $(i, j) \in \mathcal{I}^2$ avec $i < j$, l'une de ces deux conditions est vérifiée :

1. aucune bande sélectionnée ne contient la paire (i, j) selon la verticale (forcer $x_{ij} = 0$),
2. aucune bande sélectionnée ne contient la paire (i, j) selon l'horizontale (forcer $y_{ij} = 0$).

Dans le cas contraire, c'est-à-dire, si à un rangement de bandes donné, il existe une paire d'objets (i, j) ayant au moins une bande commune selon la verticale et selon l'horizontale ($x_{ij} = 1$ et $y_{ij} = 1$) ; alors, la propriété de non intersection des objets n'est pas vérifiée. Nous allons donc ajouter une contrainte disjonctive consistant à fixer l'une des variables x_{ij} ou y_{ij} à zéro :

$$\left\{x_{ij} = 0\right\} \vee \left\{y_{ij} = 0\right\} \quad (3.3)$$

Pour lier ce branchement aux colonnes générées par la procédure de génération de colonnes sans ajouter de contraintes, une séparation des sous-problèmes est nécessaire. Au préalable, nous supprimons les colonnes violant la contrainte. Nous présentons ci-dessous la procédure du schéma de branchement :

Procédure de branchement

- ① Choisir une paire d'objets $(i, j) \in \mathcal{I}^2 (i < j)$ violant la contrainte (3.1).
 - ② S'il en existe,
 - ★ Brancher selon la contrainte disjonctive $\left\{x_{ij} = 0\right\} \vee \left\{y_{ij} = 0\right\}$,
 - ★ Supprimer les colonnes violant la contrainte de branchement,
 - ★ Séparer les sous-problèmes à résoudre en fonction des branchements,
 - ★ Ré-optimiser en générant de nouvelles bandes :
 - Si la solution est réalisable, recommencer la procédure de branchement,
 - Si la solution est irréalisable, remonter au nœud père.
- Sinon, sauvegarder la solution et remonter au nœud père.

Nous appliquons ce schéma sur un exemple. Considérons l'ensemble d'objets défini par l'exemple 3.2. Supposons qu'au nœud racine, nous obtenons la solution définie par l'exemple 3.2. Nous remarquons que les objets 1 et 2 partagent une abscisse et une ordonnée commune. Nous branchons alors sur le couple $(1, 2)$. Alors, nous obtenons de nouvelles solutions. En appliquant la procédure de branchement successivement sur les nœuds de l'arbre de « Branch-and-Bound », nous obtenons un arbre comme défini sur l'exemple 3.2.

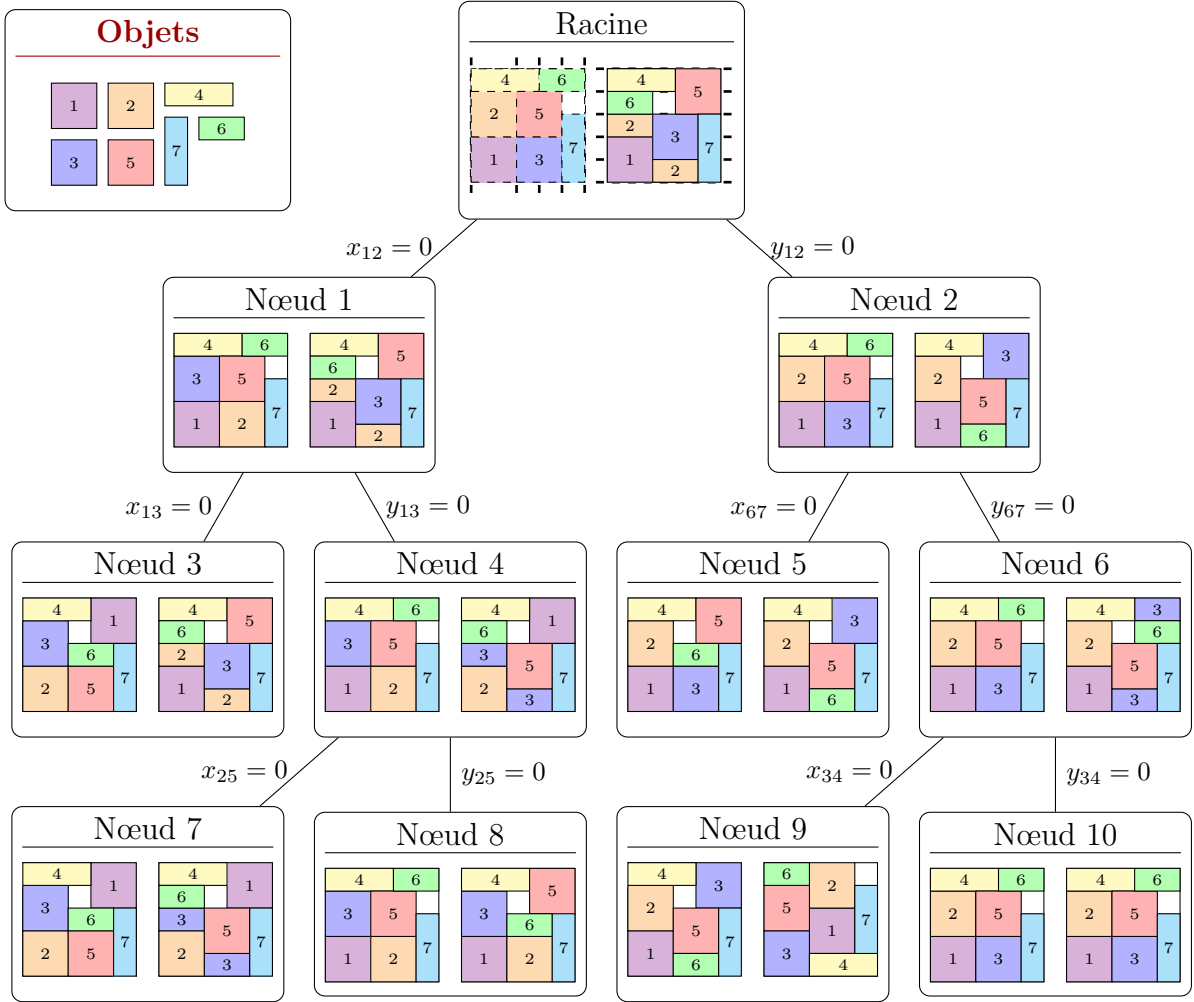


FIGURE 3.2: Exemple d'arbre de Branch-and-Price appliqué sur un exemple en fonction des différents branchements

D'après cette procédure, nous devons choisir une paire parmi les paires violant la contrainte (3.1). De plus, après chaque branchement, nous devons modifier les sous-problèmes afin de ne pas générer de bandes ne respectant pas la contrainte de branchement. En effet, à chaque variable x_{ij} (resp. y_{ij}) fixée à zéro, nous devons garantir de ne pas générer de bandes verticales (resp. horizontales) contenant les objets i et j . Donc, nous allons diviser les différents sous-problèmes en fonction des différents cas pour ne pas changer la complexité du sous-problème original.

Choix de la disjonction de branchement

D'après la procédure définie précédemment, nous devons sélectionner une paire d'objets violant la contrainte (3.1). Afin d'obtenir la paire d'objets la plus intéressante, nous essayons de comparer les différentes paires d'objets et détecter une paire violant le plus la contrainte selon un critère choisis au préalable.

Pour chaque paire d'objet $(i, j) \in \mathcal{I}^2$ avec $i < j$, soit o_{ij} un facteur représentant la violation de la contrainte (3.1) selon un critère prédéfini. Ce facteur définira alors la priorité

de branchement. Soit Ω l'ensemble des paires d'objets (i, j) avec la plus grande valeur strictement positive o_{ij} :

$$\Omega = \left\{ (i, j) \in \mathcal{I}^2 (i < j) : (i, j) = \underset{\substack{(i', j') \in \mathcal{I}^2 (i' < j') \\ o_{i'j'} > 0}}{\operatorname{argmax}} o_{i'j'} \right\}$$

Ω définira alors l'ensemble des paires les plus intéressantes. En fonction de la définition du facteur o_{ij} , l'ensemble Ω peut être différent.

Remarque Par convention, $\forall (i, j) \in \mathcal{I}^2$ avec $i < j$, nous posons $o_{ij} = 0$ si et seulement si la contrainte (3.1) n'est pas violée pour la paire (i, j) . $\blacklozenge$

Ainsi, nous définissons la procédure suivante pour choisir une paire d'objets $(i, j) \in \mathcal{I}^2$ avec $i < j$ violant la propriété de non-intersection des objets :

Procédure de choix d'une disjonction pour le branchement

- ① Calculer o_{ij} selon un critère fixé pour tout $(i, j) \in \mathcal{I}^2$ avec $i < j$.
- ② Si $\Omega = \emptyset$, alors arrêt du branchement.
Sinon, choisir une paire d'objets (i, j) dans Ω , puis brancher sur $x_{ij}y_{ij} = 0$.

Nous définissons donc des critères pour déterminer une priorité entre les couples d'objets violant la contrainte (3.1). Nous savons que :

$$\sum_{v \in \mathcal{V}} x_i^v x_j^v \lambda_v \leq \mathbf{W} x_{ij}, \quad \forall i, j \in \mathcal{I} (i < j), \quad (3.4a)$$

$$\sum_{h \in \mathcal{H}} y_i^h y_j^h \mu_h \leq \mathbf{H} y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j). \quad (3.4b)$$

A partir de ces contraintes, nous en déduisons différents critères de choix en définissant une valeur pour o_{ij} . En effet, en multipliant chaque membre des contraintes (3.4a)-(3.4b) entre eux, Nous obtenons une nouvelle contrainte :

$$\frac{1}{\mathbf{WH}} \left(\sum_{v \in \mathcal{V}} x_i^v x_j^v \lambda_v \right) \left(\sum_{h \in \mathcal{H}} y_i^h y_j^h \mu_h \right) \leq x_{ij} y_{ij}, \quad \forall i, j \in \mathcal{I} (i < j). \quad (3.5)$$

Alors, nous définissons la paire (i, j) avec la plus grande valeur $x_{ij}y_{ij}$ en considérant la relaxation linéaire. Nous définissons donc un critère de présence d'un couple d'objets sur les deux dimensions selon les largeurs des bandes contenant ces objets. Nous fixons ainsi pour $o_{i,j}$ la valeur suivante :

$$o_{ij} = \left(\sum_{v \in \mathcal{V}} x_i^v x_j^v \lambda_v \right) \left(\sum_{h \in \mathcal{H}} y_i^h y_j^h \mu_h \right), \quad \forall i, j \in \mathcal{I} (i < j). \quad (3.6)$$

Un autre exemple consiste à définir un critère de présence d'un couple d'objets sur les deux dimensions selon le nombre de colonnes contenant ce couple. Nous obtenons alors le critère suivant :

$$o_{ij} = \left(\sum_{v \in \mathcal{V}} \left\lceil \frac{1}{\mathbf{W}} x_i^v x_j^v \lambda_v \right\rceil \right) \left(\sum_{h \in \mathcal{H}} \left\lceil \frac{1}{\mathbf{H}} y_i^h y_j^h \mu_h \right\rceil \right), \quad \forall i, j \in \mathcal{I} (i < j). \quad (3.7)$$

La figure 3.3 représente les choix de disjonction faits selon le critère (3.7) pour l'exemple de la figure 3.2.

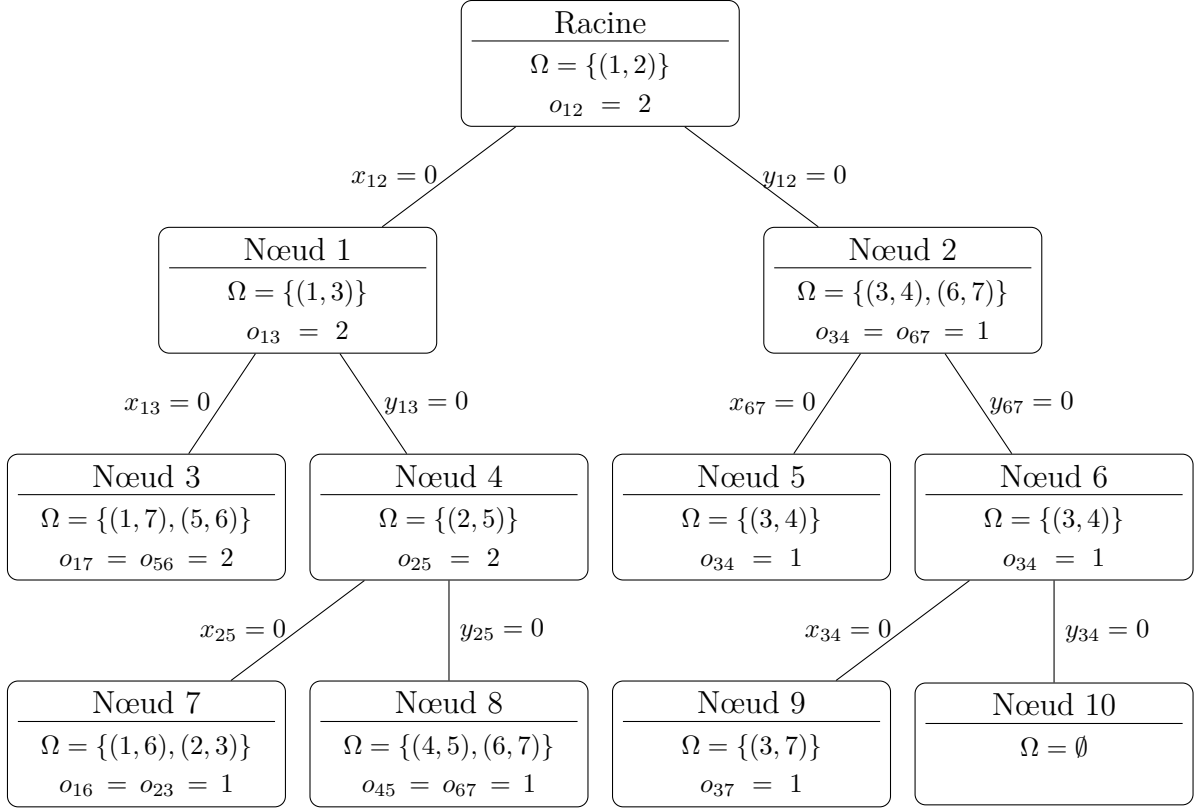


FIGURE 3.3: Différents choix possibles de branchement selon le critère donné par l'équation (3.7) pour chaque nœud de l'exemple de la figure 3.2

Séparation des sous-problèmes

Selon la procédure précédemment définie, le branchement va restreindre l'ensemble des colonnes valides. Au nœud racine, les sous-problèmes restent $\mathcal{V}$ et $\mathcal{H}$ (définis en section 2.4). Pour les autres nœuds, la définition des sous-problèmes changent en fonction du branchement. Pour éviter de complexifier les sous-problèmes, nous les séparons en plusieurs sous-problèmes indépendants tels que chaque problème caractérise une partie de l'ensemble des colonnes admissibles.

Par exemple, supposons que nous branchons sur la paire d'objets $(i, j) \in \mathcal{I}^2$ avec $i < j$ au nœud racine. L'une des branches consiste alors à fixer $x_{ij} = 0$. Alors, nous devons seulement générer des bandes verticales ne contenant pas les objets i et j simultanément. Il existe deux types de bandes verticales à générer pour vérifier cette contrainte :

- Soit les bandes ne contiennent pas l'objet i ,
- Soit les bandes contiennent l'objet i et ne contiennent pas l'objet j .

Dans ce cas, nous séparons les sous-problèmes afin de traiter ces deux cas séparément. Pour un branchement sur la variable $x_{ij} = 0$ à partir du nœud racine, nous séparons le sous-problème $\mathcal{V}$ en deux sous-problèmes indépendants : d'une part, nous générons un sous-problème interdisant la présence de l'objet i (noté $\mathcal{V}^{\bar{i}}$) ; d'autre part, nous générons un sous-problème imposant la présence de l'objet i et interdisant celle de l'objet j (noté $\mathcal{V}^{i,\bar{j}}$) :

FORMULATION 22: Sous-problèmes $[\mathcal{V}^{\bar{i}} \text{ \& } \mathcal{V}^{i,\bar{j}}]$

$$\mathcal{V}^{\bar{i}} : \quad \max \sum_{i' \in \mathcal{I}} \alpha_{i'} x_{i'} - \gamma, \quad (3.8a)$$

$$\text{s. t.} \quad \sum_{i' \in \mathcal{I}} \mathbf{h}_{i'} x_{i'} \leq \mathbf{H}, \quad (3.8b)$$

$$x_i = 0, \quad (3.8c)$$

$$x_{i'} \in \{0, 1\}, \quad \forall i' \in \mathcal{I}, \quad (3.8d)$$

$$\mathcal{V}^{i,\bar{j}} : \quad \max \sum_{i' \in \mathcal{I}} \alpha_{i'} x_{i'} - \gamma, \quad (3.8e)$$

$$\text{s. t.} \quad \sum_{i' \in \mathcal{I}} \mathbf{h}_{i'} x_{i'} \leq \mathbf{H}, \quad (3.8f)$$

$$x_i = 1, x_j = 0, \quad (3.8g)$$

$$x_{i'} \in \{0, 1\}, \quad \forall i' \in \mathcal{I}. \quad (3.8h)$$

En résolvant les sous-problèmes ci-dessus, nous générons que des bandes vérifiant la condition de branchement. Pour un nœud quelconque de l'arbre, nous devons séparer les sous-problèmes en fonction des branchements précédemment définis. Ainsi, après plusieurs itérations, de nombreux objets sont alors fixés et les sous-problèmes à résoudre se multiplient. Il faut donc définir une procédure afin de modifier et/ou ajouter de nouveaux sous-problèmes en fonction des différents cas possibles. Par exemple, pour un sous-problème donné, si l'objet i est déjà interdit dans celui-ci, il est inutile de séparer le sous-problème. Inversement, si l'objet i est obligatoirement sélectionné, il faut s'assurer que l'objet j est obligatoirement interdit. Pour caractériser les différents cas possibles, nous introduisons une nouvelle notation pour les sous-problèmes avec des objets fixés et des objets interdits. Soit $\mathcal{V}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}}$ l'ensemble des bandes verticales contenant les objets $i_1, \dots, i_{m_i}$ et interdisant les objets $j_1, \dots, j_{m_j}$ (où m_i représente le nombre d'objets obligatoires et m_j le nombre d'objets interdits). Le sous-problème associé est alors défini de cette manière :

FORMULATION 23: Sous-problème $[\mathcal{V}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}}]$

$$\mathcal{V}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}} : \quad \max \sum_{i \in \mathcal{I}} \alpha_i x_i - \gamma, \quad (3.9a)$$

$$\text{s. t.} \quad \sum_{i \in \mathcal{I}} \mathbf{h}_i x_i \leq \mathbf{H}, \quad (3.9b)$$

$$x_i = 1, \quad \forall i \in \{i_1, \dots, i_{m_i}\}, \quad (3.9c)$$

$$x_j = 0, \quad \forall j \in \{j_1, \dots, j_{m_j}\}, \quad (3.9d)$$

$$x_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}. \quad (3.9e)$$

De la même manière, soit $\mathcal{H}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}}$ l'ensemble des bandes horizontales où les ob-

jets $i_1, \dots, i_{m_i}$ sont obligatoires et les objets $j_1, \dots, j_{m_j}$ sont interdits (avec m_i le nombre d'objets obligatoires et m_j le nombre d'objets interdits). Le sous-problème est le suivant :

FORMULATION 24: Sous-problème $[\mathcal{H}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}}]$

$$\mathcal{H}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}} : \max \sum_{i \in \mathcal{I}} \beta_i y_i - \sigma, \quad (3.10a)$$

$$\text{s. t. } \sum_{i \in \mathcal{I}} \mathbf{w}_i y_i \leq \mathbf{W}, \quad (3.10b)$$

$$y_i = 1, \quad \forall i \in \{i_1, \dots, i_{m_i}\}, \quad (3.10c)$$

$$y_j = 0, \quad \forall j \in \{j_1, \dots, j_{m_j}\}, \quad (3.10d)$$

$$y_i \in \{0, 1\}, \quad \forall i \in \mathcal{I}. \quad (3.10e)$$

Supposons que nous branchons sur la variable $x_{ij} = 0$. Les différentes situations rencontrées pour séparer le sous-problème $\mathcal{V}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}}$ sont les suivantes :

1. Soit l'objet i et l'objet j appartiennent à l'ensemble $\{i_1, \dots, i_{m_i}\}$. Dans ce cas, le sous-problème peut être supprimé car le sous-problème impose une condition incompatible avec le branchement $x_{ij} = 0$.
2. Soit l'objet i et/ou l'objet j appartiennent à l'ensemble $\{j_1, \dots, j_{m_j}\}$. Dans ce cas, le sous-problème ne change pas car l'un des deux objets est déjà interdit. Donc, aucune bande contenant les deux objets ne peut être générée.
3. Soit l'objet i ou l'objet j appartient à $\{i_1, \dots, i_{m_i}\}$. Dans ce cas, l'un des objets est obligatoirement présent dans le sous-problème. Il faut garantir que la bande ne contient pas les deux objets en interdisant à l'autre objet d'être présent.
4. Soient les objets i et j ne sont contenus dans aucun des ensembles. Dans ce cas, nous générons deux nouveaux sous-problèmes remplaçant le sous-problème considéré pour garantir la condition de branchement.

A partir de ces différents cas, nous générons ou non de nouveaux sous-problèmes pour respecter la condition de branchement dans les sous-problèmes. La procédure suivante permet donc de séparer nos sous-problèmes selon un branchement sur la variable $x_{ij} = 0$:

Procédure de séparation des sous-problèmes

- ① Séparer chaque sous-problème du type $\mathcal{V}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}}$:
 - ★ Si i et $j \in \{i_1, \dots, i_{m_i}\}$ alors supprimer sous-problème,
 - ★ Si i ou $j \in \{j_1, \dots, j_{m_j}\}$ alors conserver sous-problème,
 - ★ Si $i \in \{i_1, \dots, i_{m_i}\}$ alors transformer sous-problème en $\mathcal{V}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}, \bar{j}}$,
 - ★ Si $j \in \{i_1, \dots, i_{m_i}\}$ alors transformer sous-problème en $\mathcal{V}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}, \bar{i}}$,
 - ★ Sinon remplacer sous-problème par $\mathcal{V}^{i_1, \dots, i_{m_i}, \bar{j}_1, \dots, \bar{j}_{m_j}, \bar{i}}$ et $\mathcal{V}^{i_1, \dots, i_{m_i}, \bar{i}, \bar{j}_1, \dots, \bar{j}_{m_j}, \bar{j}}$.
- ② Résoudre chaque sous-problème et générer de nouvelles bandes.

Nous définissons de manière analogue une procédure de séparation des sous-problèmes selon la variable $y_{ij} = 0$.

En appliquant cette procédure sur la figure de l'exemple 3.2, nous obtenons l'arbre des sous-problèmes présenté par la figure 3.4.

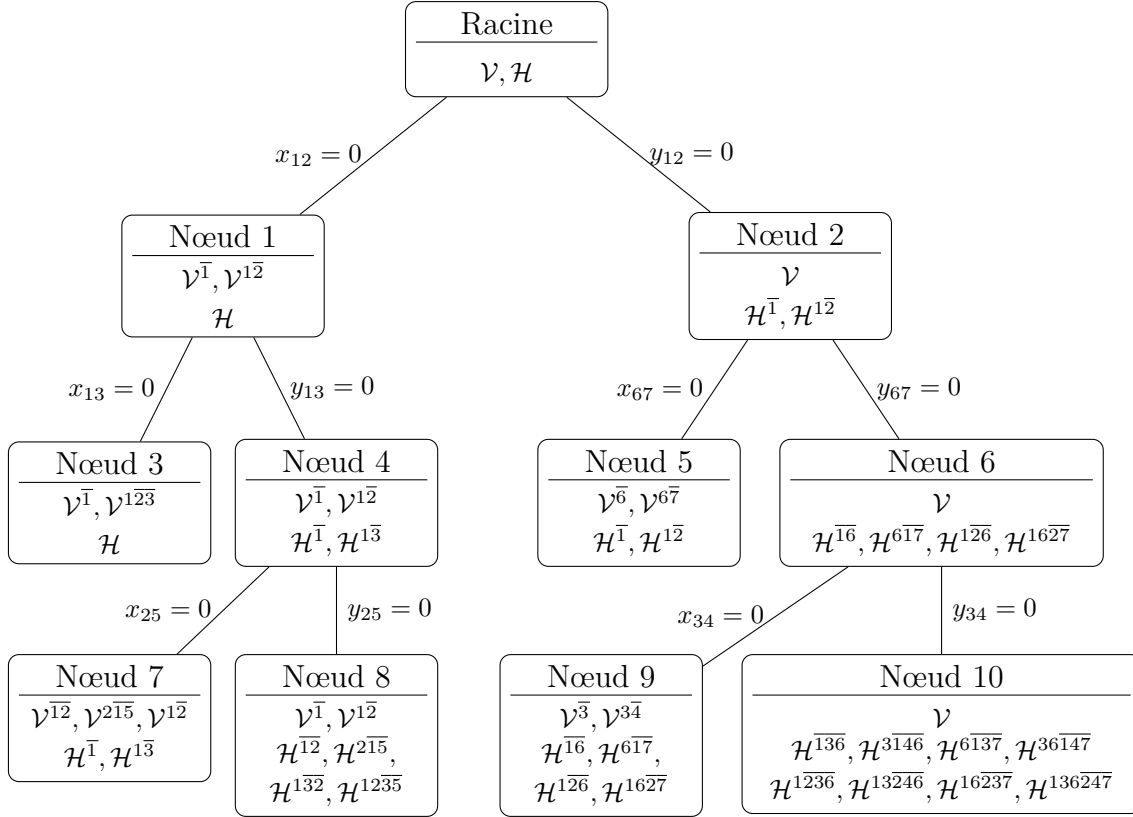


FIGURE 3.4: Différents sous-problèmes à résoudre après branchement pour la figure de l'exemple 3.2

Lemme 3.1

Le branchement selon la contrainte de non-intersection des objets est valide.

Preuve. Evident par construction. □

D'après la figure 3.4, nous remarquons qu'à certain nœud de l'arbre un nombre assez important de sous-problèmes est résolu. En effet, le nombre de sous-problèmes peut être potentiellement doublé à chaque nœud de l'arbre : dans le pire des cas, ce nombre est de $\Theta(2^l)$ où l est la profondeur du nœud dans l'arbre de « Branch-and-Bound ». Remarquons néanmoins qu'il est possible de développer une stratégie de pricing qui évite d'explorer tous les sous-problèmes à chaque itération par l'utilisation d'arbre de sous-problèmes. Vanderbeck [36] utilise de tels arbres pour diminuer le nombre de sous-problèmes à résoudre. De plus, cette stratégie peut engendrer un déséquilibre entre les sous-problèmes. En effet, alors que certains problèmes ont de nombreux objets fixés, d'autres ont seulement des objets interdits. Pour éviter une augmentation trop rapide du nombre de sous-problèmes, il faut développer une autre approche.

3.1.1.2 Stratégie 2 : Vérification de la contrainte non-intersection pour un sous-ensemble de bandes

Pour palier aux éventuels inconvénients liés au branchement précédent, notre stratégie consiste à ne rajouter qu'un sous-problème au maximum par nœud de l'arbre de « Branch-and-Bound ». L'idée est de ne brancher que sur les types de colonnes violant la contrainte de non-intersection des objets, où un type de colonne est associé à chaque sous-problème. Dans ce cas, il est seulement nécessaire de séparer le sous-problème associé au type sélectionné comme violant la contrainte.

En effet, en reprenant le branchement précédent, nous remarquons qu'à chaque branchement, de nouveaux sous-problèmes sont générés. Ces problèmes définissent en réalité des sous-ensemble de bandes disjointes, appelés types de bandes. Ces type de bandes sont donc des bandes qui diffèrent par leur composition. Notons $\mathbb{V}$ l'ensemble des types de bandes verticales à un noeud donné de l'arbre et $\mathbb{H}$ l'ensemble des types de bandes horizontales. L'idée est donc définir un branchement que pour un sous-ensemble de bandes qui violent la contrainte (3.1).

Le schéma de branchement

Contrairement au branchement définit précédemment, nous allons utiliser de nouvelles variables binaires $x_{ij}(V)$ (resp. $y_{ij}(H)$) définissant si deux objets $i, j \in \mathcal{I}^2 (i < j)$ sont ensemble sur des bandes verticales (resp. horizontales) du type $V \in \mathbb{V}$ (resp. $H \in \mathbb{H}$). Donc, la contrainte de non-intersection des objets peut aussi s'écrire de cette manière :

$$x_{ij}(V)y_{ij}(H) = 0, \quad \forall i, j \in \mathcal{I} (i < j), V \in \mathbb{V}, H \in \mathbb{H}. \quad (3.11)$$

Pour chaque couple de type de bandes sélectionnées, nous devons donc contrôler que la propriété de non-intersection des objets est vérifiée. Dans le cas contraire, il existe une paire d'objets $(i, j) \in \mathcal{I}^2$ avec $i < j$ et un couple de type de bandes $(V, H) \in \mathbb{V} \times \mathbb{H}$ ne respectant pas la contrainte (3.11). Nous allons donc appliqué le branchement suivant :

$$\left\{ x_{ij}(V) = 0 \right\} \vee \left\{ y_{ij}(H) = 0 \right\} \quad (3.12)$$

Après chaque branchement, nous devons alors modifier les sous-problèmes afin de générer que des bandes respectant la contrainte de branchement. En effet, à chaque variable $x_{ij}(V)$ (resp. $y_{ij}(H)$) fixée à zéro, nous devons garantir de ne pas générer de bandes verticales (resp. horizontales) du type $V \in \mathbb{V}$ (resp. $H \in \mathbb{H}$) et contenant les objets i et j . Ainsi, nous définissons la procédure suivante représentant le schéma de branchement :

Procédure de branchement

- ① Choisir un couple de type de bandes $(V, H) \in \mathbb{V} \times \mathbb{H}$ tel qu'il existe une paire d'objets $(i, j) \in \mathcal{I}^2(i < j)$ violant la contrainte (3.11).
 - ② S'il en existe,
 - ★ Brancher selon la contrainte disjonctive $\left\{x_{ij}(V) = 0\right\} \vee \left\{y_{ij}(H) = 0\right\}$,
 - ★ Supprimer les colonnes violant la contrainte de branchement,
 - ★ Séparer les sous-problèmes à résoudre en fonction des branchements,
 - ★ Ré-optimiser en générant de nouvelles bandes :
 - Si la solution est réalisable, recommencer la procédure de branchement,
 - Si la solution est irréalisable, remonter au nœud père.
- Sinon, sauvegarder la solution et remonter au nœud père.

Choix de la variable de branchement

Nous reprenons les critères de choix développés au branchement précédent adaptées au nouveau branchement.

Soit $o_{ij}(V, H)$ un facteur représentant la violation de la contrainte (3.11) pour un couple de type $(V, H) \in \mathbb{V} \times \mathbb{H}$ et une paire d'objets (i, j) . Ce facteur dépend du critère de choix sélectionné. Soit Ω l'ensemble des couples de types de bandes (V, H) avec une paire d'objets (i, j) violant la contrainte de branchement et ayant la plus grande valeur strictement positive $o_{ij}(V, H)$:

$$\Omega = \left\{ (i, j) \in \mathcal{I}^2(i < j), V \in \mathbb{V}, H \in \mathbb{H} : o_{ij}(V, H) = \max_{\substack{(i', j') \in \mathcal{I}^2(i' < j'), V' \in \mathbb{V}, H' \in \mathbb{H} \\ o_{i'j'}(V', H') > 0}} o_{i'j'}(V', H') \right\}.$$

Ainsi, nous définissons une procédure permettant de choisir un couple de type de bandes (V, H) avec une paire d'objet $(i, j) \in \mathcal{I}^2(i < j)$ violant la propriété de non-intersection des objets :

Procédure de choix d'une disjonction pour le branchement

- ① Calculer $o_{ij}(V, H)$ selon un critère fixé pour tout $(i, j) \in \mathcal{I}^2(i < j)$ et $(V, H) \in \mathbb{V} \times \mathbb{H}$.
- ② Si $\Omega = \emptyset$, alors arrêt du branchement.
Sinon, choisir un couple (V, H) et une paire d'objets (i, j) dans Ω , puis brancher sur $x_{ij}(V)y_{ij}(H) = 0$.

Différents critères sont possibles. Par exemple, nous définissons le couple (V, H) et la paire (i, j) avec la plus grande valeur $x_{ij}(V)y_{ij}(H)$ en considérant la relaxation linéaire :

$$o_{ij}(V, H) = \left(\sum_{v \in V} x_i^v x_j^v \lambda_v \right) \left(\sum_{h \in H} y_i^h y_j^h \mu_h \right), \quad \forall i, j \in \mathcal{I}(i < j), V \in \mathbb{V}, H \in \mathbb{H}. \quad (3.13)$$

Séparation des sous-problèmes

Ce branchement ainsi défini va générer de nouveaux sous-problèmes. Comme pour le branchement précédent, à chaque nœud de l'arbre de « Branch-and-Bound », les sous-problèmes sont de la forme $\mathcal{V}^{i_1, \dots, i_{m_i}, \overline{j_1}, \dots, \overline{j_{m_j}}}$ pour les bandes verticales et de la forme $\mathcal{H}^{i_1, \dots, i_{m_i}, \overline{j_1}, \dots, \overline{j_{m_j}}}$ pour les bandes horizontales. Chaque sous-problème définit alors un sous-ensemble de bandes vérifiant les précédentes contraintes de branchement. Ces sous-ensembles représentent un type de bande sur lesquels nous effectuons un branchement. Ainsi, chaque élément V de $\mathbb{V}$ (resp. H de $\mathbb{H}$) est associé à un unique sous-problème $\mathcal{V}^{i_1, \dots, i_{m_i}, \overline{j_1}, \dots, \overline{j_{m_j}}}$ (resp. $\mathcal{H}^{i_1, \dots, i_{m_i}, \overline{j_1}, \dots, \overline{j_{m_j}}}$).

Supposons que nous branchons sur la variable $x_{ij}(V) = 0$. Alors, le sous-problème associé $\mathcal{V}^{i_1, \dots, i_{m_i}, \overline{j_1}, \dots, \overline{j_{m_j}}}$ est modifié pour respecter la condition de branchement. Les autres sous-problèmes ne changent pas. Ainsi, nous appliquons une procédure de séparation similaire à celle développée en page 95 au seul sous-problème sur lequel nous branchons (problème associé à V). Nous définissons de manière analogue, pour un branchement selon la variable $y_{ij}(H) = 0$, une procédure de séparation du sous-problème associé à H .

Donc, d'après la procédure de branchement, un seul nouveau sous-problème peut être créé à chaque nœud de l'arbre. Cependant, contrairement à l'approche précédente où la profondeur de l'arbre est au plus $\frac{n(n-1)}{2}$, la profondeur de ce schéma ne peut exploser de manière exponentielle. En effet, le nombre de type de bandes est de l'ordre de 2^n dans le pire des cas. En pratique, nous espérons que cette explosion n'arrive que rarement.

Remarque Il est possible d'exprimer ce branchement sous la forme de contraintes à ajouter aux sous problèmes $\mathcal{V}$ et $\mathcal{H}$. Par exemple, pour le sous-problème $\mathcal{V}$, avec un branchement sur la variable $x_{ij}(\mathcal{V}) = 0$ avec $\mathcal{V}$ de la forme $\mathcal{V}^{i_1, \dots, i_{m_i}, \overline{j_1}, \dots, \overline{j_{m_j}}}$, nous pourrions ajouter directement la contrainte suivante au sous-problème $\mathcal{V}$ de départ :

$$\sum_{i'=i_1}^{i_{m_i}} y_{i'} + \sum_{j'=j_1}^{j_{m_j}} (1 - y_{j'}) + x_i + x_j \leq m_i + m_j + 1. \quad (3.14)$$

◆

3.2 Résultats numériques

Notre branchement a été testé sur les instances « ngcut ». Il améliore les bornes obtenues par les formulations $FE_{enr}^{\lambda_v \mu_h}$ et $FE_{enr}^{\lambda_v \mu_h}$ (cf tableau 2.4 du chapitre II) pour l'instance ngcut3 en passant de 249 à 247, la valeur optimale pour un temps de calcul de 2 secondes. Pour l'instance ngcut8, ce branchement a permis d'obtenir une amélioration de la borne de 862 à 856 en 2 secondes. Si nous laissons tourner le processus jusqu'à 1800 secondes, aucune amélioration n'est constatée.

Nous avons aussi étudié ce branchement sur les instances irréalisables définies par Clautiaux et al. Le problème consiste alors à maximiser le nombre d'objets à placer. Si la borne duale trouvée par le processus de génération de colonne permet de montrer que ce nombre est inférieure strictement au nombre d'objets qu'on souhaite placer, l'instance est détectée comme irréalisable. Sinon, le processus n'a pas réussi à prouver l'irréalisabilité. Donc, cette approche ne constitue qu'un test de réalisabilité partiel car, lorsqu'une solution n'est

pas démontrée irréalisable, cela ne garantit pas que l'instance est réalisable. Les résultats trouvés sont reportés dans le tableau 3.5. Notons $U_{FE_{relax}^{\lambda_v \mu_h}}^{br}$ la borne obtenue en appliquant notre branchement spécifique appliqué à la formulation $FE_{relax}^{\lambda_v \mu_h}$. Les colonnes $U_{FE_{relax}^{\lambda_v \mu_h}}^{\delta}$ et $U_{FE_{relax}^{\lambda_v \mu_h}}^{br}$ représentent les tests effectués pour les approches basées sur la programmation mathématique.

Inst.	Tests réal. vérifiés		Temps calcul			
	$U_{FE_{relax}^{\lambda_v \mu_h}}^{\delta}$	$U_{FE_{relax}^{\lambda_v \mu_h}}^{br}$	$U_{FE_{relax}^{\lambda_v \mu_h}}^{\delta}$	$U_{FE_{relax}^{\lambda_v \mu_h}}^{br}$	FS	BR
E00N10	✓	✓	0	0	0	0
E00N15	✗	✓	0	28	0	0
E00N23	✓	✓	0	0	*	0
E02N20	✗	✓	0	10	0	0
E03N10	✓	✓	0	0	0	0
E03N15	✗	✓	0	0	0	0
E03N16	✗	?	0	*	2	6
E03N17	✗	✓	0	0	0	0
E04N15	✓	✓	0	0	0	0
E04N17	✓	✓	0	0	0	0
E04N18	✓	✓	0	0	10	0
E05N15	✗	✓	0	1	0	0
E05N17	✗	?	0	*	0	0
E07N10	✗	✗	0	0	0	0
E07N15	✓	✓	0	0	0	0
E08N15	✗	?	0	*	0	0
E10N10	✓	✓	0	0	0	0
E10N15	✓	✓	8	0	0	0
E13N10	✗	?	0	*	0	0
E13N15	✗	?	0	*	0	0
E15N10	✗	?	0	*	-	-
E15N15	✗	✓	0	0	0	0
E00X23	✗	?	0	*	*	*
E05X15	✓	✓	0	0	2	0
E07X15	✗	?	0	*	0	1
E10X15	✗	?	0	*	0	0
E13X15	✗	✓	0	4	0	0

TABLEAU 3.5: Etude des tests de réalisabilité sur les instances irréalisables définies par Clautiaux et al.

Dans le tableau 3.5, le symbole ✓ représente les cas où l'instance a été détectée irréalisable et le symbole ✗ représente les cas où le processus ne prouve pas l'irréalisabilité. Les cas non résolus après 1800 secondes (cas où le processus n'a pu donner de réponse sur la réalisabilité ou non du placement après 1800 secondes) sont indiqués par le symbole ?. Les résultats qui suivent ont été effectuées sur un PC avec un processeur 2.8GHz Pentium Xeon.

Les résultats obtenus par Belov et Rohling sont aussi reportés dans le tableau 3.5 (cf.

colonne BR) et ont été obtenues sur un PC avec un processeur 2.66GHz Pentium Xeon. Leur procédure utilise l'approche de Fekete et Schepers associée à l'approche de génération de colonnes définie par la formulation $FE_{env}^{\lambda_v \mu_h}$. Ce choix permet une amélioration des performances originelles obtenues par Fekete et Schepers (cf. colonne FS).

Le tableau 3.5 montre que notre branchement permet de détecter les placements irréalisables plus efficacement (seul dans le cas de l'instance E07N10, l'algorithme termine sur une situation indécise) et représente une des approches les plus efficaces pour la résolution du problème de placement par les approches de programmation mathématiques.

Cependant, les approches de génération de colonnes n'offrent que des bornes duales au problème d'origine et une étude post-optimisation est nécessaire pour connaître la validité de la solution optimale trouvée. Afin d'améliorer la qualité des bornes, l'ajout de contraintes pour imposer la consécutivité des bandes est nécessaire et constitue un axe de recherche pour le futur. Par exemple, pour la solution trouvée pour l'instance E07N10 (représentée par la figure 3.6), la présence de triplets d'objets incompatibles (les objets 4, 7, 9 forment une clique dans le graphe associé mais ils ne sont jamais présents dans une même bande verticale) est à l'origine de la défaillance du test.

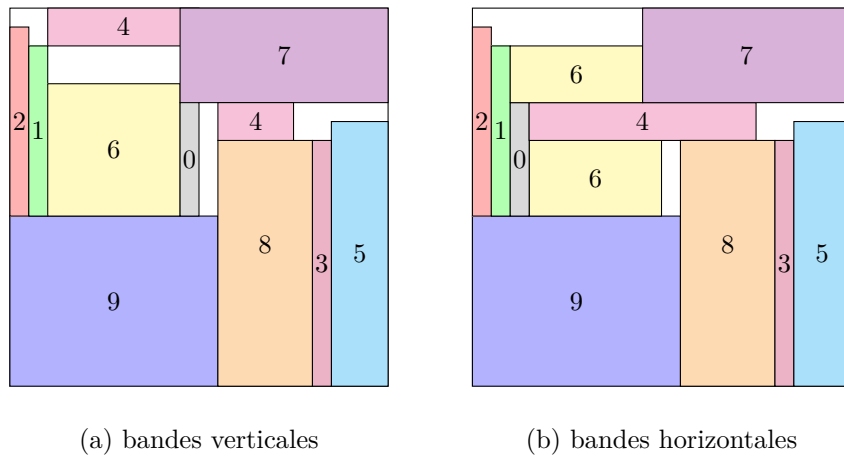


FIGURE 3.6: Solution invalide trouvée après branchement sur l'instance E07N10

De plus, ces approches restent moins efficace que les approches combinatoires (colonne FS) ou les approches hybrides entre approche combinatoires et approche de génération de colonnes (colonne BR). Les approches combinatoires restent donc, à l'heure actuelle, les approches qui permettent une meilleure résolution grâce à une élimination plus efficaces des cas symétriques. Une meilleure sélection des objets violant la contrainte d'intersection des objets et la gestion des cas avec des objets de taille identique permettraient sûrement une amélioration des résultats.

Dans le chapitre qui suit, nous proposons de nouvelles approches combinatoires de résolution du problème de placement permettant une résolution plus efficace des instances.

Sommaire

4.1	Le problème de placement multi-dimensionnel	104
4.2	Approche par génération de matrices de uns-consécutifs	105
4.3	Approche par génération de MPQ-arbres	121
4.4	Résultats numériques	133

CHAPITRE IV

Résolution par approches combinatoires

Ce chapitre est consacré aux différentes approches combinatoires développées durant cette thèse pour résoudre le problème de placement. Elles ont pour but d'essayer de corriger les inconvénients de l'approche de Fekete et Schepers constatés en section 1.4 tout en conservant ses avantages.

Pour cela, nous avons utilisé les caractérisations par les graphes d'intervalles par des matrices de uns-consécutifs (par Fulkerson et Gross) et des MPQ-arbres (par Korte et Möhring). La définition de largeur associée aux colonnes pour la première méthode et de largeur de nœud pour la deuxième nous permet alors de définir de nouvelles caractérisations des placements plus adaptées. Nous les exploitons ensuite pour développer une approche ne dépendant pas d'une discrétisation de l'espace.

Dans un premier temps, nous généralisons la définition du problème de placement au cas à D-dimensions. Puis, nous utilisons les caractérisations développées en section 1.2 pour en déduire de nouveaux algorithmes de résolution du problème de placement. Enfin, nous terminons par quelques résultats numériques.

Les résultats exploitant les MPQ-arbres ont été obtenus avec Petru Valicov.

4.1 Le problème de placement multi-dimensionnel

Nous allons généraliser d'abord certaines définitions exposées au chapitre I.

Soit $D > 1$ le nombre de dimensions considérées, c'est-à-dire la dimension de l'espace euclidien considéré. Notons $\mathcal{I} = \{1, \dots, n\}$ l'ensemble d'objets tel que chaque objet $i \in \mathcal{I}$ a une taille fixe $(\mathbf{w}_i^1, \dots, \mathbf{w}_i^D) \in \mathbb{R}_{+*}^D$. Et soit un conteneur de taille $(\mathbf{W}^1, \dots, \mathbf{W}^D) \in \mathbb{R}_{+*}^D$.

Définition 4.1 (Placement sur D dimensions)

L'ensemble $\mathcal{I}$ a un placement sur D dimensions réalisable si et seulement s'il existe un ensemble de coordonnées $(x_i^1, \dots, x_i^D) \in [0, \mathbf{W}^1[\times \dots \times [0, \mathbf{W}^D[$ associé à chaque objet $i \in \mathcal{I}$ vérifiant les conditions suivantes :

$$\forall i \in \mathcal{I}, d \in \{1, \dots, D\} : x_i^d + \mathbf{w}_i^d \leq \mathbf{W}^d, \quad (4.1a)$$

$$\forall i, j \in \mathcal{I} (i \neq j), \exists d \in \{1, \dots, D\} : [x_i^d, x_i^d + \mathbf{w}_i^d[\cap [x_j^d, x_j^d + \mathbf{w}_j^d[= \emptyset. \quad (4.1b)$$

De plus, un placement sur D dimensions vérifie la condition « calée à gauche » si, pour chaque dimension $d \in \{1, \dots, D\}$, chaque objet $i \in \mathcal{I}$ touche soit un autre objet, soit le bord du conteneur :

$$x_i^d \in \{0\} \cup \{x_j^d + \mathbf{w}_j^d : j \in \mathcal{I} \setminus \{i\}\}, \quad \forall i \in \mathcal{I}, d \in \{1, \dots, D\}.$$

◆

Lemme 4.2

Pour chaque ensemble $\mathcal{I}$ réalisable, il existe un placement associé vérifiant la condition « calée à gauche » sur chaque dimension.

Preuve. Similaire à la preuve du lemme 1.3. □

L'approche de Fekete et Schepers est valide dans le cas multi-dimensionnel :

Théorème 4.3 (Fekete et Schepers 1997 [27])

Un ensemble $\mathcal{I}$ d'objets peut être placé dans le conteneur si et seulement s'il existe un ensemble de D graphes $G_d = (\mathcal{I}, E_d)$, $d \in \{1, \dots, D\}$ avec les propriétés suivantes :

$$G_d \text{ est un graphes d'intervalles, } \forall d \in \{1, \dots, D\}, \quad (4.2a)$$

$$\alpha(G_d, \vec{\mathbf{w}}^d) \leq \mathbf{W}^d, \quad \forall d \in \{1, \dots, D\}, \quad (4.2b)$$

$$\bigcap_{d \in \{1, \dots, D\}} E_d = \emptyset. \quad (4.2c)$$

Un ensemble de D graphes vérifiant les propriétés (4.2a)-(4.2c) est appelé classe de placement.

4.2 Approche par génération de matrices de uns-consécutifs

Pour éviter d'énumérer des classes de placement « symétriques » ou dégénérées, nous avons d'abord utilisé la caractérisation de Fulkerson et Gross définie en section 1.2 pour concevoir un nouvel algorithme.

4.2.1 Décomposition par bandes d'un placement

Pour chaque dimension $d \in \{1, \dots, D\}$, soit $M^d \in \mathcal{M}_{n, m^d}(\{0, 1\})$ une matrice binaire avec $m^d \leq n$ représentant le nombre de colonnes de la matrice. $\forall 1 \leq k \leq m^d$, soit $Q_k^d = \{i \in \mathcal{I} : M_{ik}^d = 1\}$ l'ensemble des objets présents dans la colonne k et soit $C_k^d \in \{0, 1\}^n$ la colonne k de la matrice M^d (C_k^d est le vecteur caractéristique de Q_k^d). De plus, une colonne C_k^d est dite maximale si Q_k^d n'est pas incluse dans Q_l^d pour tout $l \neq k$. Soit $Q_{m^d+1}^d = \emptyset$, et soit $\mathcal{Q}^d = \{Q_1^d, \dots, Q_{m^d}^d\}$.

Enfin, $\forall d \in \{1, \dots, D\}$ et $1 \leq k \leq m^d$, nous définissons la largeur d'une colonne Q_k^d , noté λ_k^d , par :

$$\lambda_k^d = \max_{\substack{i \in \mathcal{I} \\ i \in Q_k^d, i \notin Q_{k+1}^d}} \left\{ \mathbf{w}_i^d - \sum_{1 \leq l < k / i \in Q_l^d} \lambda_l^d \right\}. \quad (4.3)$$

Définition 4.4 (Décomposition par bandes)

Une décomposition par bande associée à l'ensemble $\mathcal{I}$ est un ensemble de D matrices binaires $(M^1, \dots, M^D) \in \mathcal{M}(\{0, 1\})$ avec la propriété des uns-consécutifs et sans ligne nulle tel que :

- Pour chaque dimension $d \in \{1, \dots, D\}$,

$$\text{Chaque colonne de } M^d \text{ est maximale,} \quad (4.4a)$$

$$\forall i \in \mathcal{I}, \forall 1 \leq k \leq m^d \text{ tel que } i \in Q_k^d, \mathbf{w}_i^d - \sum_{1 \leq l < k / i \in Q_l^d} \lambda_l^d > 0, \quad (4.4b)$$

$$\sum_{1 \leq k \leq m^d} \lambda_k^d \leq \mathbf{w}^d. \quad (4.4c)$$

- Pour chaque paire d'objets $i, j \in \mathcal{I}$ avec $i < j$, il existe une dimension $d \in \{1, \dots, D\}$ telle que :

$$\forall 1 \leq k \leq m^d, i \notin Q_k^d \text{ ou } j \notin Q_k^d. \quad (4.4d)$$

◆

La figure 4.1 donne un exemple d'une décomposition par bandes pour un placement sur deux dimensions. La condition (4.4a) permet de n'avoir que des matrices sommets/cliques maximales de uns-consécutifs. La condition (4.4b) évite de considérer des cas de matrices dégénérés. La condition (4.4c) assure que la taille du conteneur est respectée et la condition (4.4d) garantit la non intersection des objets.

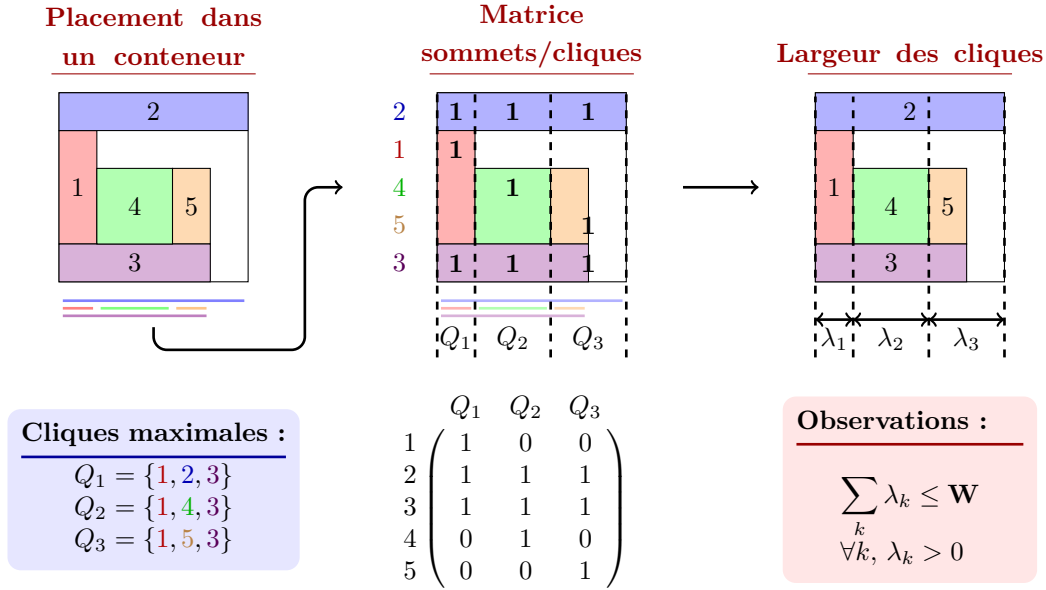


FIGURE 4.1: Une décomposition par bandes sur la première dimension d'un placement sur deux dimensions

Remarque Puisque $\lambda_k^d > 0$, nous éliminons les cas dégénérés décrit par Ferreira et Oliveira [34]. En effet, pour chaque matrice de uns-consécutifs avec au moins une colonne de largeur nulle, nous pouvons construire une matrice de uns-consécutifs sans ces colonnes. ♦

Nous avons alors les lemmes suivants illustrant les liens entre placement et décomposition par bandes (cf. figure 4.1) :

Lemme 4.5

Pour chaque objet $i \in \mathcal{I}$ et dimension $d \in \{1, \dots, D\}$, $\mathbf{w}_i^d \leq \sum_{1 \leq k \leq m^d / i \in Q_k^d} \lambda_k^d$.

Preuve. Pour chaque $i \in \mathcal{I}$ et $d \in \{1, \dots, D\}$, soit $1 \leq l \leq m^d$ tel que $i \in Q_l^d$ et $i \notin Q_{l+1}^d$.

Nous avons $\lambda_l^d = \max_{\substack{j \in \mathcal{I} \\ j \in Q_l^d, j \notin Q_{l+1}^d}} \left\{ \mathbf{w}_j^d - \sum_{1 \leq k < l / j \in Q_k^d} \lambda_k^d \right\} \geq \mathbf{w}_i^d - \sum_{1 \leq k < l / i \in Q_k^d} \lambda_k^d$.

D'où, $\mathbf{w}_i^d \leq \sum_{1 \leq k \leq m^d / i \in Q_k^d} \lambda_k^d$. □

Soit $M \in \mathcal{M}_{n,m}(\{0, 1\})$ une matrice avec la propriété des uns-consécutifs et seulement des colonnes maximales tel que $\sum_k \lambda_k \leq \mathbf{W}$. Soit G le graphe d'intervalles associé. Alors, la propriété (4.2b) est vérifiée pour chaque graphe associé à chaque dimension de l'espace :

Lemme 4.6

$\forall d \in \{1, \dots, D\}, \alpha(G_d, \vec{\mathbf{w}}^d) \leq \sum_k \lambda_k^d \leq \mathbf{W}^d$.

Preuve. Soit $\mathcal{Q}^d$ l'ensemble des cliques maximales de G_d . Pour chaque sommet i de G_d et

chaque clique $Q \in \mathcal{Q}^d$, soit $\delta_{i \in Q} = 1$ si $i \in Q$; 0 sinon. D'après le lemme 4.5, nous avons :

$$\sum_{Q \in \mathcal{Q}^d} \delta_{i \in Q} \lambda_Q^d \geq \mathbf{w}_i^d, \quad \forall i \in \mathcal{I}.$$

Soit S^d un ensemble stable de G_d tel que $\sum_{i \in S^d} \mathbf{w}_i^d = \alpha(G_d, \vec{\mathbf{w}}^d)$. Alors, nous avons :

$$\alpha(G_d, \vec{\mathbf{w}}^d) = \sum_{i \in S} \mathbf{w}_i^d \leq \sum_{i \in S^d} \sum_{Q \in \mathcal{Q}^d} \delta_{i \in Q} \lambda_Q^d = \sum_{Q \in \mathcal{Q}^d} \sum_{i \in S^d} \delta_{i \in Q} \lambda_Q^d \leq \sum_{Q \in \mathcal{Q}^d} \lambda_Q^d \leq \mathbf{W}^d.$$

En effet, par définition d'un ensemble stable, $\sum_{i \in S^d} \delta_{i \in Q} \leq 1, \forall Q \in \mathcal{Q}$. □

Remarque Dans un graphe d'intervalles, le nombre de cliques maximales est borné par le nombre de sommets. Or, comme chaque colonne d'une décomposition par bandes représente une clique maximale (d'après la maximalité des colonnes). Nous en déduisons donc que $m^d \leq n, \forall d \in \{1, \dots, D\}$. ◆

Les deux lemmes suivants permettent de démontrer que la décomposition par bandes est une caractérisation nécessaire et suffisante pour les problèmes de placement.

Lemme 4.7

Si nous avons un placement réalisable pour $\mathcal{I}$ avec la condition « calée à gauche », alors il existe une décomposition par bandes associée à $\mathcal{I}$.

Preuve. Soit un placement réalisable avec la condition « calée à gauche » $(x_i^1, \dots, x_i^D) \in \mathbb{R}_+^D$ pour chaque objet $i \in \mathcal{I}$. Pour tout $d \in \{1, \dots, D\}$, soit $X^d = \{x \in \mathbb{R} : \exists i \in \mathcal{I}, x = x_i^d\}$. Posons $m^d = |X^d|$. Sans perte de généralité, nous posons $X^d = \{X_1^d, \dots, X_{m^d}^d\}$ avec $X_1^d < \dots < X_{m^d}^d$.

Comme les coordonnées du placement vérifient la condition « calée à gauche », alors pour chaque $d \in \{1, \dots, D\}$, $2 \leq k \leq m^d$, il existe $i \in \mathcal{I}$ tel que $X_k^d = x_i^d + \mathbf{w}_i^d$. De plus, $X_1^d = 0$. Soit $X_{m^d+1}^d = \max\{x_i^d + \mathbf{w}_i^d : i \in \mathcal{I}\}$.

Evidemment, pour chaque $d \in \{1, \dots, D\}$, il existe une fonction surjective :

$$\sigma^d : \mathcal{I} \rightarrow \{1, \dots, m^d\} \text{ telle que } \forall i \in \mathcal{I}, x_i^d = X_{\sigma^d(i)}^d.$$

$\forall d \in \{1, \dots, D\}$, $1 \leq k \leq m^d$, posons $Q_k^d = \{i \in \mathcal{I} / x_i^d \leq X_k^d < x_i^d + \mathbf{w}_i^d\}$. Et soit, $M^d \in \mathcal{M}_{(n, m^d)}(\{0, 1\})$ la matrice booléenne telle que $M_{ik}^d = 1$ si et seulement si $i \in Q_k^d$. Montrons alors que cet ensemble de D matrices vérifient les propriétés de la décomposition par bandes.

- M^d possède la propriété des uns-consécutifs. En effet, $\forall i \in \mathcal{I}$, soit $1 \leq k, l \leq m^d$, avec $X_k^d \leq X_l^d$, tel que $i \in Q_k^d$ et $i \in Q_l^d$. Alors, par définition, $x_i^d \leq X_k^d < x_i^d + \mathbf{w}_i^d$ et $x_i^d \leq X_l^d < x_i^d + \mathbf{w}_i^d$. Soit $1 \leq h \leq m^d$ avec $X_k^d \leq X_h^d \leq X_l^d$, nous avons donc l'inégalité suivante : $x_i^d \leq X_k^d \leq X_h^d \leq X_l^d < x_i^d + \mathbf{w}_i^d$. D'où, $i \in Q_h^d$ et M^d vérifie la propriété des uns-consécutifs.
- Nous devons vérifier que chaque colonne M^d est maximale. Soit $1 \leq k, l \leq m^d$ avec $X_k^d < X_l^d$ et soit C_k^d (resp. C_l^d) le vecteur caractéristique de Q_k^d (resp. Q_l^d). Comme le placement vérifie la condition « calée à gauche », soient i et $j \in \mathcal{I}$ deux objets tels

que $x_i^d + \mathbf{w}_i^d = X_{k+1}^d$ et $x_j^d = X_l^d$. Par définition, $X_k^d < X_l^d$. D'où, $j \notin Q_k^d$ et $j \in Q_l^d$. Alors, $Q_l^d \not\subseteq Q_k^d$. De plus, $X_{k+1}^d \leq X_l^d$. D'où, $i \in Q_k^d$ et $i \notin Q_l^d$. Alors, $Q_k^d \not\subseteq Q_l^d$ et donc C_k^d est maximale.

- Pour tout $d \in \{1, \dots, D\}$, $1 \leq k \leq m^d$, soit $\lambda_k^d = \max_{\substack{i \in \mathcal{I} \\ i \in Q_k^d, i \notin Q_{k+1}^d}} \left\{ \mathbf{w}_i^d - \sum_{1 \leq l < k/i \in Q_l^d} \lambda_l^d \right\}$.

Soit $\mathcal{I}_k^d = \{i \in \mathcal{I} / x_i^d \leq X_k^d < x_i^d + \mathbf{w}_i^d \leq X_{k+1}^d\} = \{i \in \mathcal{I} / i \in Q_k^d, i \notin Q_{k+1}^d\}$. D'après la condition « calée à gauche », nous avons $X_k^d = \max_{i \in \mathcal{I}_{k-1}^d} \{x_i^d + \mathbf{w}_i^d\}$ pour $2 \leq k \leq m^d$.

Nous allons montrer par récurrence que $\lambda_k^d = X_{k+1}^d - X_k^d > 0$, pour $1 \leq k \leq m^d$.

$$\begin{aligned} \cdot \text{ Pour } k = 1, \lambda_1^d &= \max_{i \in \mathcal{I}_1^d} \left\{ \mathbf{w}_i^d - \sum_{1 \leq l < k/i \in Q_l^d} \lambda_l^d \right\} = \max_{i \in \mathcal{I}_1^d} \{\mathbf{w}_i^d\} = X_2^d - X_1^d > 0. \\ \cdot \text{ Pour } k \geq 2, \lambda_k^d &= \max_{i \in \mathcal{I}_k^d} \left\{ \mathbf{w}_i^d - \sum_{1 \leq l < k/i \in Q_l^d} \lambda_l^d \right\} \\ &= \max_{i \in \mathcal{I}_k^d} \left\{ \mathbf{w}_i^d - \sum_{1 \leq l < k/i \in Q_l^d} (X_l^d - X_{l+1}^d) \right\} \quad \text{par hyp. de réc.} \\ &= \max_{i \in \mathcal{I}_k^d} \{\mathbf{w}_i^d + X_{\sigma^d(i)}^d - X_k^d\} \\ &= \max_{i \in \mathcal{I}_k^d} \{x_i^d + \mathbf{w}_i^d\} - X_k^d = X_{k+1}^d - X_k^d > 0. \end{aligned}$$

Donc, $\forall d \in \{1, \dots, D\}$,

$$\begin{aligned} \cdot \sum_{1 \leq k \leq m^d} \lambda_k^d &= \sum_{1 \leq k \leq m^d} (X_{k+1}^d - X_k^d) = X_{m^d+1}^d = \max\{x_i^d + \mathbf{w}_i^d : i \in \mathcal{I}\} \leq \mathbf{W}^d. \\ \cdot \forall 1 \leq k \leq m^d, \forall i \in \mathcal{I}_k^d, \\ \mathbf{w}_i^d - \sum_{1 \leq l < k/i \in Q_l^d} \lambda_l^d &= \mathbf{w}_i^d - \sum_{1 \leq l < k/i \in Q_l^d} (X_l^d - X_{l+1}^d) \\ &= \mathbf{w}_i^d + X_{\sigma^d(i)}^d - X_k^d = x_i^d + \mathbf{w}_i^d - X_k^d > 0. \end{aligned}$$

- Comme nous avons un placement réalisable, grâce à la condition (4.1b), nous avons $\forall i, j \in \mathcal{I}$ avec $i < j$, $\exists d \in \{1, \dots, D\} : [x_i^d, x_i^d + \mathbf{w}_i^d) \cap [x_j^d, x_j^d + \mathbf{w}_j^d) = \emptyset$. Donc, soit $x_i^d + \mathbf{w}_i^d \leq x_j^d$, soit $x_j^d + \mathbf{w}_j^d \leq x_i^d$. D'où, pour tout $1 \leq k \leq m^d$, par définition de Q_k^d , $i \notin Q_k^d$ ou $j \notin Q_k^d$.

Donc $(M^1, \dots, M^D)$ est une décomposition par bandes associée à $\mathcal{I}$. □

Lemme 4.8

Si nous avons une décomposition par bandes associée à $\mathcal{I}$, alors il existe un placement réalisable pour $\mathcal{I}$ avec la condition « calée à gauche ».

Preuve. Soit un ensemble de D matrices $(M^1, \dots, M^D)$ vérifiant les conditions d'une décomposition par bandes. $\forall d \in \{1, \dots, D\}$, $i \in \mathcal{I}$, soit f_i^d (resp. l_i^d) le plus petit (resp. grand) indice k avec $1 \leq k \leq m^d$ tel que $M_{ik}^d = 1$.

Posons $x_i^d = \sum_{1 \leq k \leq f_i^d - 1} \lambda_k^d$, $\forall d \in \{1, \dots, D\}$, $i \in \mathcal{I}$.

Nous devons montrer que l'ensemble de coordonnées $(x_i^1, \dots, x_i^D)_{i \in \mathcal{I}}$ définit un placement valide.

$$- \forall d \in \{1, \dots, D\}, \forall i \in \mathcal{I},$$

$$x_i^d + \mathbf{w}_i^d = \sum_{1 \leq k \leq f_i^d - 1} \lambda_k^d + \mathbf{w}_i^d \leq \sum_{1 \leq k \leq f_i^d - 1} \lambda_k^d + \sum_{f_i^d \leq k \leq l_i^d} \lambda_k^d \leq \mathbf{W}^d.$$

D'où, la contrainte (4.1a) est satisfaite.

$$- \forall i, j \in \mathcal{I} \text{ avec } i < j, \text{ il existe } d \in \{1, \dots, D\} \text{ tel que pour tout } 1 \leq k \leq m^d, i \notin Q_k^d \text{ ou } j \notin Q_k^d. \text{ Comme, } M^d \text{ est une matrices de uns-consécutifs, alors } l_i^d < f_j^d \text{ ou } l_j^d < f_i^d. \text{ Donc, } x_i^d + \mathbf{w}_i^d \leq x_j^d \text{ ou } x_j^d + \mathbf{w}_j^d \leq x_i^d. \text{ D'où, la contrainte (4.1b) est satisfaite.}$$

$$- \text{Montrons que la condition « calée à gauche » est vérifiée. } \forall d \in \{1, \dots, D\}, i \in \mathcal{I},$$

$$. \text{ Si } f_i^d = 1 \text{ alors } x_i^d = 0.$$

$$. \text{ Sinon, soit } j \in \mathcal{I} \text{ tel que } j \in Q_{f_i^d - 1}^d, j \notin Q_{f_i^d}^d \text{ et } \mathbf{w}_j^d = \sum_{1 \leq l \leq f_i^d - 1 / j \in Q_l^d} \lambda_l^d.$$

$$\text{D'où, } \mathbf{w}_j^d = \sum_{f_j^d \leq l < f_i^d - 1} \lambda_l^d. \text{ Et donc, } x_i^d = \sum_{k \in \{1, \dots, f_j^d - 1\}} \lambda_k^d + \sum_{k \in \{f_j^d, \dots, f_i^d - 1\}} \lambda_k^d = x_j^d + \mathbf{w}_j^d.$$

□

4.2.2 L'algorithme

En partant des résultats des lemmes 4.7 et 4.8, pour contrôler la réalisabilité d'un ensemble d'objets $\mathcal{I}$, nous avons conçu un algorithme retournant, s'il en existe, une décomposition par bandes associée à $\mathcal{I}$.

Pour cela, l'algorithme 2 illustre la manière dont la procédure évolue : pour chaque dimension, nous cherchons une matrice de uns-consécutifs de taille $n \times n$ satisfaisant les contraintes (4.4a), (4.4b) et (4.4c). Ainsi, l'algorithme procède, dimension par dimension, en énumérant toutes les ensembles de D matrices de uns consécutifs valides pour chaque dimension. Cette énumération de matrices s'effectue par l'appel d'une fonction récursive spécifique.

La contrainte (4.4d) est vérifiée seulement à la fin de la procédure dans le cas où nous avons trouvé un ensemble de D matrices valides.

Remarque En appliquant l'algorithme 2, nous considérons donc seulement une dimension à la fois. Dans le cas du problème de placement sur deux dimensions, l'essentiel du travail est fait surtout dans la première dimension. ♦

Pour une dimension donnée, une fonction récursive est donc appelé pour permettre de chercher une matrice valide. Le cœur de la récursion énumérant les matrices de uns-consécutifs valides est décrit dans l'algorithme 3 en partant de `row=column=1`.

Dans l'algorithme 3, nous remarquons que le nombre de colonnes de la matrice est de taille n . En effet, nous connaissons pas, a priori, le nombre de bandes nécessaires pour

Algorithme 2 Fonction recherchant une décomposition valide

Require: an instance I (a set of items $\mathcal{I}$ and a container)

Ensure: print a valid strip decomposition

```

decomp (instance  $I$ )
  begin
    integer  $\text{dim} = 1$ 
    while  $\text{dim} \geq 0$  do
      initialisation of the boolean matrix  $\text{mat}[n][n]$ 
      recurse (1,1, $\text{dim}$ , $\text{mat}$ , $I$ )    # call function to search an other valid matrix in  $\text{dim}$ 
      if (find valid matrix in dimension  $\text{dim}$  and  $\text{dim} = D$ ) then
        if (compatibility between matrices according to constraint 4.4d) then
          print valid decomposition strip
          return true
        else
           $\text{dim} = \text{dim} - 1$ 
        end if
      else
        if (find valid matrix in dimension  $\text{dim}$ ) then
           $\text{dim} = \text{dim} + 1$ 
        else
           $\text{dim} = \text{dim} - 1$ 
        end if
      end if
    end while
    return false
  end

```

trouver une décomposition par bandes. Mais, nous savons qu'il est borné par n . Notre algorithme énumère donc des matrices $n \times n$ avec m colonnes maximales et des colonnes vides.

Remarque La vérification de la contrainte (4.4d) peut aussi se faire au cours de la recherche de matrices valides sur la dimension D de la façon suivante : soit $(M^1, \dots, M^{D-1})$ un ensemble de $D-1$ matrices valides trouvé au cours de l'algorithme 2. La condition (4.4d) peut alors être vérifiée dans l'algorithme 3 lorsque le test de la consécutivité des uns est effectué. ♦

4.2.3 Améliorations des performances de l'algorithme

4.2.3.1 Détection d'irréalisation

Dans cette sous-section, nous allons donner des contraintes supplémentaires valides pour toutes les décompositions par bandes. Ces contraintes sont utilisées dans notre algorithme pour réduire l'arbre d'énumération des décompositions par bandes.

Algorithme 3 Fonction récursive générant des matrices de uns-consécutifs satisfaisant les contraintes (4.4a), (4.4b) et (4.4c)

Require: an instance I , a dimension d , and a boolean matrix M

Ensure: print all valid matrices $n \times n$ without 0 rows

```

MatrixRecurse (integer row, integer column, integer dim, boolean mat[n][n], instance I)
  begin
    if column > n+1 then
      return
    end if
    if row > n+1 then
      if (constraints (4.4a), (4.4b) and (4.4c) are satisfied by mat on dimension dim) then
        if (number of non empty rows = n) then
          print mat
        else
          MatrixRecurse(1, column+1, dim, mat[n][n], I)
        end if
      end if
    end if
    return
  end if
  mat(row, column) = true
  if (mat is still a consecutive ones matrix) then
    MatrixRecurse(row+1, column, dim, mat[n][n], I)
  end if
  mat(row, column) = false
  MatrixRecurse(row+1, column, dim, mat[n][n], I)
end

```

Contrainte de respect de la « largeur » disponible pour les objets non encore placés

Supposons que les k premières colonnes de la matrice énumérée par l'algorithme 3 sont fixées. Chaque objet non encore placé (objets dont la ligne de la matrice associé est nulle) doit être placé dans les colonnes restantes. Donc, la contrainte suivante est une contrainte valide (cf. figure 4.2) :

Lemme 4.9

Pour tout $d \in \{1, \dots, D\}$ and $1 \leq k \leq m^d$,

$$\sum_{1 \leq l \leq k} \lambda_l^d + \max_{i \notin \{Q_1^d, \dots, Q_k^d\}} \mathbf{w}_i^d \leq \mathbf{W}^d. \quad (4.5)$$

Preuve. En effet, $\forall d \in \{1, \dots, D\}$ et $1 \leq k \leq m^d$, $\max_{i \notin \{Q_1^d, \dots, Q_k^d\}} \mathbf{w}_i^d \leq \sum_{k+1 \leq l \leq m^d} \lambda_l^d$. $\square$

Contrainte de respect de la « hauteur » pour chaque colonne

Pour le problème de placement sur deux dimensions, une clique maximale Q sur une dimension induit un ensemble stable dans l'autre dimension. D'où, pour chaque colonne

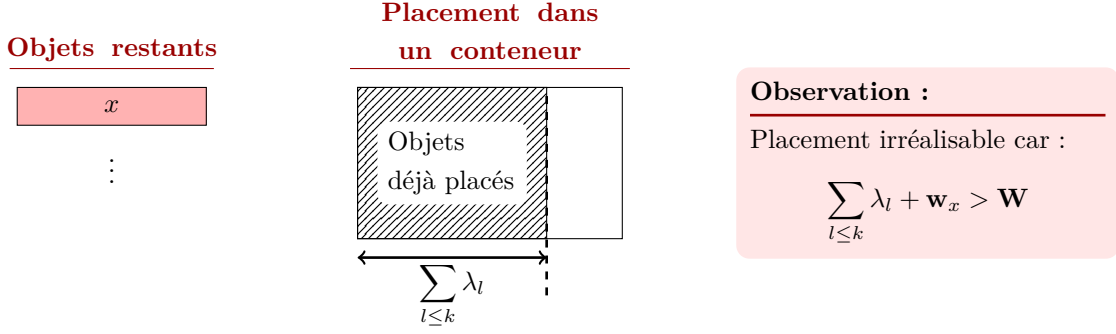


FIGURE 4.2: Exemple de placement irréalisable car la largeur de l'objet x est trop importante pour rentrer dans la partie disponible du conteneur

Q , la somme des poids des objets de Q (ou « hauteur » de Q) ne peut pas excéder la taille du conteneur dans l'autre dimension d'après la propriété (4.2b). Pour D dimensions, nous avons la contrainte valide suivante :

Lemme 4.10

Pour tout $d \in \{1, \dots, D\}$ et $1 \leq k \leq m^d$,

$$\sum_{i \in Q_k^d} \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} w_i^{d'} \right) \leq \prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} w^{d'}. \quad (4.6)$$

Preuve. Soit $d \in \{1, \dots, D\}$ et $1 \leq k \leq m^d$. D'après le lemme 4.5, nous avons :

$$\sum_{i \in Q_k^d} \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} w_i^{d'} \right) \leq \sum_{i \in Q_k^d} \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \sum_{\substack{1 \leq l \leq m^{d'} \\ i \in Q_l^{d'}}} \lambda_l^{d'} \right).$$

Soit $\mathcal{Z} = \{1, \dots, m^1\} \times \dots \times \{1, \dots, m^{d-1}\} \times \{1, \dots, m^{d+1}\} \times \dots \times \{1, \dots, m^D\}$.

Soit $L_i^{d'} = \{1 \leq l \leq m^{d'} : i \in Q_l^{d'}\}$, pour tout $i \in \mathcal{I}$ et $d' \in \{1, \dots, D\}$ avec $(d' \neq d)$.

Soit $\mathcal{Z}_i = L_i^1 \times \dots \times L_i^{d-1} \times L_i^{d+1} \times \dots \times L_i^D \subseteq \mathcal{Z}$. Remarquons que $\mathcal{Z}_i$ n'est pas vide comme $L_i^{d'} \neq \emptyset$ pour tout $d' \in \{1, \dots, D\}$ avec $d' \neq d$.

Soit P un polynôme de $\mathbb{R}[X_{z \in \mathcal{Z}}]$ et, pour tout $i \in \mathcal{I}$, P_i des polynômes de $\mathbb{R}[X_{z \in \mathcal{Z}_i}]$ définis par :

$$P = \sum_{(l_1, \dots, l_{d-1}, l_{d+1}, \dots, l_D) \in \mathcal{Z}} X_{(1, l_1)} X_{(2, l_2)} \dots X_{(d-1, l_{d-1})} X_{(d+1, l_{d+1})} \dots X_{(D, l_D)},$$

$$P_i = \sum_{(l_1, \dots, l_{d-1}, l_{d+1}, \dots, l_D) \in \mathcal{Z}_i} X_{(1, l_1)} X_{(2, l_2)} \dots X_{(d-1, l_{d-1})} X_{(d+1, l_{d+1})} \dots X_{(D, l_D)},$$

$\forall d' \in \{1, \dots, D\}$ avec $d' \neq d$ et $\forall (l_1, \dots, l_{d-1}, l_{d+1}, \dots, l_D) \in \mathcal{Z}$, posons $X_{(d', l_{d'})} = \lambda_{l_{d'}}^{d'}$ et notons par $P(\lambda)$ (resp. $P_i(\lambda)$) la valeur du polynôme P (resp. P_i pour tout $i \in \mathcal{I}$) pour variable λ .

D'après l'équation (4.4d), $\forall i, j \in \mathcal{I} (i \neq j)$, il existe $d' \in \{1, \dots, D\}$ tel que pour tout $1 \leq l \leq m^{d'}$, $i \notin Q_l^{d'}$ ou $j \notin Q_l^{d'}$. D'où, pour chaque paire d'objets distincts $i, j \in Q_k^d$, nous avons $\mathcal{Z}_i \cap \mathcal{Z}_j = \emptyset$.

Donc, le polynôme $\sum_{i \in Q_d^k} P_i(\lambda)$ est un polynôme homogène de degré $D - 1$ tel que tous les coefficients non nuls sont égaux à 1. Nous en déduisons alors l'inégalité suivante :

$$\sum_{i \in Q_k^d} \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \sum_{\substack{1 \leq l \leq m^{d'} \\ i \in Q_l^{d'}}} \lambda_l^{d'} \right) = \sum_{i \in Q_d^k} P_i(\lambda) \leq P(\lambda) = \prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \left(\sum_{1 \leq l \leq m^{d'}} \lambda_l^{d'} \right)$$

D'où, l'inégalité (4.6) est valide car pour tout $d' \in \{1, \dots, D\}$, $\sum_{1 \leq l \leq m^{d'}} \lambda_l^{d'} \leq \mathbf{W}^{d'}$. $\square$

Contrainte de respect de l'« aire » disponible pour les objets non encore placés

Supposons, de nouveau, que les k premières colonnes dans l'algorithme 3 sont fixés. La somme des volumes (ou aire en deux dimensions) de tous les objets non encore placés dans les premières colonnes ne doit pas excéder le volume du conteneur encore disponible. D'où, la contrainte suivante est valide (cf. figure 4.3).

Lemme 4.11

Pour tout $d \in \{1, \dots, D\}$ et $1 \leq k \leq m^d$,

$$\begin{aligned} \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{W}^{d'} \right) \left(\sum_{1 \leq l \leq k} \lambda_l^d \right) + \sum_{i \in Q_k^d \cap Q_{k+1}^d} \left(\left(\mathbf{w}_i^d - \sum_{\substack{1 \leq l \leq k \\ i \in Q_l^d}} \lambda_l^d \right) \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}_i^{d'} \right) \right) \\ + \sum_{i \notin \{Q_1^d, \dots, Q_k^d\}} \left(\prod_{d' \in \{1, \dots, D\}} \mathbf{w}_i^{d'} \right) \leq \prod_{d' \in \{1, \dots, D\}} \mathbf{W}^{d'}. \quad (4.7) \end{aligned}$$

Preuve. D'après le lemme 4.5, $\forall d \in \{1, \dots, D\}$ et $1 \leq k \leq m^d$, nous avons :

$$\begin{aligned} - \sum_{i \in Q_k^d \cap Q_{k+1}^d} \left(\mathbf{w}_i^d - \sum_{\substack{1 \leq l \leq k \\ i \in Q_l^d}} \lambda_l^d \right) \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}_i^{d'} \right) &\leq \sum_{i \in Q_k^d \cap Q_{k+1}^d} \left(\sum_{\substack{k+1 \leq l \leq m^d \\ i \in Q_l^d}} \lambda_l^d \right) \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}_i^{d'} \right), \\ - \sum_{i \notin \{Q_1^d, \dots, Q_k^d\}} \prod_{d' \in \{1, \dots, D\}} \mathbf{w}_i^{d'} &\leq \sum_{i \notin \{Q_1^d, \dots, Q_k^d\}} \left(\sum_{\substack{k+1 \leq l \leq m^d \\ i \in Q_l^d}} \lambda_l^d \right) \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}_i^{d'} \right). \end{aligned}$$

D'où,

$$\begin{aligned} \sum_{i \in Q_k^d \cap Q_{k+1}^d} \left(\mathbf{w}_i^d - \sum_{\substack{1 \leq l \leq k \\ i \in Q_l^d}} \lambda_l^d \right) \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}_i^{d'} \right) + \sum_{i \notin \{Q_1^d, \dots, Q_k^d\}} \left(\prod_{d' \in \{1, \dots, D\}} \mathbf{w}_i^{d'} \right) &\leq \\ \sum_{i \in \{Q_{k+1}^d, \dots, Q_{m^d}^d\}} \left(\sum_{\substack{k+1 \leq l \leq m^d \\ i \in Q_l^d}} \lambda_l^d \right) \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}_i^{d'} \right) &= \sum_{l \in \{k+1, \dots, m^d\}} \left(\lambda_l^d \left(\sum_{\substack{i \in Q_l^d \\ d' \in \{1, \dots, D\} \\ d' \neq d}} \prod_{d' \neq d} \mathbf{w}_i^{d'} \right) \right). \end{aligned}$$

D'après l'inégalité valide (4.6), nous obtenons :

$$\begin{aligned} \sum_{i \in Q_k^d \cap Q_{k+1}^d} \left(\mathbf{w}_i^d - \sum_{\substack{1 \leq l \leq k \\ i \in Q_l^d}} \lambda_l^d \right) \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}_i^{d'} \right) + \sum_{i \notin \{Q_1^d, \dots, Q_k^d\}} \left(\prod_{d' \in \{1, \dots, D\}} \mathbf{w}_i^{d'} \right) \\ \leq \left(\sum_{k+1 \leq l \leq m^d} \lambda_l^d \right) \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}^{d'} \right). \end{aligned}$$

D'où, l'inégalité (4.7) est valide. $\square$

En utilisant les lemmes 4.9, 4.10, 4.11, notre algorithme ne génère que des matrices réalisables et satisfaisants les contraintes (4.5), (4.6) et (4.7).

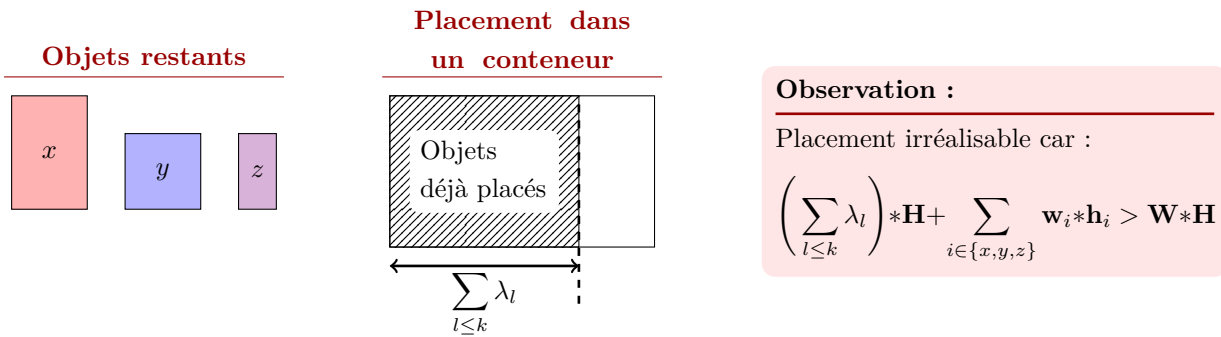


FIGURE 4.3: Exemple de placement irréalisable car la somme des aires des objets restants est trop volumineuse pour rentrer dans la partie disponible

4.2.3.2 Elimination de décompositions par bandes équivalentes

Un des avantages de l'approche de Fekete et Schepers est le fait que différents placements correspondaient à une même classe de placement. Cependant, par notre approche, cet avantage est perdu dans certains cas (cf figure 4.4).

La détection de configurations inutiles est impérative pour améliorer la performance et l'efficacité de notre algorithme pour éviter l'énumération de solutions symétriques.

Par exemple, de nombreuses symétries surviennent lorsque deux objets sont identiques (objets ayant la même taille sur toutes les dimensions) car les lignes correspondantes peuvent être librement permutés sur la première dimension. La première technique est donc de comparer les lignes correspondantes sur la première dimension pour placer en premier les objets de plus petit indice.

Cependant, des inconvénients liés à notre approche subsistent toujours. Nous avons donc défini des ordres lexicographiques pour remédier à ces problèmes.

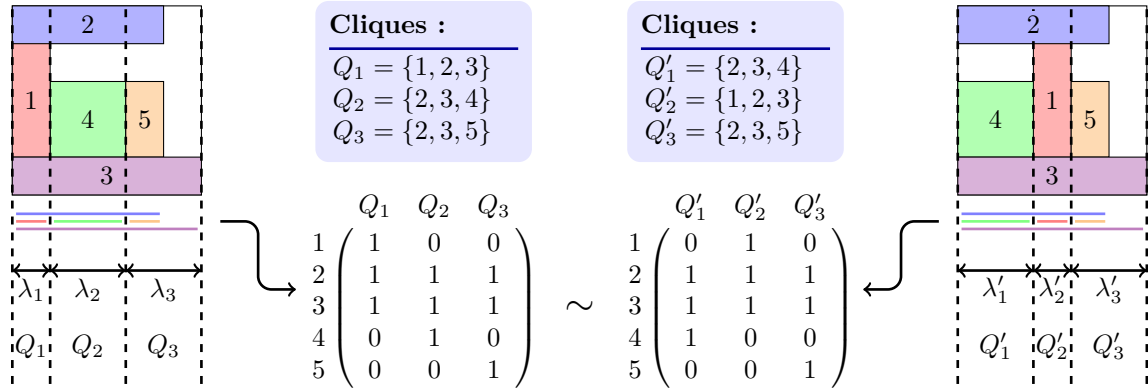


FIGURE 4.4: Deux décompositions par bandes similaires

Si $\mathbf{u}$ et $\mathbf{v}$ sont deux vecteurs binaires de même taille, nous définissons un ordre lexicographique $>_{lex}$ entre $\mathbf{u}$ et $\mathbf{v}$: $\mathbf{u} >_{lex} \mathbf{v}$ si $\mathbf{u}$ est lexicographiquement plus grand ou égale que $\mathbf{v}$.

Soit M et M' deux matrices dans $\mathcal{M}_{n,m}(\mathbb{B})$. $M >_{lex} M'$ s'il existe $l \in \{1, \dots, m\}$ tel que $\forall k < l, Q_k = Q'_k$ et $Q_l >_{lex} Q'_l$. Nous notons $M \sim M'$ si ces deux matrices ont les mêmes ensembles de vecteurs colonnes (M' peut être obtenu à partir de M par des permutations de colonnes). Dans ce cas, M et M' sont dites équivalentes.

Définition 4.12 (Domination)

Une décomposition par bandes est dominée par une autre si et seulement si, pour chaque dimension, la matrice associée de la première décomposition est similaire et lexicographiquement plus petite ou égale que la matrice associée de la seconde décomposition. ♦

Il suffit donc d'énumérer les matrices non dominées pour éviter des solutions symétriques. Cependant, il faut connaître, a priori, les matrices complètes pour pouvoir les comparer. Il paraît donc difficile d'utiliser une telle définition.

Cependant, nous remarquons que les premières matrices énumérées par l'algorithme 3 sont des matrices avec le plus grand ordre lexicographique. Donc, si la procédure d'énumération a été capable de détecter une matrice valide selon les contraintes d'une décomposition par bandes, alors, soit la solution avec le plus grand ordre lexicographique aura été trouvé et dans ce cas la procédure retourne vrai ; soit elle aura été déjà étudiée auparavant et dans ce cas la matrice devient invalide et la procédure peut retourner faux.

Exemple 4.13

Supposons que, durant la procédure, nous sommes à l'étape 2 comme indiqué ci-dessous (où $*$ et $-$ représentent respectivement la position courant et les positions non étudiées) :

$$\begin{array}{c|c} \begin{pmatrix} 1 & 0 & 0 & - \\ 0 & 1 & 0 & - \\ 0 & 0 & * & - \\ 0 & 0 & - & - \end{pmatrix} & \begin{pmatrix} 0 & 1 & 0 & - \\ 1 & 0 & 0 & - \\ 0 & 0 & * & - \\ 0 & 0 & - & - \end{pmatrix} \\ \text{Etape 1} & \text{Etape 2} \end{array}$$

Alors, l'étape 1 doit avoir été visitée avant par la procédure. Donc, la procédure peut arrêter l'énumération des matrices de même valeur sur les positions antérieures à la position courante de l'étape 2 si nous garantissons qu'aucune matrice valide non encore traitée est supprimée. $\blacklozenge$

Nous allons donc définir des structures permettant de trouver des configurations symétriques plus rapidement sans connaître, a fortiori, les matrices complètes.

Définition 4.14 (sous-matrice)

Soit $M \in \mathcal{M}_{n,m}(\{0,1\})$ une matrice binaire.

Pour tout $1 \leq p \leq m$ et $1 \leq q \leq m - p$, nous notons par $\text{SM}_{p,q}(M)$ la sous-matrice M' de M construite par les colonnes numérotées de p à $p + q - 1$, c'est-à-dire, pour tout $1 \leq i \leq n$ et $1 \leq k \leq q$, $M'_{i,k} = M_{i,k+p-1}$. $\blacklozenge$

Définition 4.15 (bloc)

Une sous matrice $\text{SM}_{p,q}(M)$ est un *bloc* de M , notée $B_{p,q}(M)$, si pour tout $1 \leq i \leq n$ et $p \leq k \leq p + q - 1$ avec $M_{i,k} = 1$, nous avons $M_{i,1} = M_{i,2} = \dots, M_{i,p-1} = 0$ et $M_{i,p+q} = M_{i,p+q+1} = \dots = M_{i,m} = 0$ (cf. figure 4.5). $\blacklozenge$

Pour une décomposition par bandes, nous interprétons la notion de bloc de la manière suivante : tout objet appartenant à au moins une clique du bloc ne peut pas appartenir aux cliques à l'extérieur du bloc.

$$\begin{pmatrix} 0 & \boxed{1 & 0} & 0 \\ 1 & \boxed{0 & 0} & 0 \\ 0 & \boxed{1 & 1} & 0 \\ 0 & \boxed{0 & 1} & 0 \\ 0 & \boxed{0 & 0} & 1 \\ 1 & \boxed{0 & 0} & 0 \end{pmatrix}$$

bloc

FIGURE 4.5: Exemple d'une matrice bloc $B_{2,2}(M)$

Lemme 4.16

Soit $(M^1, \dots, M^D)$ une décomposition par bandes.

S'il existe $d \in \{1, \dots, D\}$ tel que M^d a deux blocs $B_{p_1,q_1}(M^d)$ et $B_{p_2,q_2}(M^d)$ avec $p_1 + q_1 \leq p_2$ et $B_{p_2,q_2}(M^d) \underset{\text{lex}}{>} B_{p_1,q_1}(M^d)$; alors, il existe une matrice M'^d tel que $M'^d \underset{\text{lex}}{>} M^d$ et l'ensemble de matrices $(M^1, \dots, M'^d, \dots, M^D)$ est une décomposition par bandes équivalente (cf figure 4.6).

Preuve. Soit M'^d une matrice obtenue à partir de M^d en échangeant les deux blocs $B_{p_1,q_1}(M^d)$ et $B_{p_2,q_2}(M^d)$. Les colonnes C_k^d de M'^d sont données alors par :

$$C_k^{id} = \begin{cases} C_k^d & \text{si } k < p_1 \\ C_{k+p_2-p_1}^d & \text{si } p_1 \leq k < p_1 + q_2 \\ C_{k+q_1-q_2}^d & \text{si } p_1 + q_2 \leq k < p_2 + q_2 - q_1 \\ C_{k+p_1+q_1-p_2-q_2}^d & \text{si } p_2 + q_2 - q_1 \leq k < p_2 + q_2 \\ C_k^d & \text{si } p_2 + q_2 \leq k \end{cases}.$$

Il est alors facile de vérifier que $(M^1, \dots, M^{id}, \dots, M^D)$ est une décomposition par bandes équivalente à $(M^1, \dots, M^d, \dots, M^D)$ telle que $M^{id} \underset{lex}{>} M^d$ car les largeurs des bandes ne changent pas. $\square$

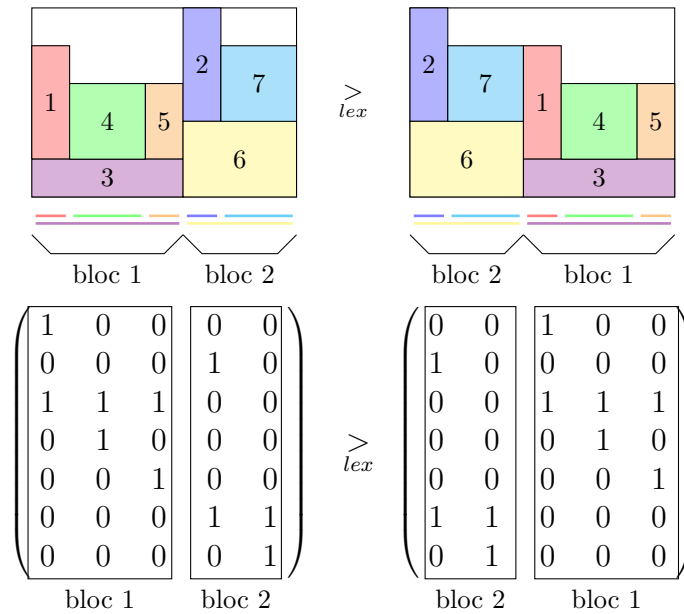


FIGURE 4.6: La permutation de deux blocs donne deux décompositions par bandes équivalentes

D'où, par le lemme 4.16, nous arrêtons la récursion pour certaines matrices quand il existe deux blocs qui ne sont pas lexicographiquement bien ordonnés (cf. le cas de droite dans la figure 4.6).

Nous remarquons aussi qu'il est possible d'inverser les colonnes d'un même bloc tel que la première colonne du bloc ait un plus grand ordre lexicographique que la dernière colonne du bloc (cf. figure 4.7). Cependant, nous allons généraliser la structure de bloc afin de trouver plus de décompositions par bandes symétriques.

Définition 4.17 (bloc interne)

Une sous matrice $SM_{p,q}(M)$ est un *bloc interne* de M , notée $IB_{p,q}(M)$, si pour tous indices $1 \leq i \leq n$ et $p \leq k \leq p + q - 1$ tels que $M_{i,k} = 1$, nous avons $(M_{i,p} = M_{i,p+1} = \dots = M_{i,p+q-1} = 1)$ ou $(M_{i,1} = M_{i,2} = \dots = M_{i,p-1} = 0$ et $M_{i,p+q} = M_{i,p+q+1} = \dots = M_{i,m} = 0)$ (cf. figure 4.8).

Un bloc interne $IB_{p,q}(M)$ est dit « *minimal* » si $Q_p \underset{lex}{>} Q_{p+q-1}$. $\blacklozenge$

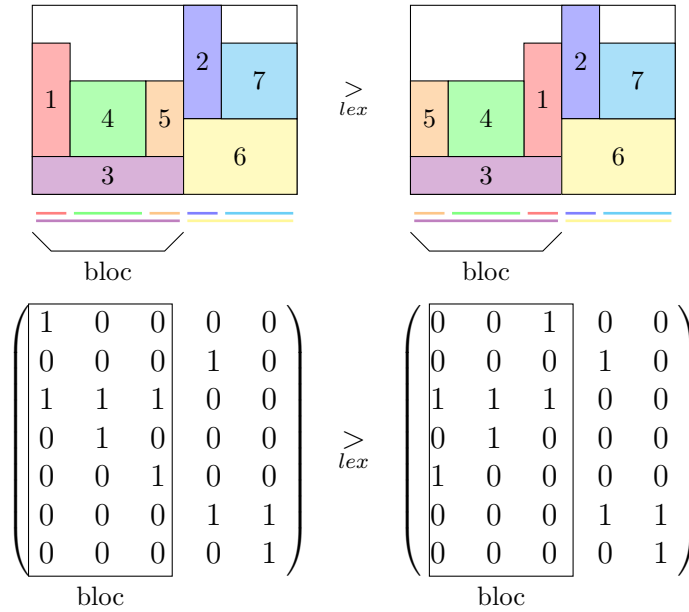


FIGURE 4.7: L'inversion des colonnes dans un bloc donne deux décompositions par bandes équivalentes

Pour une décomposition par bandes, nous interprétons la notion de bloc de la manière suivante : tout objet appartenant à au moins une clique du bloc interne doit soit appartenir à toutes les cliques du bloc interne, soit n'appartenir à aucune clique en dehors du bloc interne.

$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

bloc interne

FIGURE 4.8: Exemple d'une matrice bloc interne $IB_{2,2}$

Lemme 4.18

Soit $(M^1, \dots, M^D)$ une décomposition par bandes associée à l'ensemble d'objets $\mathcal{I}$. S'il existe $d \in \{1, \dots, D\}$ tel que M^d a un bloc interne $IB_{p,q}(M^d)$ non « minimal » ; alors il existe une matrice M'^d tel que $M'^d >_{lex} M^d$ et l'ensemble de matrices $(M^1, \dots, M'^d, \dots, M^D)$ est une autre décompositon par bandes équivalente associée à $\mathcal{I}$ (cf. figure 4.9).

Preuve. Soit $d \in \{1, \dots, D\}$ tel que M^d a un bloc interne $IB_{p,q}(M^d)$ non « minimal ». Pour tout $d' \in \{1, \dots, D\}$ et tout objet $i \in \mathcal{I}$, soit $f_i^{d'}$ (resp. $l_i^{d'}$) le plus petit (resp. grand) indice k avec $1 \leq k \leq m^{d'}$ tel que $M_{i,k}^{d'} = 1$. Et soit, $x_i^{d'} = \sum_{1 \leq k \leq f_i^{d'} - 1} \lambda_k^{d'}$.

D'après la preuve du lemme 4.8, $(x_i^1, \dots, x_i^D)_{i \in \mathcal{I}}$ est un placement réalisable. Nous allons

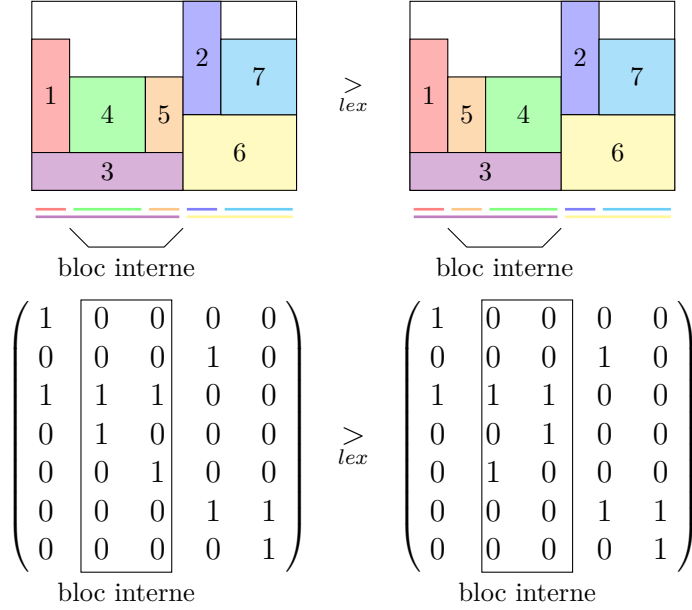


FIGURE 4.9: L'inversion des colonnes dans un bloc interne donne deux décompositions par bandes équivalentes

construire un autre placement réalisable $(x_i^1, \dots, x_i^D)_{i \in \mathcal{I}}$.

Soit $\alpha_{p,q}^d = \sum_{1 \leq k \leq p-1} \lambda_k^d$ et $\beta_{p,q}^d = \sum_{1 \leq k \leq p+q-1} \lambda_k^d$. Posons alors :

- $\forall i \in \mathcal{I}$ et $d' \in \{1, \dots, D\}$ avec $d' \neq d$, soit $x_i^{d'} = x_i^{d'}$.
- $\forall i \in \mathcal{I}$ tel que $f_i^d < p$ ou $l_i^d > p + q - 1$, soit $x_i^d = x_i^d = \sum_{1 \leq k \leq f_i-1} \lambda_k^d$.
- $\forall i \in \mathcal{I}$ tel que $f_i^d \geq p$ et $l_i^d \leq p + q - 1$, soit $x_i^d = \alpha_{p,q}^d + (\beta_{p,q}^d - \sum_{1 \leq k \leq l_i^d} \lambda_k^d)$.

D'où, nous avons :

$$x_i^d = \sum_{1 \leq k \leq p-1} \lambda_k^d + \sum_{l_i^d+1 \leq k \leq p+q-1} \lambda_k^d.$$

Montrons que la contrainte (4.1a) est satisfaite sur la dimension d pour le nouveau placement :

- $\forall i \in \mathcal{I}$ tel que $f_i^d < p$ ou $l_i^d > p + q - 1$, $x_i^d + \mathbf{w}_i^d \leq \mathbf{W}^d$.
- $\forall i \in \mathcal{I}$ tel que $f_i^d \geq p$ et $l_i^d \leq p + q - 1$, par le lemme (4.5), nous avons :

$$x_i^d + \mathbf{w}_i^d \leq \sum_{k \in \{1, \dots, p-1\}} \lambda_k^d + \sum_{k \in \{l_i^d+1, \dots, p+q-1\}} \lambda_k^d + \sum_{k \in \{f_i^d, \dots, l_i^d\}} \lambda_k^d \leq \mathbf{W}^d.$$

Montrons que la constraint (4.1b) est encore satisfaite pour le nouveau placement.

Si $i, j \in \mathcal{I}$ avec $i < j$ est tel que $[x_i^d, x_i^d + \mathbf{w}_i^d) \cap [x_j^d, x_j^d + \mathbf{w}_j^d) \neq \emptyset$, alors c'est évident. Supposons qu'il existe $i, j \in \mathcal{I}$ avec $i < j$ tel que $[x_i^d, x_i^d + \mathbf{w}_i^d) \cap [x_j^d, x_j^d + \mathbf{w}_j^d) = \emptyset$. Nous devons montrer que $[x_i^d, x_i^d + \mathbf{w}_i^d) \cap [x_j^d, x_j^d + \mathbf{w}_j^d) = \emptyset$, impliquant que la contrainte (4.1b) est satisfaite pour la paire (i, j) . Nous rencontrons quatre cas :

- Si $(f_i^d < p$ ou $l_i^d > p + q - 1)$ et $(f_j^d < p$ ou $l_j^d > p + q - 1)$, alors $x_i^d = x_i^d$ et $x_j^d = x_j^d$. D'où, $[x_i^d, x_i^d + \mathbf{w}_i^d) \cap [x_j^d, x_j^d + \mathbf{w}_j^d) = \emptyset$.

- Si $(f_i^d \geq p, l_i^d \leq p + q - 1)$ et $(f_j^d < p)$, alors $x_j^d + \mathbf{w}_j^d \leq x_i^d$. Par définition d'un bloc interne, nous avons $l_j^d < p$. D'où, par le lemme (4.5), nous avons :

$$x_j'^d + \mathbf{w}_j^d = x_j^d + \mathbf{w}_j^d \leq \sum_{1 \leq k \leq f_j^d - 1} \lambda_k^d + \sum_{f_j^d \leq k \leq l_j^d} \lambda_k^d \leq \sum_{1 \leq k \leq p-1} \lambda_k^d \leq x_i'^d.$$

- Si $(f_i^d \geq p, l_i^d \leq p + q - 1)$ et $(l_j^d > p + q - 1)$, alors $x_j^d \geq x_i^d + \mathbf{w}_i^d$. Par définition d'un bloc interne, nous avons $f_j^d > p + q - 1$. D'où, par le lemme (4.5), nous avons :

$$x_j'^d = x_j^d \geq \sum_{1 \leq k \leq f_j^d - 1} \lambda_k^d \geq \sum_{1 \leq k \leq p+q-1} \lambda_k^d \geq x_i'^d + \mathbf{w}_i^d.$$

- Si $(f_i^d \geq p, l_i^d \leq p + q - 1)$ et $(f_j^d \geq p, l_j^d \leq p + q - 1)$. Supposons que $x_j^d + \mathbf{w}_j^d \leq x_i^d$, alors $l_j^d < f_i^d$. D'où, par le lemme (4.5), nous avons :

$$\begin{aligned} x_i'^d + \mathbf{w}_i^d &\leq \sum_{1 \leq k \leq p-1} \lambda_k^d + \sum_{l_i^d + 1 \leq k \leq p+q-1} \lambda_k^d + \sum_{f_i^d \leq k \leq l_i^d} \lambda_k^d \\ &\leq \sum_{1 \leq k \leq p-1} \lambda_k^d + \sum_{l_j^d + 1 \leq k \leq p+q-1} \lambda_k^d \leq x_j'^d. \end{aligned}$$

Donc, $(x_i'^1, \dots, x_i'^D)_{i \in \mathcal{I}}$ est un placement réalisable. D'après le lemme 4.2, nous obtenons un placement réalisable $(x_i''^1, \dots, x_i''^D)_{i \in \mathcal{I}}$ avec la condition « calée à gauche », tel que $x_i^{d'} = x_i''^{d'}$ pour tout $d' \in \{1, \dots, D\}$ avec $d' \neq d$ et $i \in \mathcal{I}$.

Maintenant, nous devons montrer que la décomposition par bandes associée à ce nouveau placement $(x_i''^1, \dots, x_i''^D)_{i \in \mathcal{I}}$ (définie de la même manière que dans la preuve du lemme 4.7) a un plus grand ordre lexicographique sur la dimension d .

Nous avons donc une décomposition par bandes $(M^1, \dots, M''^d, \dots, M^D)$ en utilisant la preuve du lemme 4.7. Les $p - 1$ premières colonnes M''^d sont identiques M^d car les coordonnées des premiers objets sont identiques.

Par hypothèse, $Q_{p+q-1}^d >_{lex} Q_p^d$. Soit a (resp. b) le plus petit objet $i \in \mathcal{I}$ tel que $M_{i,p}^d = 1$ et $M_{i,p+q-1}^d = 0$ (resp. $M_{i,p}^d = 0$ et $M_{i,p+q-1}^d = 1$).

Par la définition du bloc interne, $x_b''^d = \sum_{1 \leq k \leq p-1} \lambda_k^d = x_a^d$.

Or, selon la condition « calée à gauche », il existe $j \in \mathcal{I}$ tel que $x_a^d = x_j^d + \mathbf{w}_j^d$.

Donc, $x_b''^d = x_j''^d + \mathbf{w}_j^d$ et l'objet b satisfait la condition « calée à gauche ».

D'où $b \in Q_p''^d$ et donc $Q_p''^d >_{lex} Q_p^d$ et $M''^d >_{lex} M^d$. □

D'où, par le lemme 4.16, nous arrêtons la récursion pour certaines matrices quand il existe un bloc interne non « minimal » (cf. le cas de droite dans la figure 4.9).

Ainsi, à partir de maintenant, toute matrice possédant soit un couple de bloc ne vérifiant pas le bon ordre lexicographique ; soit un bloc interne non « minimal » est considérée comme invalide.

Remarque Pour tout ensemble de colonnes $\mathcal{Q}$, une matrice $M(\mathcal{Q})$ tel que tous les couples de bloc vérifient l'ordre lexicographique et tous les blocs internes sont « minimaux » n'est pas forcément une matrice dominante selon la définition 4.12. Par exemple, les deux matrices suivantes sont des matrices valides mais la matrice de gauche est dominée par la

matrice de droite :

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Donc, des symétries ne sont pas éliminées par ces techniques. ♦

4.3 Approche par génération de MPQ-arbres

Une autre approche que nous avons expérimentée consiste à utiliser la caractéristion de Korte et Möhring rappelée en section 1.2.

4.3.1 L'algorithme de Korte et Möhring

Rappelons qu'un MPQ-arbre est un PQ-arbre étiqueté permettant de caractériser les graphes d'intervalles. Nous allons montrer comment un MPQ-arbre est construit pour un graphe d'intervalle. Ensuite, nous faisons le lien entre les MPQ-arbres et le problème de placement.

Nous commençons par rappeler l'algorithme de LexBFS, qui est un algorithme classique de parcours en largeur lexicographique des graphes (cf algorithme 4). Celui-ci est utilisé par la procédure de Korte et Möhring. L'ordre obtenu par l'algorithme 4 est dit ordre LexBFS et est noté $\sigma = [v_1, \dots, v_n]$. L'ordre inverse est noté $\sigma^R = [u_1, \dots, u_n] = [v_n, \dots, v_1]$. La figure 4.10 présente le déroulement de l'algorithme de LexBFS sur un exemple.

Algorithme 4 Algorithme de parcours LexBFS

Require: a graph $G = (V, E)$ and a vertex source $s \in V$

Ensure: an ordering σ of the vertices

LexBFS (graph G , integer s)

begin

 initialisation of the integer array $\sigma[n]$ to 0

$\text{label}(v) = \emptyset, \quad \forall v \in V$

 # label is a list of vertices for vertex v

$\text{label}(s) = \{1\}$

for all $i = 1$ to n **do**

 choose a vertex not visited $u \in V$ with the smallest non empty label

$\sigma[i] = u$

 # vertex u is now visited

for all $v \in V(u)$ not visited **do**

$\text{label}(v) = \text{label}(v) \cup \{i\}$

end for

end for

end

Remarque Si G est un graphe triangulé, alors le dernier sommet visité par l'algorithme de LexBFS est un sommet simplicial, c'est-à-dire que ce sommet et son voisinage forme une clique. L'ordre inversé obtenu par l'algorithme 4 définit alors un ordre dit d'élimination parfait, c'est-à-dire que chaque sommet u_i de l'ordre $\sigma^R = [u_1, \dots, u_n]$ est un sommet simplicial du sous-graphe induit par les sommets $\{u_i, \dots, u_n\}$.

Si G est un graphe d'intervalles, alors le dernier sommet visité par l'algorithme de LexBFS et son voisinage est une clique maximale pouvant être choisi comme départ d'un ordre de classement des cliques maximales. ♦

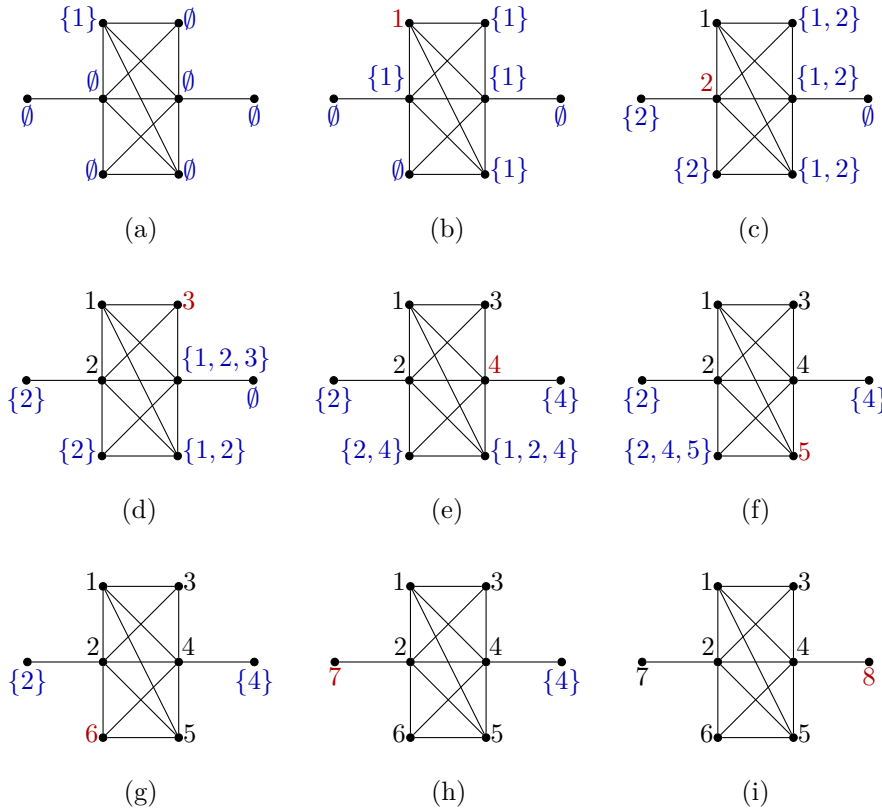


FIGURE 4.10: Déroulement de l'algorithme LexBFS sur un graphe G quelconque (en bleu : labels des sommets, en noir : numéro associé aux sommets)

L'algorithme de Korte et Möhring nécessite d'avoir en entrée un ordre LexBFS pour pouvoir fonctionner. En effet, ils utilisent le théorème suivant :

Théorème 4.19 (Korte et Möhring [50])

Soit $G = (V, E)$ un graphe avec un ordre LexBFS $\sigma = [v_1, \dots, v_n]$. Soit G_i le sous graphe induit par l'ensemble de sommets $V_i = \{v_1, \dots, v_i\}$. Alors, G est un graphe d'intervalles si et seulement si, pour chaque $1 \leq i \leq n$, le sous graphe G_i est un graphe d'intervalles et $V(v_i + 1) \cap V_i$ est une clique externe (première ou dernière clique dans un ordre de classement des cliques) de G_i .

Donc, en partant d'un ordre défini par l'algorithme LexBFS, leur procédure garantit

que le MPQ-arbre reste toujours valide, pour chaque nouveau sommet de cet ordre ajouté à l'arbre. A chaque étape, l'arbre va être modifié en conséquence. Supposons maintenant que chaque nouveau sommet est adjacent à au moins un sommet contenu dans le MPQ-arbre courant (sinon, la modification du MPQ-arbre est évidente). Ainsi, pour modifier l'arbre en fonction du nouveau sommet à rajouter, Korte et Möhring utilisent le théorème suivant :

Théorème 4.20 (Korte et Möhring [50])

Soit $G = (V, E)$ un graphe d'intervalles et soit T son MPQ-arbre associé. Alors, $G + u$ est un graphe d'intervalles si et seulement si les propriétés suivantes sont vraies :

1. Tous les sommets de G adjacent à u sont contenus dans une seule branche de T ,
2. Pour tout nœud N de la branche, de type Q et de sections $A_1, \dots, A_m$, les sommets de $V_N \cap V(u)$ sont contenus dans une section sur le bord : $V_N \cap V(u) \subseteq A_1$ ou $V_N \cap V(u) \subseteq A_m$.

Donc, seule une branche à l'extrémité de l'arbre (branche pouvant se situer sur le bord après application des règles de permutations définies en section 1.2) devra être modifiée car, à chaque nouveau sommet, une clique externe est ajoutée. Korte et Möhring donnent donc un nouvel algorithme incrémental (cf. [50] pour les détails les plus techniques) consistant à introduire pas à pas les sommets définis selon un ordre LexBFS. Pour chaque nouveau sommet u à rajouter, leur algorithme cherche un unique chemin à l'extrémité de l'arbre à modifier. S'il en trouve aucune, le graphe n'est pas d'intervalles. Sinon, ils appliquent la procédure suivante :

Procédure d'ajout d'un sommet au MPQ-arbre courant

- ① Chercher l'unique chemin à l'extrémité de l'arbre contenant tous les sommets adjacents à u et tel que :
 - ★ le nœud de la branche extrême de plus grande profondeur, noté N_* , contient au moins un sommet dans le voisinage de u : $\exists v \in N_*/v \in V(u)$,
 - ★ le nœud de la branche extrême de plus petite profondeur, noté N^* , ne contenant pas uniquement des objets dans le voisinage de u . Sinon, poser $N^* = N_*$:

$$\{N^* \setminus V(u) \neq \emptyset\} \text{ ou } \{N^* = N_*\}.$$

Donc ce chemin est contenu dans une branche externe.

- ② Parcourir le chemin de N_* à N^* et modifier chaque nœud en en appliquant le patron correspondant parmi les patrons définis par les figures 4.11, 4.12 et 4.13. Pour tout nœud N , soit $V_N = A \cup B$ la partition de sommets de V_N tel que $A = V_N \cap V(u)$ et $B = V_N \setminus V(u)$. Si aucun patron ne peut être sélectionné, alors il n'existe pas de branches externes contenant tous les sommets et le graphe n'est pas d'intervalle.

4.3.2 Application aux problèmes de placement

Dans le cas des placements, les sommets sont représentés par les objets. Alors, un MPQ-arbre T^d va être associé à chaque dimension $d \in \{1, \dots, D\}$. A chaque nœud N du MPQ-arbre T^d , nous associons une largeur λ_N^d définie selon la règle suivante :

- Si N est une feuille, posons $\lambda_N^d = \max_{i \in V_N} \{\mathbf{w}_i^d\}$ si $V_N \neq \emptyset$. Sinon, posons $\lambda_N^d = 0$.

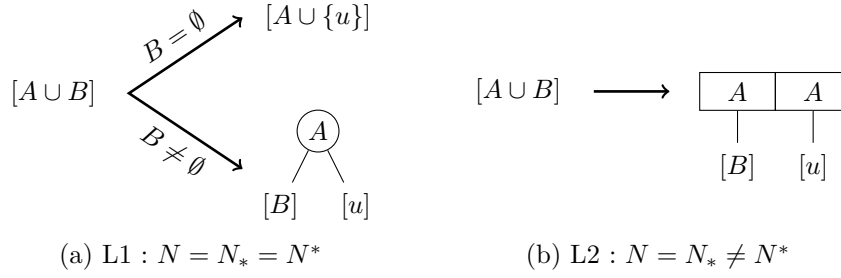
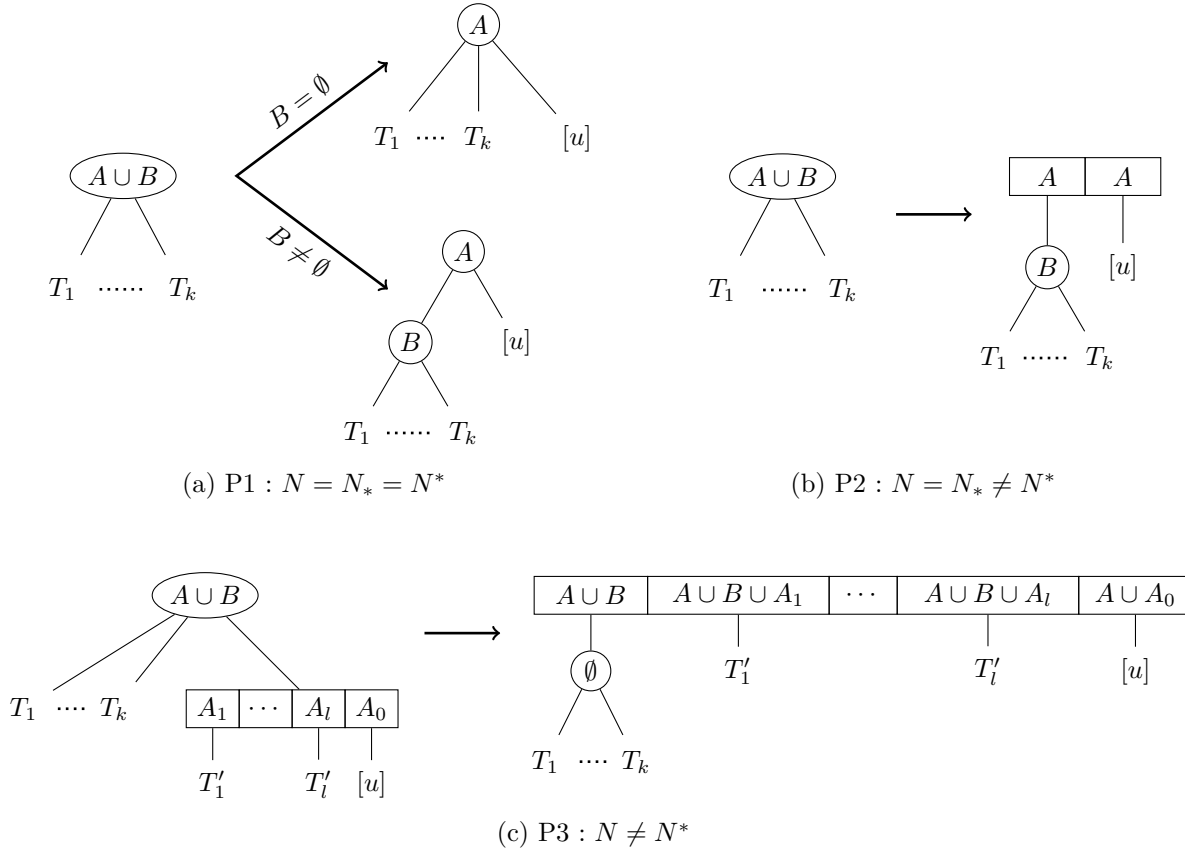


FIGURE 4.11: Patrons pour une feuille

FIGURE 4.12: Patrons pour un nœud de type P

- Si N est un nœud de type P , soit $\lambda_{T_1}^d, \dots, \lambda_{T_m}^d$ les largeurs de chacun de ces fils $T_1, \dots, T_m$. Posons alors :

$$\lambda_N^d = \max \left\{ \sum_{k=1}^m \lambda_{T_k}^d; (\mathbf{w}_i^d)_{i \in V_N} \right\}$$

- Si N est un nœud de type Q de sections $A_1, \dots, A_m$, soit $\lambda_{T_k}^d$ les largeurs de chaque nœud fils T_k correspondant à la section A_k , pour tout $1 \leq k \leq m$. La largeur de chaque section A_k , notée $\lambda_{A_k}^d$, est donnée par :

$$\lambda_{A_k}^d = \max \left\{ 0; \lambda_{T_k}^d; \left(\mathbf{w}_i^d - \sum_{l < k: i \in A_l} \lambda_{A_l}^d \right)_{i \in A_k, i \notin A_{k+1}} \right\}.$$

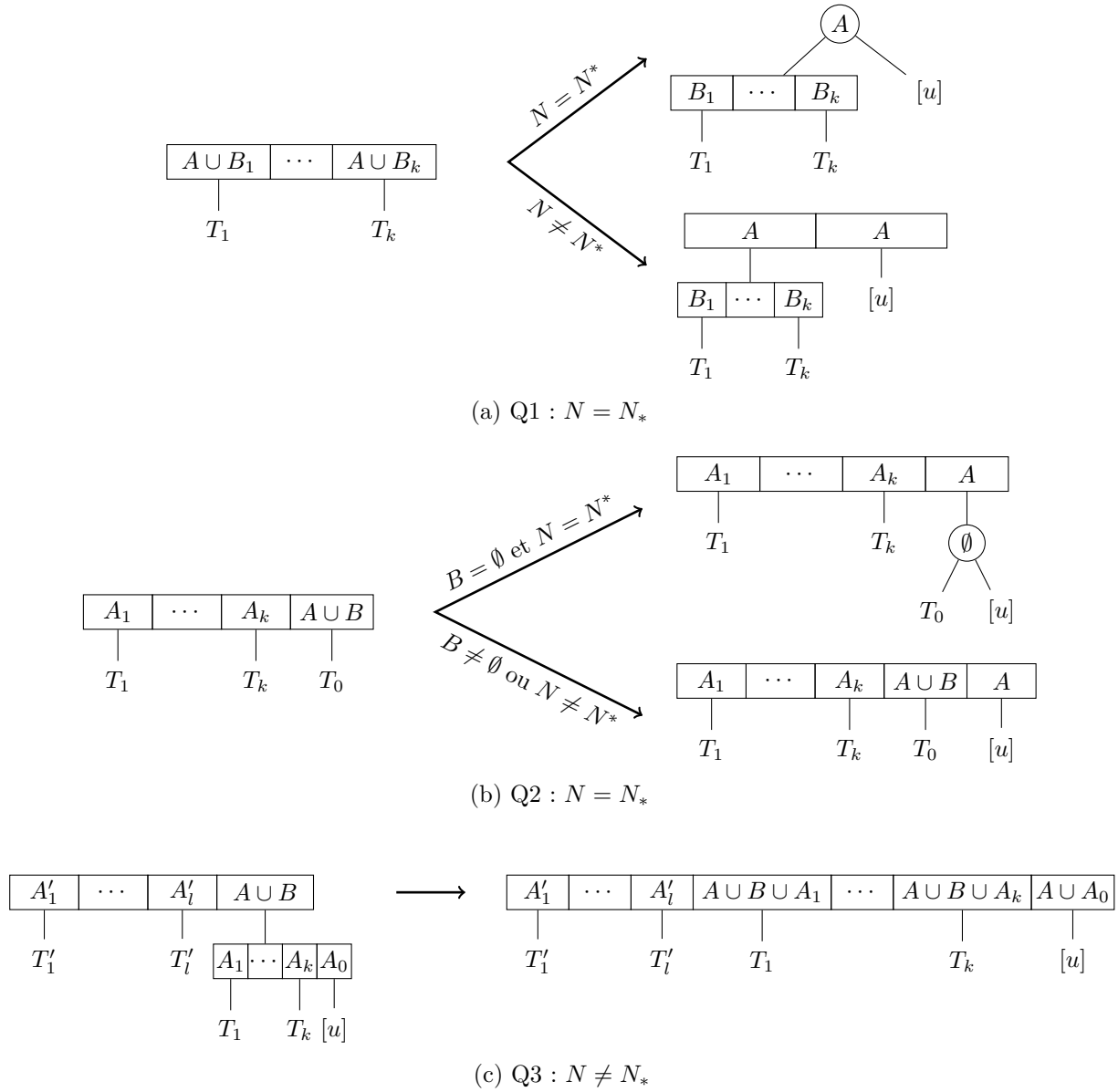


FIGURE 4.13: Patrons pour un nœud de type Q

Posons alors $\lambda_N^d = \sum_{k=1}^m \lambda_{A_k}^d$.

Remarque A ce stade, plusieurs remarques peuvent être faites. D'abord, le calcul de la largeur d'un nœud de type Q se fait en deux étapes :

- La première consiste à calculer les largeurs de chaque section. Chacune d'elle est calculée en fonction des sections précédemment calculées. Ainsi, ce calcul se fait de la section A_1 (située à gauche) à la section A_m (située à droite).
- Une fois toutes les largeurs de section déterminées, le calcul de la largeur du nœud de type Q est immédiat.

Nous remarquons aussi que, pour chaque nœud N avec des fils $T_1, \dots, T_m$, $\lambda_N^d \geq \sum_{k=1}^m \lambda_{T_k}^d$, pour tout $d \in \{1, \dots, D\}$. Enfin, nous remarquons que chaque largeur est positive. ♦

Lemme 4.21

Pour tout dimension $d \in \{1, \dots, D\}$, soit T^d un MPQ-arbre en dimension d . Soit i un objet de $\mathcal{I}$. Les propriétés suivantes sont vraies :

- Pour chaque nœud N feuilles ou de type P tel que i appartient à N , nous avons $\mathbf{w}_i^d \leq \lambda_N^d$.
- Pour chaque nœud N de type Q contenant i dans certaines sections de N , nous avons $\mathbf{w}_i^d \leq \sum_{h=k}^l \lambda_{A_h}^d$, où $A_k, \dots, A_l$ sont les sections de N contenant i dans leur étiquette.

Preuve. Le cas des nœuds feuilles ou de type P sont triviaux.

Supposons que N est un nœud de type Q . Soit $A_k, \dots, A_l$ les sections de N contenant i .

Donc, par définition, nous avons, $\lambda_{A_i}^d \geq \mathbf{w}_i^d - \sum_{h=k}^{l-1} \lambda_{A_h}^d$. La propriété est vraie. $\square$

Remarquons que la propriété (4.2c) du théorème 4.3 peut être facilement traduite dans les MPQ-arbres. En effet, chaque branche de l'arbre représente les cliques maximales du graphe d'intervalles associé. Ils nous suffit donc de transposer la propriété (4.2b) :

Lemme 4.22

Pour tout $d \in \{1, \dots, D\}$, soit $G_d = (\mathcal{I}, E_d)$ un graphe d'intervalle et T^d son MPQ-arbre associé. Si G_d satisfait la propriété (4.2b); alors la largeur du nœud racine r de T^d , noté λ_r^d , vérifie $\lambda_r^d \leq \mathbf{W}^d$.

Preuve. Soit $d \in \{1, \dots, D\}$. La preuve se fait par récurrence sur la distance des nœuds à la racine. Soit $R(k)$ la propriété de récurrence suivante :

« Si le nœud N de T à la distance k de la racine; alors il existe un ensemble stable S_N du graphe induit par les sommets contenus dans le sous-arbre induit enraciné en N (noté $G[N]$) tel que $\lambda_N \leq \sum_{i \in S_N} \mathbf{w}_i^d$. »

- Soit Φ la plus grande distance à la racine r dans T . Pour chaque nœud N de T à la distance Φ de r . Donc, N est une feuille. Par définition d'une feuille, il existe $i \in V_N$ tel que $\mathbf{w}_i^d = \lambda_N$. D'où, $S_N = \{i\}$ et $R(\Phi)$ est vraie.
- Supposons que $R(k)$ est vraie (hypothèse de récurrence). Montrons que chaque nœud N de T à la distance $k - 1$ de r vérifie la propriété de récurrence.

. Si N est une feuille alors l'argument précédent s'applique de nouveau.

. Si N est un nœud de type P avec m fils $T_1, \dots, T_m$, nous avons alors deux cas :

Si $\max_{i \in V_N} \mathbf{w}_i^d \leq \sum_{k=1}^m \lambda_{T_k}^d$ alors posons $S_N = \cup_{k=1}^m S_{T_k}$ où S_{T_k} est le stable associé à chacun des fils T_k , par hypothèse de récurrence. Comme dans le graphe G_d , il n'existe pas d'arêtes entre deux sommets contenus dans deux fils disjoints de N , S_N est donc un ensemble stable de $G[N]$. Donc, en appliquant la condition de récurrence, $\lambda_N^d \leq \sum_{k=1}^m \lambda_{T_k}^d \leq \sum_{i \in S_N} \mathbf{w}_i^d$. La propriété $R(k - 1)$ est vraie pour ce cas-ci.

Si $\max_{i \in V_N} \mathbf{w}_i^d > \sum_{k=1}^m \lambda_{T_k}^d$ alors posons $S_N = \operatorname{argmax}_{i \in V_N} \mathbf{w}_i^d$. D'où, $\lambda_N^d = \sum_{i \in S_N} \mathbf{w}_i^d$ et la propriété $R(k - 1)$ est vraie pour ce cas-ci.

. Si N est un nœud de type Q de sections $A_1, \dots, A_m$. Soit $T_1, \dots, T_m$ les fils des sections $A_1, \dots, A_m$. Par hypothèse de récurrence, S_{T_k} désigne le stable associé à chacun des fils T_k , pour tout $1 \leq k \leq m$. Nous allons regrouper les sections par lots tel que chacun des lots est indexé, noté X_l , et un ensemble stable est associé à chaque lot. La construction des lots se fait par un parcours de droite à gauche des sections du nœud N selon la procédure suivante :

Procédure de construction d'un stable

- ① Initialisation des indices : $k = m, l = 1$
- ② Tant que $k \geq 1$ faire
 - ★ Si $(\lambda_{A_k}^d = \lambda_{T_k}^d)$ ou $(\lambda_{A_k}^d = 0)$,
Alors poser $X_l = \{A_k\}$, $S_{X_l} = S_{T_k}$, $\lambda_{X_l} = \lambda_{A_k}^d \leq \sum_{i \in S_{T_k}} \mathbf{w}_i^d$,
 $k = k - 1$.
 - ★ Sinon, soit $i^* = \operatorname{argmax}_{i \in A_k, i \notin A_{k+1}} \left\{ \mathbf{w}_i^d - \sum_{h < k, i \in A_h} \lambda_{A_h}^d \right\}$,
Alors poser $X_l = \{A_h : i^* \in A_h\}$, $S_{X_l} = \{i^*\}$, $\lambda_{X_l} = \sum_{h \leq k : i^* \in A_h} \lambda_{A_h}^d = \mathbf{w}_{i^*}^d$,
 $k = \min_{1 \leq h \leq k} \{h : i^* \in A_h\} - 1$.
 - ★ Poser $l = l + 1$.

Soit α le nombre de lots de sections créés. A chaque lot X_l , un ensemble stable, noté S_{X_l} est aussi construit. De même, une largeur $\lambda_{X_l}^d$ est aussi associé. Posons $S_N = \cup_{l=1}^{\alpha} S_{X_l}$. Nous allons montrer que S_N est un ensemble valide pour prouver la véracité de $R(k-1)$.

Par construction, $\cap_{l=1}^{\alpha} X_l = \emptyset$. Donc, comme chaque S_{X_l} est stable, S_N est stable.

De plus, $\cup_{l=1}^{\alpha} X_l = \{A_1, \dots, A_n\}$, donc, $\lambda_N^d = \sum_{k=1}^m \lambda_{A_k}^d = \sum_{l=1}^{\alpha} \lambda_{X_l}^d$. Or, pour tout

$1 \leq l \leq \alpha$, $\lambda_{X_l} \leq \sum_{i \in S_{X_l}} \mathbf{w}_i^d$. Donc, $\lambda_N^d \leq \sum_{i \in S_N} \mathbf{w}_i^d$.

D'où, $R(k-1)$ est vraie quelque soit le type de nœud.

Par récurrence, $R(0)$ est vraie, et donc $\lambda_r^d \leq \sum_{i \in S_r} \mathbf{w}_i^d \leq \mathbf{W}^d$. □

La réciproque est vraie :

Lemme 4.23

Pour tout $d \in \{1, \dots, D\}$, T^d un MPQ-arbre et G_d son graphe d'intervalles associé. Si la largeur du nœud racine r de T^d , noté λ_r^d , vérifie $\lambda_r^d \leq \mathbf{W}^d$; alors G_d satisfait la propriété (4.2b).

Preuve. Soit $d \in \{1, \dots, D\}$. Supposons que $\lambda_r^d \leq \mathbf{W}^d$. Soit un ensemble stable S de G_d . Par définition d'un MPQ-tree, $\forall i \in S$, il existe dans T^d deux cas :

- Soit i appartiennent à un nœud N soit de type feuille soit de type P . Donc, posons $\Lambda_i = \lambda_N^d$.
- Soit i appartient à des sections d'un nœud N de type Q . Soit $A_k, \dots, A_l$ les sections du nœud N contenant i . Posons alors $\Lambda_i = \sum_{h=k}^l \lambda_{S_h}^d$.

Comme S est un ensemble stable, les objets de S n'appartiennent pas aux mêmes branches dans l'arbre. Ainsi, nous en déduisons que $\sum_{i \in S} \Lambda_i \leq \lambda_r^d$. Par le lemme 4.21, nous avons $\mathbf{w}_i^d \leq \Lambda_i$, pour tout $\forall i \in S$. D'où, $\sum_{i \in S} \mathbf{w}_i^d \leq \lambda_r^d \leq \mathbf{W}^d$. □

Nous avons donc un lien évident entre les classes de placement et le placement par MPQ-arbre. Notre algorithme doit donc énumérer les ensembles de D MPQ-arbres vérifiant les conditions de la définition 4.24.

Définition 4.24 (Placement par MPQ-arbres)

Un placement par MPQ-arbres associée à l'ensemble $\mathcal{I}$ est un ensemble de D MPQ-arbre $(T^1, \dots, T^D)$ avec les propriétés suivantes :

- Pour chaque dimension $d \in \{1, \dots, D\}$, le nœud racine r de T^d vérifie $\lambda_r^d \leq \mathbf{W}^d$.
- Pour chaque paire d'objets $i, j \in \mathcal{I}$ avec $i < j$, il existe une dimension $d \in \{1, \dots, D\}$ telle que i et j ne partagent pas de branche commune dans T^d .

◆

Remarque Les cas dégénérés liés à l'approche de Fekete et Schepers existent toujours dans la définition d'un placement par les MPQ-arbres. ◆

4.3.3 Le cœur de l'algorithme de contrôle de la réalisabilité d'un ensemble d'objets

A partir des lemmes 4.22 et 4.23, nous concevons un algorithme pour contrôler la réalisabilité d'un ensemble d'objets $\mathcal{I}$.

Contrairement à l'algorithme de Korte et Möhring qui nécessite de connaître un ordre LexBFS valide pour un graphe d'intervalles donné, nous ne connaissons pas a priori d'ordre valide. La première étape consiste donc à énumérer, pour chaque dimension, l'ensemble des ordres LexBFS possibles afin de pouvoir commencer la génération de MPQ-arbre valide (cf. algorithme 5 en partant de `position=1`). Une fois l'ordre généré, nous générerons l'ensemble des MPQ-arbres pouvant être trouvé à partir de cet ordre en appliquant les patrons définis précédemment.

Cependant, plusieurs ordres valides existent pour un même MPQ-arbre. Donc, nous risquons d'énumérer des MPQ-arbres ayant un ordonnancement identique associé à un même MPQ-arbre. Or, l'ordonnancement d'un MPQ-arbre est un ordre LexBFS valide pour un graphe d'intervalles donné. En effet, cet ordonnancement peut être obtenu en partant de son premier sommet. Donc, nous ne considérons que les ordres correspondant au même ordonnancement final dans l'arbre. Il s'agit donc d'ordres valides suffisant. Ce constat permet de déduire que seule la branche la plus à droite du MPQ-arbre est modifiée contrairement à l'algorithme originel où des permutations entre les branches étaient possibles. Donc, à chaque ajout d'un nouvel objet, il est suffisant d'appliquer strictement les patrons sur des chemins situés sur la branche de droite sans effectuer de rotation au préalable entre les branches.

Ainsi, pour une dimension $d \in \{1, \dots, D\}$ et un ordre d'introduction de objets σ donnés, l'algorithme 6 construit de manière récursive tous les MPQ-arbres d'ordonnancement identique à σ et vérifiant la contrainte de non dépassement du conteneur au nœud racine. Pour chaque objet rajouté selon l'ordre σ , nous sélectionnons un chemin modifiant parmi l'ensemble des chemins pouvant être modifié avec $N_* = \text{node}$, le nœud de référence. S'il existe un chemin encore non visité, nous passons à l'objet suivant (de façon récursive) avec pour nœud de référence la feuille nouvellement créée. S'il en existe plus, nous prenons comme nœud de référence le père du nœud actuel.

Au final, ces algorithmes permettent de générer l'ensemble des MPQ-arbres possibles

Algorithme 5 Algorithme de génération d'ordre valide

Require: an instance I and an order σ

Ensure: search all possible orders of items

OrderGen (integer position, integer $\sigma[n]$, instance I)

```

begin
  if position > n then
    initialisation of the MPQ-tree T
    MPQtreeRecurse( $\emptyset, 1, \sigma, T, I$ )  # call function to search a valid MPQ-tree for  $\sigma$ 
    if T is a valid MPQ-tree then
      return true
    else
      return false
    end if
  end if
  for i=1 to n do
    j=1
    while (j < position and not ( $\sigma[j]=i$ )) do
      j=j+1
    end while
    if (j=position) then
       $\sigma[\text{position}] = i$ 
      OrderGen(position+1,  $\sigma, I$ )
    end if
  end for
end

```

pour un ensemble d'objets à une dimension donnée. Il est possible d'appeler ces algorithmes dans l'algorithme 2 en remplaçant la fonction **recurse** par la fonction **OrderGen**.

4.3.4 Améliorations des performances de l'algorithme

Les cas de dégénérescence liés à l'approche de Fekete et Schepers évoqués précédemment sont encore présents dans l'approche développée. Ils peuvent être éliminés par un contrôle similaire à l'approche des matrices de uns-consécutifs. De plus, comme évoqué précédemment, la structure d'un MPQ-arbre produit une information partielle sur les positionnement des objets les uns par rapport aux autres. En effet, l'ordonnancement des objets associé au MPQ-arbre donne un ordre d'apparition de gauche à droite des objets dans le placement. Ce constat nous permet d'adapter des contraintes similaires à l'approche par les matrices de uns-consécutifs pour détecter une irréalabilité au plus tôt. Nous définissons aussi des ordres lexicographiques pour éviter d'énumérer des MPQ-arbres équivalents.

4.3.4.1 Elimination des cas dégénérés

Nous allons commencer par éliminer les cas de placements dégénérés issus de la caractérisation par les graphes d'intervalles d'un placement.

Algorithme 6 Fonction récursive de construction et de contrôle de la réalisabilité d'un MPQ-tree

Require: an instance I , an order σ , a MPQ-tree T , a dimension d

Ensure: return true if there exists a feasible packing, false otherwise

```

recurse(MPQ-node node, integer vertex, integer d, integer  $\sigma[n]$ , MPQ-tree T, instance I)
  begin
    if vertex > n then
      return true
    end if
    repeat
      for all corresponding MPQ-patterns to node for  $\sigma[\text{vertex}]$  do
        # it exists at less one modifying path to transform the MPQ-tree with  $N_* = \text{node}$ 
        apply the modification with the vertex  $\sigma[\text{vertex}]$ 
        node = new created leaf L contained  $\sigma[\text{vertex}]$ 
        for all  $N$  in leftmost branch of T do
          update  $\lambda_N^d$ 
        end for
        if  $\lambda_r^d \leq W^d$  then
          if recurse(node, vertex+1, d,  $\sigma$ , T, I) then
            return true
          end if
        end if
        undo the modification
      end for
      node = father of node
    until node  $\neq$  NULL
    return false
  end

```

Lemme 4.25

Soit T^d un MPQ-arbre valide sur la dimension $d \in \{1, \dots, D\}$ associé à un placement. Pour chaque nœud interne N de l'arbre avec m fils $T_1, \dots, T_m$, nous imposons les inégalités suivantes :

- Si N est un nœud de type P , $\forall i \in V_N$, alors $w_i^d - \sum_{k=1}^{m-1} \lambda_{T_k}^d > 0$,
- Si N est un nœud de type Q de sections $A_1, \dots, A_m$, alors pour tout $1 \leq k \leq m$ et $i \in A_k$, $w_i^d - \sum_{h < k: i \in A_h} \lambda_{A_h}^d > 0$.

Preuve. Soit $d \in \{1, \dots, D\}$. D'après la définition des largeurs de nœud, chaque nœud ou section X du MPQ-arbre T^d vérifie $\lambda_X^d \geq 0$.

Supposons qu'il existe au moins un nœud N dans l'arbre T^d contenant un objet $i \in V_N$ ne satisfaisant pas les inégalités du lemme :

- Si N est un nœud P alors $w_i^d - \sum_{k=1}^{m-1} \lambda_{T_k}^d \leq 0$. Soit $k^* = \min\{k : w_i^d \leq \sum_{h=1}^k \lambda_{T_h}^d\}$. Nous construisons un nouveau MPQ-arbre T'^d en appliquant le patron de remplacement de la figure 4.14, où $B = \{j \in V_N : w_j^d \leq \sum_{h=1}^{k^*} \lambda_{T_h}^d\}$ est l'ensemble des objets dégénérés en position k^* et $A = V_N \setminus B$ est les objets non dégénérés en position k^* présent dans le nœud N . D'où, $i \in B$. Notons $\lambda_X'^d$ les nouvelles largeurs asso-

ciées au nœud ou à la section X de l'arbre T'^d . Pour ce nouvel arbre T'^d , nous avons $\lambda_B'^d = \sum_{h=1}^{k^*} \lambda_{T_h}^d$ et $\sum_{h=1}^m \lambda_{T_h}^d = \lambda_B'^d + \sum_{h=k^*+1}^m \lambda_{T_h}^d$. D'où $\lambda_A'^d = \lambda_N^d$. Enfin, $w_i^d - \sum_{h=1}^{k^*-1} \lambda_{T_h}^d > 0$. Nous avons donc construit un arbre ne changeant pas la largeur globale du placement et éliminant la dégénérescence de l'objet i sur un nœud P .

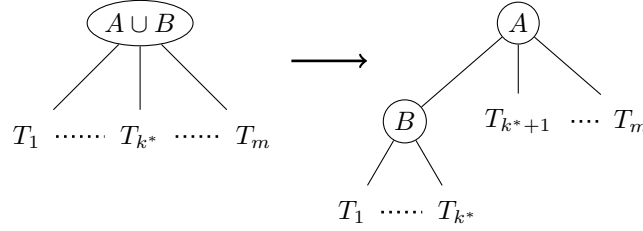


FIGURE 4.14: Patron de remplacement pour un nœud de type P possédant un objet dégénéré

- Si N est un nœud Q de sections $A_1, \dots, A_m$ alors il existe $1 \leq k^*, l^* \leq m$ tel que $k^* = \min\{k : i \in A_k \text{ et } w_i^d \leq \sum_{h < k: i \in A_h} \lambda_{A_h}^d\}$ et $l^* = \max\{l : i \in A_l\}$. D'où, nous avons $k^* \leq l^*$ et pour tout $k^* \leq h \leq l^*$ et $w_i^d - \sum_{h' < h: i \in A_{h'}} \lambda_{A_{h'}}^d \leq 0$. Nous construisons

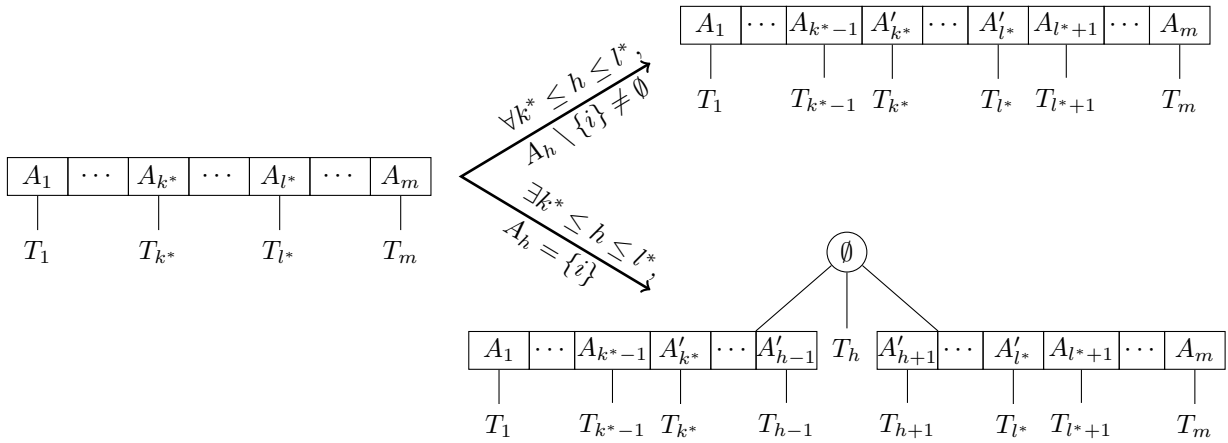


FIGURE 4.15: Patron de remplacement pour un nœud de type Q possédant un objet dégénéré

un nouveau MPQ-arbre T'^d en appliquant le patron de remplacement de la figure 4.15, où pour tout $k^* \leq h \leq l^*$, nous posons $A'_h = A_h \setminus \{i\}$. Notons $\lambda_X'^d$ les nouvelles largeurs associées au nœud ou à la section X de l'arbre T'^d . Si nous avons pour tout $k^* \leq h \leq l^*$, $A_h \setminus \{i\} \neq \emptyset$ (cas du dessus dans la figure 4.15), alors nous avons comme largeurs : pour tout $1 \leq h < k^* - 1$, $\lambda_{A_h}^d = \lambda_{A_h}^d$. Comme dans T^d , $w_i^d - \sum_{h < k^*: i \in A_h} \lambda_{A_h}^d \leq 0$, nous avons $\lambda_{A_{k^*-1}}^d = \lambda_{A_{k^*-1}}^d$ et pour tout $k^* \leq h \leq l^*$, $\lambda_{A_h \setminus \{i\}}^d = \lambda_{A_h}^d$. Et donc pour tout $l^* < h \leq m$, $\lambda_{A_h}^d = \lambda_{A_h}^d$. D'où, $\lambda_N'^d = \lambda_N^d$. S'il existe au moins $k^* \leq h \leq l^*$ tel que $A_h = \{i\}$ (cas du dessous dans la figure 4.15). Alors posons $k^* \leq h_1 < \dots < h_\alpha \leq l^*$ l'ensemble des indices des sections tels que $A_h = \{i\}$. Soit N_0 le nouveau nœud de type P servant de père et soit $N_1, \dots, N_{\alpha+1}$ les nœud fils de type Q engendrés par le patron (Chaque ensemble $A_h = \{i\}$ engendre

de nouveaux fils). Nous avons les largeurs suivantes : En utilisant des arguments similaires aux arguments précédents, nous avons pour tout $1 \leq h < k^*$, $\lambda_{A_h}^d = \lambda_{A_h}^d$. De même, pour tout $k^* \leq h < h_1$, $\lambda_{A_h \setminus \{i\}}^d = \lambda_{A_h}^d$. Comme $S_{h_1} = \{i\}$, son fils T_{h_1} n'est pas vide sinon le MPQ-arbre contient une clique non maximale (impossible par la méthode de construction de Korte et Möhring). Donc, $\lambda_{T_{h_1}}^d = \lambda_{T_{h_1}}^d = \lambda_{A_{h_1}}^d > 0$. En itérant de la même manière, nous montrons que chaque fils de N_0 et en utilisant les mêmes arguments, nous démontrons que les largeurs des feuilles ou des sections des nœuds de type Q correspond à une unique largeur d'une section dans l'arbre d'origine. Donc, au final, nous avons $\lambda_{N_0}^d = \sum_{p=1}^{\alpha+1} \lambda_{N_p}^d + \sum_{p=1}^{\alpha} \lambda_{T_{h_p}}^d = \lambda_N^d$. Nous avons donc construit un arbre ne changeant pas la largeur globale du placement et éliminant la dégénérescence de l'objet i sur un nœud Q . Remarquons aussi que dans les deux cas, il n'existe pas d'objet satisfaisant la condition du lemme dans T^d et ne la satisfaisant pas dans T^d .

Nous appliquons itérativement ces procédures (patrons pour les nœuds de type P ou Q) sur les objets ne respectant pas les inégalités du lemme pour montrer qu'il existe un arbre valide T^d sans objets dégénérés. $\square$

4.3.4.2 Ajout d'inégalités valides pour détecter des irréalismes au plus tôt

Soit $d \in \{1, \dots, D\}$ et T^d le MPQ-arbre en construction. Soit $T_1, \dots, T_m$ les m fils du nœud racine. Les inégalités définies pour la caractérisation des placements par des matrices de uns-consécutifs sont encore valides pour les MPQ-arbres. En effet, à l'étape i de l'algorithme 6, le MPQ-arbre construit contient les objets $\sigma(1), \dots, \sigma(i)$. Après l'étape i , les largeurs des fils $T_1, \dots, T_{m-1}$ ne peuvent plus être modifiées comme seule la branche la plus à droite est changée au cours du processus d'énumération. De plus, chaque branche représente une clique maximale pour le graphe d'intervalles associé. Les inégalités suivantes sont donc vraies, pour tout $d \in \{1, \dots, D\}$:

$$\sum_{1 \leq k \leq m-1} \lambda_{T_k}^d + \max_{j \in \{i+1, \dots, n\}} \mathbf{w}_{\sigma(j)}^d \leq \mathbf{W}^d, \quad (4.8a)$$

$$\sum_{i \in B} \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}_{\sigma(i)}^{d'} \right) \leq \prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{W}^{d'}, \quad \forall B \in \mathcal{B} \quad (4.8b)$$

$$\begin{aligned} & \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{W}^{d'} \right) \left(\sum_{1 \leq k \leq m-1} \lambda_{T_k}^{d'} \right) + \sum_{j \in \{i+1, \dots, n\}} \left(\prod_{d' \in \{1, \dots, D\}} \mathbf{w}_{\sigma(j)}^{d'} \right) \\ & + \sum_{j \in N_r} \left(\left(\mathbf{w}_{\sigma(i)}^d - \sum_{\substack{1 \leq k \leq m-1: \\ j \in \text{père}(T_k)}} \lambda_{T_k}^d \right) \left(\prod_{\substack{d' \in \{1, \dots, D\} \\ d' \neq d}} \mathbf{w}_{\sigma(i)}^{d'} \right) \right) \leq \prod_{d' \in \{1, \dots, D\}} \mathbf{W}^{d'}. \end{aligned} \quad (4.8c)$$

Les démonstrations sont des démonstrations assez proches de celles définies pour les matrices de uns consécutifs, elles ne sont donc pas développées.

4.3.4.3 Elimination de quelques symétries

Comme pour l'approche par matrices de uns-consécutifs, des MPQ-arbres équivalents sont énumérés plusieurs fois pour un même placement. En effet, par définition des MPQ-arbres équivalents, les arbres possèdent plusieurs représentations équivalentes. Il paraît donc inutile de toutes les énumérer. Nous allons donc définir des ordres lexicographiques

avec pour objectif de n'énumérer qu'une seule fois chaque arbre.

Soit N un nœud de l'arbre et T_N le sous-arbre enraciné en N . Soit i_{T_N} l'objet de plus petit indice contenu dans T_N : $i_{T_N} = \min\{i \in \mathcal{I} : i \in T_N\}$. Alors, un MPQ-arbre T est lexicographiquement ordonné si

- Pour chaque feuille ou nœud de type P , les éléments la composant doivent apparaître du plus petit au plus grand dans l'ordre σ .
- Pour chaque nœud de type P de fils $T_1, \dots, T_m$, les sous-arbres engendrés par ces derniers sont classés du plus petit au plus grand selon l'ordre de lecture gauche droite (cas de gauche de la figure 4.16) :

$$i_{T_i} < i_{T_j}, \quad \forall 1 \leq i < j \leq m.$$

- Pour chaque nœud de type Q de fils $T_1, \dots, T_m$, le sous-arbre engendré par le fils le plus à gauche (T_1) est lexicographiquement avant le sous-arbre le plus à droite (cas de droite de la figure 4.16) :

$$i_{T_1} < i_{T_m}.$$

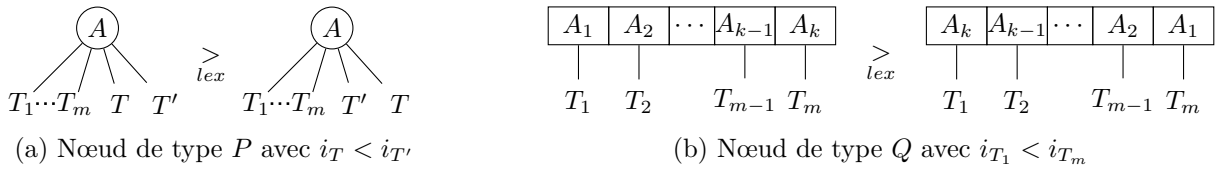


FIGURE 4.16: Ordre lexicographique entre des nœuds de MPQ-arbres équivalents

Notons enfin que pour chaque MPQ-arbre, il existe un unique MPQ-arbre lexicographiquement ordonné. L'ordonnancement associé à ce MPQ-arbre définit le seul ordre d'entrée nécessaire de l'algorithme 6. Par conséquent, notre algorithme n'énumère que les MPQ-arbres lexicographiquement ordonnés.

4.4 Résultats numériques

Ces algorithmes de contrôle de réalisabilité ont été utilisés dans un algorithme de « Branch-and-Bound » (exposé en section 1.4.5) pour résoudre le problème de sac-à-dos sur deux dimensions. Ces procédures ont été implémentées en Java 6 et testées sur un PC avec un processeur 1.73GHz Pentium core duo. La procédure de Fekete & Schepers a été développée en C++ et testée sur un PC avec un processeur 2.7GHz Pentium IV. Les algorithmes de Caprara et Monaci ont été testés sur un PC avec un processeur 800MHz Pentium III et ceux de Baldacci et Boschetti sur un PC avec un processeur 2.5GHz Pentium IV. Nos binaires sont disponibles à l'adresse suivante :

<https://wiki.bordeaux.inria.fr/realopt/pmwiki.php/Project/SoftwareAlgoKP>

Dans le tableau 4.17, les colonnes JP (resp. JP2) correspondent à notre algorithme utilisant l'approche des matrices de uns-consécutifs (resp. utilisant l'approche des MPQ-arbres). Similairement, les colonnes FS correspondent aux résultats de l'algorithme de Fekete et Schepers [33], la colonne BB aux résultats de l'approche de Baldacci et Boschetti

[3] et la colonne CM aux meilleurs résultats obtenus pour l'approche de Caprara et Monaci [13].

Le tableau 4.17 reporte le nombre de nœud de l'arbre de « Branch-and-Bound » étudié et le nombre d'appels à la procédure de contrôle de la réalisabilité d'une solution (contrôle fait après l'application de MDFF) pour nos approches et celle de Fekete et Schepers. Il montre que le nombre de nœud de l'arbre de « Branch-and-Bound » et le nombre d'appels aux procédures de contrôle est plus important sur notre approche que pour l'algorithme de Fekete et Schepers. Cela provient d'un arbre de « Branch-and-Bound » moins sophistiqué que celui de Fekete et Schepers pour sélectionner les sous-ensembles d'objets à contrôler. En particulier, aucune heuristique n'est utilisée par notre algorithme.

Enfin, nous reportons les différents temps de calcul des algorithmes dans le tableau 4.17. Lorsqu'un tiret apparaît, les instances n'ont pas été résolues par les auteurs. Pour les instances « ngcut », tous les algorithmes sont efficaces. Les instances « wang », « cgcut » et « gcut » sont plus difficiles à résoudre comme la taille du conteneur et/ou le nombre d'objets dans les solutions augmentent. Cependant, sur ces instances, nos algorithmes semblent être les plus performants, en particulier pour les instances « cgcut2 », « gcut4 », « gcut8 » et « gcut12 ». Les résultats pour les cinq instances « okp » sont plus mitigés : l'algorithme de Fekete et Schepers est plus rapide (à l'exception de « okp1 ») comparé à l'algorithme utilisant les matrices de uns-consécutifs. Cependant, nous remarquons que les nombres d'appels à la procédure de contrôle peuvent expliquer cette différence entre les algorithmes. En revanche, pour l'algorithme utilisant les MPQ-arbres, l'algorithme reste très performant pour ces instances.

L'instance « gcut13 » reste encore ouverte. Nous améliorons cependant la solution réalisable obtenue par Fekete et Schepers avec une solution impliquant 17 objets de valeur 8647565 (cf. figure 4.18).

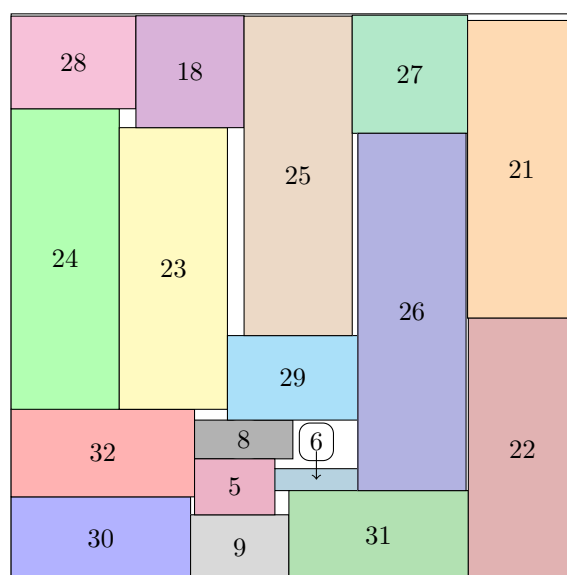


FIGURE 4.18: Une solution réalisable de gcut13 de valeur 8647565

Inst.	JP	JP	FS	FS	JP	JP2	FS	BB	CM
	nœud KNP	appel OPP	nœud KNP	appel OPP	temps calcul	temps calcul	temps calcul	temps calcul	temps calcul
ngcut1	23	13	19	1	0	0	0	-	-
ngcut2	74	19	5	0	0	0	0	-	-
ngcut3	29	19	25	6	0	0	0	-	-
ngcut4	7	6	1	0	0	0	0	-	-
ngcut5	8	7	1	0	0	0	0	-	-
ngcut6	38	26	15	5	0	0	0	-	-
ngcut8	67	62	53	23	0	0	0	-	-
ngcut9	21	18	3	0	0	0	0	-	-
ngcut10	7	0	1	0	0	0	0	-	-
ngcut11	115	78	36	10	0	0	0	-	-
ngcut12	210	46	48	14	0	0	0	-	-
cgcut1	44	11	14	1	0	0	0	0	0
cgcut2	889	870	*	*	36	135	>1800	>1800	531
cgcut3	582	227	356	102	0	3	0	95	4
wang20	4567	413	356	102	1	3	0	-	2
gcut1	69	18	33	0	0	0	0	0	0
gcut2	2873	169	519	51	0	0	0	0	0
gcut3	16503	564	2234	235	0	0	4	2	2
gcut4	310308	36026	72159	18316	31	179	195	46	346
gcut5	92	45	52	13	0	0	0	0	0
gcut6	1348	78	278	22	0	0	0	1	0
gcut7	8471	207	852	124	0	0	2	3	0
gcut8	313847	14685	55485	9037	19	76	255	186	108
gcut9	75	10	12	2	0	0	0	0	0
gcut10	1176	88	335	31	0	0	0	0	0
gcut11	29446	1426	1616	212	1	2	8	3	14
gcut12	102310	3096	8178	593	5	6	109	12	16
gcut13	*	*	*	*	>1800	>1800	>1800	>1800	>1800
okp1	9467	234	3244	661	1	1	10	779	24
okp2	72855	27585	23626	7310	675	67	20	288	1535
okp3	58161	8936	8233	816	10	1	5	0	1
okp4	26088	4227	1458	15	23	1	2	14	0
okp5	54401	29731	5733	643	>1800	319	11	190	40
non résolu	-	-	-	-	2	1	2	2	1

TABLEAU 4.17: Temps de résolution pour les différentes approches pour le problème de sac-à-dos sur deux dimensions

Pour le problème de réalisabilité, nous avons testé notre algorithme sur les instances définies par Clautiaux et al. Leur algorithme est testé sur un PC avec un processeur 2.6 GHz Pentium IV. La colonne CCM correspond aux résultats obtenus par Clautiaux et al. le tableau 4.17 reporte les temps de calcul obtenus par Fekete et Schepers, Clautiaux et al.

Pour ces instances, les temps de calcul sont moins concluant, en particulier pour les instances non réalisables. Une des explications est que la détection d'instances irréalisable

Instances réalisables				Instances irréalisables							
Inst.	JP	FS	CCM	Inst.	JP	FS	CCM	Inst.	JP	FS	CCM
E02F17	0	7	0	E00N10	9	0	0	E08N15	74	0	1
E02F20	0	*	1	E00N15	1	0	0	E10N10	0	0	0
E02F22	0	167	0	E00N23	45	*	0	E10N15	8	0	0
E04F15	2	0	0	E02N20	0	0	0	E13N10	0	0	0
E04F17	0	13	0	E03N10	0	0	0	E13N15	0	0	0
E04F19	0	560	0	E03N15	21	0	0	E15N10	0	0	0
E04F20	0	22	0	E03N16	22	2	1	E15N15	1	0	1
E05F15	0	0	0	E03N17	57	0	0	E00X23	*	*	2
E05F18	3	0	0	E04N15	5	0	0	E05X15	336	2	0
E05F20	0	491	0	E04N17	35	0	0	E07X15	236	0	0
E07F15	0	0	0	E04N18	26	10	0	E10X15	162	0	0
E08F15	0	0	0	E05N15	27	0	0	E13X15	7	0	0
E20F15	0	0	0	E05N17	42	0	0	non résolu	1	2	0
E03X18	17	0	0	E07N10	0	0	0				
E20X15	3	0	0	E07N15	6	0	0				

TABLEAU 4.19: Instances pour le problème de réalisabilité

peut-être plus difficile pour nos algorithmes. Les qualités des temps de calculs obtenus par Clautiaux et al. sont dus à l'utilisation de techniques de programmation par contraintes et de MDFF pour détecter rapidement des irréalisabilités. De plus, la taille du conteneur (20×20) implique une discrétisation de l'espace efficace pour leurs algorithmes. Il serait intéressant d'adapter de telles techniques non soumises à une discrétisation pour notre algorithme permettant de détecter des irréalisabilités rapidement après chaque nouvelle clique générée. Enfin, pour les instances irréalisables, nous remarquons que seul un sous-ensemble d'objets (parfois de l'ordre d'une dizaine d'objets sur 20) peut rentrer dans le conteneur. Ceci augmente ainsi artificiellement la difficulté mais en pratique elles ne seront pas traitées en l'état dans un algorithme de sac-à-dos sur deux dimensions.

Nous avons donc proposé deux nouveaux algorithmes compétitifs pour la résolution des problèmes de placement. L'algorithme basé sur les MPQ-arbres paraît être l'algorithme le plus compétitif car il est plus efficace dans l'élimination des cas symétriques. Cependant, sa structure de données est plus lourde à mettre en place.

*
* *

Conclusion

Suite à notre revue de la littérature des différentes modélisations du problème de placement sur deux dimensions, nous avons identifié et corrigé les formulations qui ne présentaient qu'une relaxation du problème. Nous avons également déduit de nouvelles contraintes valides éliminant des symétries. Une nouvelle formulation compacte fut aussi proposée ainsi que nouvelles formulations étendues. La comparaison numérique entre les qualités de bornes duales des formulations est un complément aux comparaisons théoriques. Cette étude nous a permis de mieux apprécier le compromis entre la qualité des bornes duales et les temps de calcul nécessaires. Nous avons observé que les contraintes de non-intersection des objets n'ont pas de grande conséquence sur la qualité des bornes duales mais peuvent justifier le gap résiduel entre bornes duales et solutions primales.

Pour modéliser au mieux la contrainte de non-intersection des objets pour les formulations étendues, nous avons proposé un schéma de branchement basée sur l'ajout de contraintes disjonctives. Cette approche originale se base sur une séparation des ensembles de colonnes en fonction des branchements effectués afin de ne pas modifier la structure du problème. Ce schéma a permis l'amélioration des bornes duales obtenues par les formulations étendues. Cependant, l'ajout de ces contraintes de branchement est insuffisante pour garantir de résoudre le problème à l'optimalité entière. Il serait intéressant de développer d'autres branchements pour garantir la contrainte de consécuité des bandes.

Enfin, nous avons proposé de nouvelles approches de résolution combinatoires pour résoudre le problème de placement sur deux dimensions. Elles sont basées sur des caractérisations des graphes d'intervalles : les matrices de uns consécutifs et les MPQ-arbres. Les algorithmes utilisées s'avèrent alors plus performant que les algorithmes de la littérature pour les problèmes de sac-à-dos sur deux dimensions. Cependant, pour certaines instances possédant de très nombreuses symétries, l'algorithme utilisant les matrices de

uns-consécutifs est moins performant que l'algorithme utilisant les MPQ-arbres ou que l'algorithme de Fekete et Schepers. Ce constat est dû à une moins bonne caractérisation des symétries par les matrices de uns-consécutifs que l'algorithme utilisant les MPQ-arbres. Pour le problème de réalisabilité dans un conteneur de petite taille, l'approche de Clautiaux et al. est performante. Cette performance se justifie par l'utilisation de la programmation par contraintes combiné avec une discrétisation de l'espace. En observant que nos algorithmes ne sont pas soumis à une discrétisation de l'espace, il serait intéressant d'adapter ces techniques de programmation par contraintes non soumises à une discrétisation de l'espace afin d'améliorer la détection de cas irréalisables après chaque clique générée. Enfin, nous envisageons d'utiliser nos approches pour résoudre le problème de « bin packing » sur deux dimensions en utilisant un algorithme de génération de colonnes associé avec notre algorithme pour résoudre les sous-problèmes.

Les annexes qui suivent présentent des travaux effectués en parallèle de ma thèse. Le premier travail annexe a consisté à étudier un problème industriel de grand taille. Une modélisation du problème à deux niveaux est présentée. Cette modélisation est à la l'origine de l'approche de résolution du problème et nous a permis d'obtenir des résultats performants parmi les meilleures équipes de recherche du challenge ROADEF/EURO 2010. Cette étude a permis de me confronter à des problèmes réels de taille industrielle pour lesquels la modélisation et la mise en place d'un algorithme performant soumis à des temps de calcul contraints sont primordiales. Le deuxième travail a consisté à développer des approches génériques d'heuristiques primales pour la génération de colonnes. En effet, une étude et analyse des différentes approches développées dans la littérature pour différentes applications nous ont permis de définir de telles heuristiques. Ces approches génériques nous permettent d'obtenir des solutions réalisables d'une qualité similaires à des approches de méta-heuristiques. Ces études complémentaires m'ont permis d'étendre mes connaissances méthodologiques et algorithmiques en programmation mathématique (utilisation avancée de CPLEX, génération de colonnes, heuristiques basées sur la programmation mathématique).

*

* *

Bibliographie

- [1] ROADEF/EURO Challenge 2010. A large-scale energy management problem with varied constraints. <http://challenge.roadef.org/2010/index.en.htm>.
- [2] A. Amaral and A. N. Letchford. An exact algorithm for general orthogonal two-dimensional knapsack problems. Technical report, Lancaster University, United Kingdom, 2002.
- [3] R. Baldacci and M. A. Boschetti. A cutting plane approach for the two-dimensional orthogonal non-guillotine cutting stock problem. *European Journal of Operational Research*, 183(3) :1136–1149, 2007.
- [4] J. E. Beasley. An exact two-dimensional non-guillotine cutting tree search procedure. *Operations Research*, 33(1) :49–64, 1985.
- [5] J. E. Beasley and A. Mingozzi. A new formulation for the two-dimensional orthogonal cutting problem. Technical report, University of Bologna, Italy, 1996.
- [6] N. Beldiceanu and M. Carlsson. New filtering for the cumulative constraint in the context of non-overlapping rectangles. In *Lecture Notes in Computer Science*, volume 5015, pages 21–25. CPAIOR08, 2008.
- [7] N. Beldiceanu, M. Carlsson, and S. Thiel. Sweep synchronization as a global propagation mechanism. *Computers and Operations Research*, 33(10) :2835–2851, 2006.
- [8] G. Belov, V. Kartak, H. Rohling, and G. Scheithauer. One-dimensional relaxations and LP bounds for orthogonal packing. *International Transactions in Operational Research*, 16(6) :745–766, 2009.
- [9] G. Belov and H. Rohling. A branch-and-price graph-theoretical algorithm for orthogonal-packing feasibility, 2009.

- [10] G. Belov and G. Scheithauer. Lower-dimensional bounds and a new model for higher-dimensional orthogonal packing, 2007.
- [11] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences*, 13(3) :335–379, 1976.
- [12] M. A. Boschetti, A. Mingozzi, and E. Hadjiconstantinou. New upper bounds for the two-dimensional orthogonal non-guillotine cutting stock problem. *IMA Journal of Management Mathematics*, 13(2) :95–119, 2002.
- [13] A. Caprara and M. Monaci. On the two-dimensional knapsack problem. *Operations Research Letters*, 32(1) :5–14, 2004.
- [14] A. Caprara and M. Monaci. Bidimensional packing by bilinear programming. *Mathematical Programming*, 2007.
- [15] A. Caprara, M. Monaci, and D. Vigo. An exact approach to the strip-packing problem. *INFORMS Journal on Computing*, 13(3) :310–319, 2003.
- [16] J. Carlier and E. Néron. A new LP-based lower bound for the cumulative scheduling problem. *European Journal of Operational Research*, 127(2) :363–382, 2000.
- [17] J. Carlier and E. Néron. On linear lower bounds for the resource constraint project scheduling problem. *European Journal of Operational Research*, 149(2) :314–324, 2003.
- [18] J. Carlier and E. Néron. Computing redundant resources for the resource constrained project scheduling problem. *European Journal of Operational Research*, 176(3) :1452–1463, 2007.
- [19] C. S. Chen, S. M. Lee, and Q. S. Shen. An analytical model for the container loading problem. *European Journal of Operational Research*, 86(1) :68–76, 1995.
- [20] F. Clautiaux, C. Alves, and J. V. de Carvalho. A survey of dual-feasible and super-additive functions. *Annals of Operations Research*, 2008.
- [21] F. Clautiaux, J. Carlier, and A. Moukrim. A new exact method for the orthogonal packing problem. *European Journal of Operational Research*, 183(3) :1196–1211, 2007.
- [22] F. Clautiaux, A. Jouglet, J. Carlier, and A. Moukrim. A new constraint programming approach for the orthogonal packing problem. *Computers and Operations Research*, 35(3) :944–959, 2008.
- [23] F. Clautiaux, A. Moukrim, and J. Carlier. New data-dependent dual-feasible functions and lower bounds for a two-dimensional bin-packing problem. Technical report, University of Lille, 2007.
- [24] G. Desaulniers, J. Desrosiers, and M. M. Solomon. *Column generation*. Springer, 2005.
- [25] H. Dyckhoff. A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2) :145–159, 1990.
- [26] G. Fasano. MIP-based heuristic for non-standard 3d-packing. *A Quarterly Journal of Operational Research*, 6(3) :291–310, 2008.

- [27] S. P. Fekete and J. Schepers. A new exact algorithm for general orthogonal d-dimensional knapsack problems. In *Algorithms – ESA ’97*, volume 1284 of *Lecture Notes in Computer Science*, pages 144–156. Springer-Verlag, 1997.
- [28] S. P. Fekete and J. Schepers. On more-dimensional packing i : Modeling. Technical report, University of Köln, Germany, 1997.
- [29] S. P. Fekete and J. Schepers. On more-dimensional packing ii : Bounds. Technical report, University of Köln, Germany, 1997.
- [30] S. P. Fekete and J. Schepers. New classes of lower bounds for the bin packing problem. In *Integer Programming and Combinatorial Optimization (IPCO 98)*, volume 1412 of *Lecture Notes in Computer Science*, pages 257–270. Springer-Verlag, 1998.
- [31] S. P. Fekete and J. Schepers. New classes of lower bounds for the bin packing problem. *Mathematical Programming*, 91 :11–31, 2001.
- [32] S. P. Fekete and J. Schepers. A combinatorial characterization of higher-dimensional orthogonal packing. *Mathematics of Operations Research*, 29(2) :353–368, 2004.
- [33] S. P. Fekete, J. Schepers, and J. van der Veen. An exact algorithm for higher-dimensional orthogonal packing. *Operations Research*, 55(3) :569–587, 2007.
- [34] E.P. Ferreira and J.F. Oliveira. *Fekete and Schepers’ Graph-based Algorithm for the Two-Dimensional Orthogonal Packing Problem Revisited*. Gabler, 2008.
- [35] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15 :835–855, 1965.
- [36] F. Vanderbeck. Branching in branch-and-price : a generic scheme. *MP*, 20(1) :1–46, 2010.
- [37] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
- [38] P. C. Gilmore and A. J. Hoffman. A linear programming approach to the cutting stock problem – part i. *Operations Research*, 9 :849–859, 1961.
- [39] P. C. Gilmore and A. J. Hoffman. A linear programming approach to the cutting stock problem – part ii. *Operations Research*, 11 :863–888, 1963.
- [40] P. C. Gilmore and A. J. Hoffman. A characterization of comparability graphs and interval graphs. *Canadian Journal of Mathematics*, 16 :539–548, 1964.
- [41] E. Hadjiconstantinou and N. Christofides. An exact algorithm for general, orthogonal, two-dimensional knapsack problems. *European Journal of Operational Research*, 83(1) :39–56, 1995.
- [42] E. Huang and R. E. Korf. New improvements in optimal rectangle packing. In *IJCAI*, pages 511–516, 2009.
- [43] C. Joncour, S. Michel, R. Sadykov, D. Sverdlov, and F. Vanderbeck. Column generation based primal heuristics. volume 36, pages 695–702. *Electronic Notes in Discrete Mathematics*, 2010.

- [44] C. Joncour and A. Pêcher. Consecutive ones matrices for multi-dimensional orthogonal packing problems. In *20th International Symposium of Mathematical Programming*, 2009.
- [45] C. Joncour and A. Pêcher. Consecutive ones matrices for multi-dimensional orthogonal packing problems. In *International Symposium on Combinatorial Optimization*, volume 36, pages 327–334. Electronic Notes in Discrete Mathematics, 2010.
- [46] C. Joncour, A. Pêcher, P. Pesneau, and F. Vanderbeck. Mathematical programming formulations for the orthogonal 2d knapsack problem : a survey. In *9^e congrès annuel de la Société française de Recherche Opérationnelle et d’Aide à la Décision*, 2008.
- [47] C. Joncour, A. Pêcher, and P. Valicov. Mpq-trees for orthogonal packing problem. In *International Symposium on Combinatorial Optimization*, volume 36, pages 423–429. Electronic Notes in Discrete Mathematics, 2010.
- [48] R. E. Korf. Optimal rectangle packing : New results. In *ICAPS*, pages 142–149, 2004.
- [49] Richard E. Korf. Optimal rectangle packing : Initial results. In *ICAPS*, pages 287–295, 2003.
- [50] N. Korte and R. Möhring. An incremental linear-time algorithm for recognizing interval graphs. *SIAM Journal on Computing*, 18(1) :68–81, 1989.
- [51] S. Martello and D. Vigo. Exact solution of the two-dimensional finite bin packing problem. *Management Sciences*, 44(3) :388–399, 1998.
- [52] S. Olariu. An optimal greedy heuristic to color interval graphs. *Information Processing Letters*, 37(1) :21–25, 1991.
- [53] H. Onodera, Y. Taniguchi, and K. Tamaru. Branch-and-bound placement for building block layout. In *28th ACM/IEEE design automation conference*, pages 433–439, 1991.
- [54] M. Padberg. Packing small boxes into a big box. *Mathematical Methods Operations Research*, 52(1) :1–21, 2000.
- [55] D. Pisinger and M. Sigurd. Using decomposition techniques and constraint programming for solving the two-dimensional bin packing problem. *INFORMS Journal on Computing*, 19(1) :36–51, 2007.
- [56] G. Scheithauer. LP-based bounds for the container and multi-container loading problem. *International Transactions in Operational Research*, 6 :199–213, 1999.
- [57] H. Simonis and B. O’Sullivan. Search strategies for rectangle packing, 2008.
- [58] R. D. Tsai, E. M. Malstorm, and H. D. Meeks. A two-dimensional palletizing procedure for loading operations. *IIE Transactions*, 20 :418–425, 1988.
- [59] P. van Hentenryck. Scheduling and packing in the constraint language cc(fd). In *in intelligent scheduling*. Morgan Kaufmann, 1994.
- [60] F. Vanderbeck and L. A. Wolsey. Reformulation and decomposition of integer programs. In *“50 Years of Integer Programming”*, pages 431–502, 2010.

- [61] P. Y. Wang. Two algorithms for constrained two-dimensional cutting stock problems. *Operations Research*, 31 :573–586, 1983.
- [62] G. Wäscher, H. Haußner, and H. Schuman. An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3) :1109–1130, 2007.

*
* *

ANNEXES

Sommaire

1	Formulation	148
2	A 2-Stage Hierarchical Optimisation Approach	157
3	Selected Approach	166

ANNEXE A

Power Plant Production Planning

Summary of the approach developped for the ROADEF Challenge 2010

The electricity production planning problem considered herein can be summarized as follows: Each nuclear reactor of the network must be shut down periodically for maintenance and refueling. Scheduling these shutdowns is complicated and must consider a number of factors related to regulatory and technical requirements (such as restrictions on the number of weeks between maintenance of each nuclear reactor, the rate of consumption of nuclear fuel, etc.)

Moreover, these decisions must take into account the inherent uncertainty of the problem environment. Specifically, one must meet the demand of its clientele for electrical power in each of a large set of *scenarios*, each of which may occur with a given probability. It must schedule the nuclear reactor maintenance operations *before* knowing which scenario may occur. It is important to avoid making decision which could have very costly consequences under some possible scenarios (an example of a costly consequence would be that one would be forced to burn a large amount of coal during a given period of time to cover customer demand because its scheduled nuclear shutdowns prevent it from having enough capacity to generate power by other means). Given this, the goal is to determine a maintenance and refueling schedule of all of the nuclear power plants in the network that respects the technical constraints, and that minimizes the total expected value of all the costs involved (both those directly related to refueling and maintenance and those related to the generation of power). For more explanation, see:

<http://challenge.roadef.org/2010/index.en.htm>

1 Formulation

1.1 Notation

- $t = 0, \dots, \mathbf{T} - 1$ denotes time periods (steps), where $\mathbf{T} \simeq 3000, 5000$. Each period corresponds to time interval $[t, t + 1]$ that has length $\mathbf{D}^t$. We also define the set $\mathcal{T} = \{0, \dots, \mathbf{T} - 1\}$ and enumerate time periods $t \in \mathcal{T}$.
- $w = 0, \dots, \mathbf{W} - 1$ denotes weeks where $\mathbf{W} \simeq 156, 260$. We define the set $\mathcal{W} = \{0, \dots, \mathbf{W} - 1\}$ and enumerate weeks $w \in \mathcal{W}$. t_w is the period starting on week $w \in \mathcal{W}$, inversely, w_t is the week that includes period t .
- $i = 0, \dots, \mathbf{I} - 1$ denotes nuclear reactor (Type II production plant), $\mathbf{I} = 50$. We also use notation $i \in \mathcal{I}$. $\mathbf{T} + 1$.
- $j = 0, \dots, \mathbf{J} - 1$ denotes other power plants (Type I production plant), $\mathbf{J} = 100$. We also use notation $j \in \mathcal{J}$.
- $k = -1, \dots, \mathbf{K} - 1$ denotes the sequence number of a type II production campaign or cycle that is separate from the next campaign by a shutdown or “outage” during which maintenance and refueling are performed, for every type II plant i . $\mathbf{K} \simeq 5$. A **cycle** k is made of **two consecutive phases**: an **outage** (while maintenance and refueling take place) and a **production campaign** (while electric production and consuming fuel level). The latter is itself divided into an **on-production** period where power production follows a production profile, and a potentiel **off-production period**, after a **shutdown**, while the production is null until the next maintenance. See an illustration in Figure 1. By convention, for every type II plant i , the initial condition is denoted cycle $k = -1$ and the plant is within the on-production period with an initial stock level $\mathbf{XI}_i$. We use notation $k \in \mathcal{K}$.
- $s = 0, \dots, \mathbf{S} - 1$ denotes scenarios that define demand levels, power production bounds and costs for type I plant. $\mathbf{S} \simeq 500$. We also use notation $s \in \mathcal{S}$.
- $m = 0, \dots, \mathbf{Np}_{i,k} - 1$ denotes the production mode at type II plant i that depends on the residual stock level as defined in the production profile k . In mode 0, the fuel level is greater or equal $\mathbf{BO}_{i,k}$, the power production is normally at its upper bound $\mathbf{PMAX}_i^t$, although power modulation is possible. Subsequent modes correspond to production under low fuel stock level (lower than $\mathbf{BO}_{i,k}$) and maximum production level follows a piecewise linear decreasing function with $\mathbf{Np}_{i,k}$ break points, going down to zero production when fuel level is below a critical level. For $m = 0$, $(\mathbf{f}_{i,k,0}, \mathbf{c}_{i,k,0}) = (\mathbf{BO}_{i,k}, 1)$. For $m = 1, \dots, \mathbf{Np}_{i,k} - 1$, while fuel level decreases from $\mathbf{f}_{i,k,m-1}$ to $\mathbf{f}_{i,k,m}$, power decreases from $\mathbf{c}_{i,k,m-1}$ to $\mathbf{c}_{i,k,m}$ percent of $\mathbf{PMAX}_i^t$. A tolerance of $\pm\epsilon$ is possible. See illustration in Figure 2.

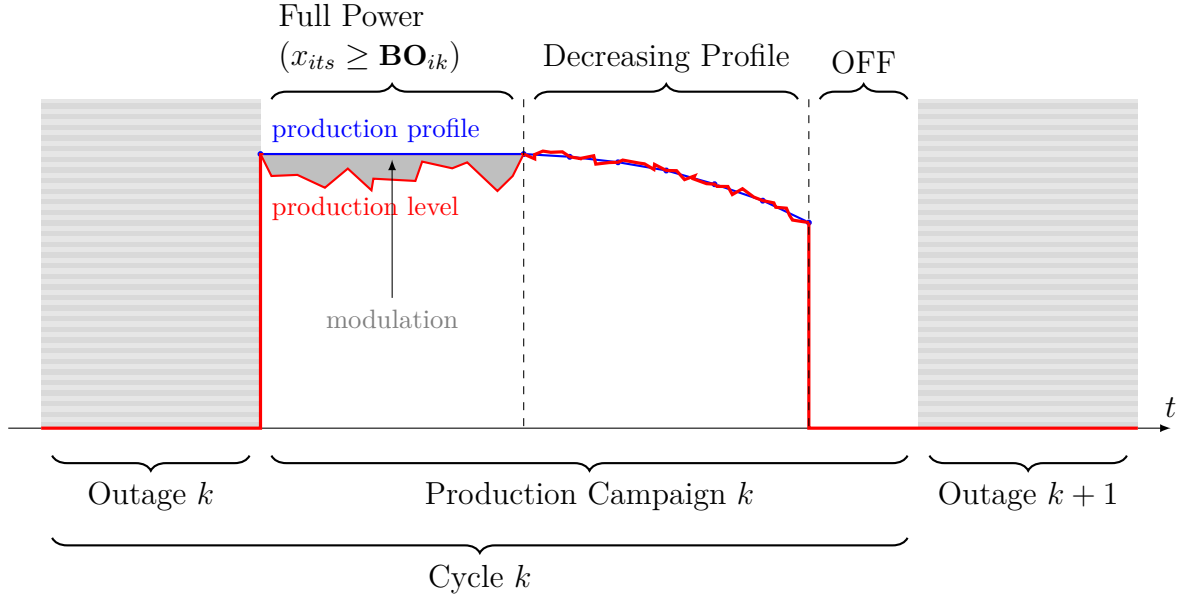


Figure 1: Illustration of notation

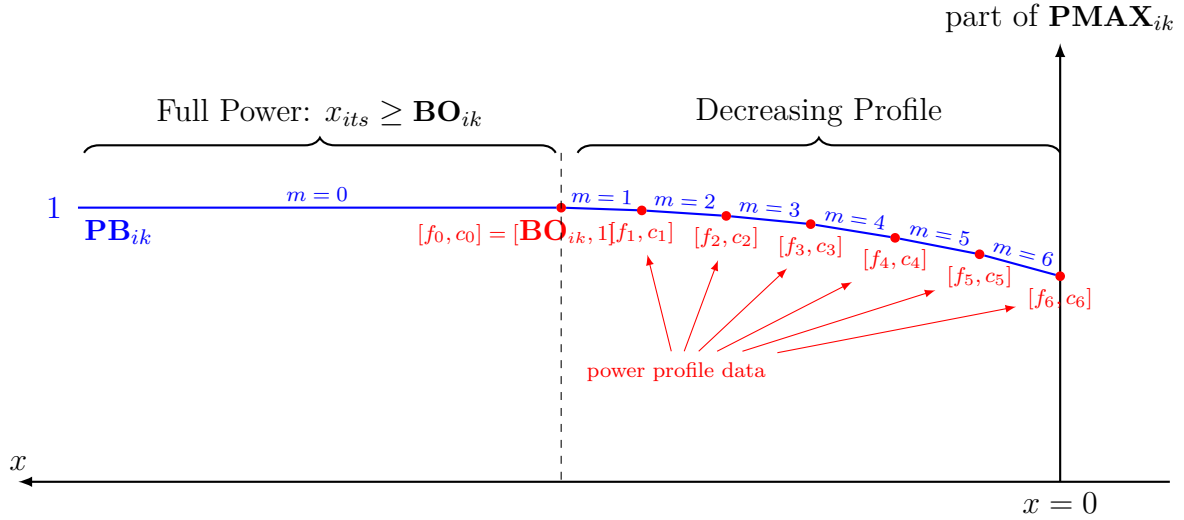


Figure 2: Imposing decreasing power profile function

1.2 Variables

1.2.1 Decision Variables

- $d_{i,k,w} = 1$ if the decoupling that sets the start of cycle $k \neq -1$ arises at the beginning of week w or earlier (i.e. production campaign $k - 1$ has ended by the end of week $w - 1$) at plant $i \in \mathcal{I}$; 0 otherwise. The earliest week w for which $d_{i,k,w} = 1$ is called a “decoupling” week. The duration of outage k is fixed to $\mathbf{DA}_{i,k}$ and therefore the coupling stating of the on-production phase of cycle k must have been started by the decoupling week $+\mathbf{DA}_{i,k}$. By convention, we put $d_{i,-1,w} = 1$ and $\mathbf{DA}_{i,-1} = 0$.
- $z_{i,k,m,s,t} = 1$ if plant $i \in \mathcal{I}$ is in on-production running production campaign k in the mode m (i.e. at stock level greater or equal to the minimum stock characterizing

- mode m) under scenario s during time period t ; 0 otherwise.
- $o_{i,k,s,t} = 1$ if plant $i \in \mathcal{I}$ is running the campaign k in the off-production phase in period t for scenario s ; 0 otherwise.
- $r_{i,k,w}$ is the amount of fuel reloaded during outage k in week w if the current outage is that of cycle $k \neq -1$. Refueling takes place in the first period of the outage.
- $p_{i,s,t}$ is the production at plant $i \in \mathcal{I}$, in period t under scenario s .
- $p_{j,s,t}$ is the production at plant $j \in \mathcal{J}$, in period t under scenario s .

1.2.2 Required State Variables (resulting from the above decisions)

- $x_{i,s,t}$ is the fuel stock level at plant $i \in \mathcal{I}$, at the outset of period t , under scenario s .
- $p_{i,k,m,s,t}$ is the target production level at plant i if the production mode is m under scenario s in period t .
- $f_{i,s,w}$ is the amount of fuel supplied under scenario s in week w (it is non zero only on the first period of an outage; and defined only for $w \in [\mathbf{TO}_{i,k}, \mathbf{TA}_{i,k}]$ for some k).
- $a_{i,s,w}$ is the adjustment in fuel level for type II plant i in week w under scenario s during refueling.
- $v_{i,k,s,t}$ is the power modulation at plant i in period t of production campaign k under scenario s .
- $g_{i,k,k',i'} = 1$ if the decoupling week of outage k at plant i precedes the decoupling week of outage k' at plant i' ; 0 otherwise.

1.2.3 Redundant State Variables (resulting from the above decisions)

- $\delta_{i,k,w} = 1$ if the decoupling that sets the start of cycle $k \neq -1$ arises at the beginning of week w at plant $i \in \mathcal{I}$; 0 otherwise.
- $y_{i,k,w} = 1$ if plant $i \in \mathcal{I}$ is running production campaign k (either in the on-production or off-production phase) in week w ; 0 otherwise.
- $ha_{i,k}$ is the starting week during which plant i goes into outage k ; 0 otherwise.

1.3 Constraints

Outage and Production campaign setup/shutdown consistency

$$d_{i,k,w} \geq d_{i,k,w-1}, \quad \forall i, k, w. \quad (1)$$

$$d_{i,k,w} \leq d_{i,k-1,w-\mathbf{DA}_{i,k-1}}, \quad \forall i, k, w. \quad (2)$$

Outage variables and Outage week

$$\delta_{i,k,w} = (d_{i,k,w} - d_{i,k,w-1}) \in \{0, 1\}, \quad \forall i, k \neq -1, w. \quad (3)$$

$$ha_{i,k} = \sum_w w \delta_{i,k,w} = \sum_w w (d_{i,k,w} - d_{i,k,w-1}), \quad \forall i, k \neq -1. \quad (4)$$

where $ha_{i,k}$ is the starting week during which plant i goes off-line for outage k .

Production campaign variables

$$y_{i,k,w} = (d_{i,k,w-\mathbf{DA}_{i,k}} - d_{i,k+1,w}) \in \{0, 1\}, \quad \forall i, k, w. \quad (5)$$

$$\sum_k y_{i,k,w} + \sum_{k \neq -1} \sum_{w' = w - \mathbf{DA}_{i,k}}^w \delta_{i,k,w'} = 1, \quad \forall i, w. \quad (6)$$

On-production mode step variables and Off-production variables

$$\sum_{m=0}^{\mathbf{Np}_{i,k}-1} z_{i,k,m,s,t} + o_{i,k,s,t} = y_{i,k,w_t}, \quad \forall i, k, s, t. \quad (7)$$

$$z_{i,k,m,s,t} + y_{i,k,w_{t-1}} \leq 1 + \sum_{m' \leq m} z_{i,k,m',s,t-1}, \quad \forall i, k, s, t \text{ and for } m = 0, \dots, \mathbf{Np}_{i,k} - 1. \quad (8)$$

$$o_{i,k,s,t} + y_{i,k,w_{t+1}} \leq 1 + o_{i,k,s,t+1}, \quad \forall i, k, s, t. \quad (9)$$

CT1: Demand covering

$$\sum_i p_{i,s,t} + \sum_j p_{j,s,t} = \mathbf{DEM}^{t,s}, \quad \forall s, t. \quad (10)$$

where $\mathbf{DEM}^{t,s}$ is the global power demand (load) that must be satisfied in period t under scenario s .

CT2: Bounded production levels at type I plant

$$\mathbf{PMIN}_{j,t}^s \leq p_{j,s,t} \leq \mathbf{PMAX}_{j,t}^s, \quad \forall j, s, t. \quad (11)$$

where $\mathbf{PMAX}_{j,t}^s$ is the maximum power production of power plant j in period t under scenario s and $\mathbf{PMIN}_{j,t}^s$ is the minimum power production of power plant j in period t under scenario s .

CT3: Off-line production upper bound at type II plant

$$p_{i,s,t} \leq \mathbf{PMAX}_i^t \sum_k (y_{i,k,w_t} - o_{i,k,s,t}), \quad \forall i, s, t. \quad (12)$$

where $y_{i,k,w} = 1$ if plant $i \in \mathcal{I}$ is running production campaign k in week w and $\mathbf{PMAX}_i^t$ is the maximum power production of power plant i in period t .

CT4: Minimum production at type II plant

$$p_{i,s,t} \geq 0, \quad \forall i, s, t. \quad (13)$$

Minimum production at type II plant when fuel level is high

$$p_{i,s,t} \geq \sum_k \left(\mathbf{PMAX}_i^t - \frac{\mathbf{MMAX}_{i,k}}{\mathbf{D}^t} \right) z_{i,k,0,s,t}, \quad \forall i, s, t. \quad (14)$$

where $\mathbf{MMAX}_{i,k}$ is the maximum deviation (modulation) from the maximum power production of power plant i in any period of campaign k .

CT5 : Maximum production at type II plant when fuel level is high

$$p_{i,s,t} \leq \mathbf{PMAX}_i^t, \quad \forall i, s, t. \quad (15)$$

redundant with CT3.

CT6 : Production profile at type II plant when fuel level is low

$$p_{i,k,m,s,t} \leq \mathbf{PMAX}_i^t(\mathbf{c}_{i,k,m} + \mathbf{PB}_{i,k,m}(x_{i,s,t} - \mathbf{f}_{i,k,m})), \quad \forall i, k, s, t \text{ and for } m = 1, \dots, \mathbf{Np}_{i,k} - 1. \quad (16)$$

$$p_{i,k,m,s,t} \leq \mathbf{PMAX}_i^t \mathbf{c}_{i,k,m-1} z_{i,k,m,s,t}, \quad \forall i, k, s, t \text{ and for } m = 1, \dots, \mathbf{Np}_{i,k} - 1. \quad (17)$$

$$p_{i,k,m,s,t} \geq \mathbf{PMAX}_i^t(\mathbf{c}_{i,k,m} + \mathbf{PB}_{i,k,m}(x_{i,s,t} - \mathbf{f}_{i,k,m})) - \mathbf{PMAX}_i^t \mathbf{c}_{i,k,m-1}(1 - z_{i,k,m,s,t}), \quad (18)$$

$$\forall i, k, t \text{ and for } m = 1, \dots, \mathbf{Np}_{i,k} - 1.$$

where $\mathbf{PB}_{i,k,m}$ are the decreasing production rates of the production profile of plant i during campaign k for $m = 1, \dots, \mathbf{Np}_{i,k} - 1$: $\mathbf{PB}_{i,k,m} = \frac{\mathbf{c}_{i,k,m-1} - \mathbf{c}_{i,k,m}}{\mathbf{f}_{i,k,m-1} - \mathbf{f}_{i,k,m}}$.

Thus $p_{i,k,m,s,t} = 0$ if the plant is not in mode m .

$$(1 - \varepsilon) \sum_k \sum_{m=1}^{\mathbf{Np}_{i,k}-1} p_{i,k,m,s,t} \leq p_{i,s,t} \leq (1 + \varepsilon) \sum_k \sum_{m=1}^{\mathbf{Np}_{i,k}-1} p_{i,k,m,s,t} + \mathbf{PMAX}_i^t z_{i,k,0,s,t}, \quad \forall i, k, s, t. \quad (19)$$

Bounded fuel stock at type II plant

$$\sum_k \sum_{m=0}^{\mathbf{Np}_{i,k}-1} \mathbf{f}_{i,k,m} z_{i,k,m,s,t} \leq x_{i,s,t} \leq \sum_k \sum_{m=0}^{\mathbf{Np}_{i,k}-1} \mathbf{f}_{i,k,m-1} z_{i,k,m,s,t}, \quad \forall i, s, t \quad (20)$$

where $\mathbf{f}_{i,k,m}$ is the lower threshold of the fuel stock level of power plant i that characterizes mode $m = 0, \dots, \mathbf{Np}_{i,k} - 1$ with $\mathbf{f}_{i,k,0} = \mathbf{BO}_{i,k}$ and $\mathbf{f}_{i,k,-1} = \mathbf{SMAX}_{i,k}$.

CT7: Bounded Fuel Reloading

$$\mathbf{RMIN}_{i,k} \leq r_{i,k,w} \leq \mathbf{RMAX}_{i,k}, \quad \forall i, k \neq -1, w. \quad (21)$$

$\mathbf{RMAX}_{i,k}$ (resp. $\mathbf{RMIN}_{i,k}$) is the maximum (resp. minimum) refilling fuel level power production of power plant i in the shutdown (outage) following production campaign k .

CT8: Initial Fuel stock

$$x_{i,s,0} = \mathbf{XI}_i, \quad \forall i, s. \quad (22)$$

where $\mathbf{XI}_i$ is the initial fuel stock level at plant $i \in \mathcal{I}$.

CT9 - CT10: Fuel Dynamic

$$\begin{cases} x_{i,s,t+1} = x_{i,s,t} - \mathbf{D}^t p_{i,s,t} + f_{i,s,w_t}, & \forall i, s, t = t_w, \\ x_{i,s,t+1} = x_{i,s,t} - \mathbf{D}^t p_{i,s,t}, & \forall i, s, t \neq t_w. \end{cases} \quad (23)$$

where $p_{i,s,t} = 0$ during shutdowns, $\mathbf{D}^t$ is the duration (length in hours) of time period t (it also measures the fuel consumption per unit of power production in period t , and $f_{i,s,w}$ is the amount (possibly negatif) of fuel supplied under scenario s in week w_t (it is nonzero only on the first period of a shutdown period).

CT10: Fuel Supply during an outage

$$f_{i,s,w} \leq \sum_{k \neq -1} (\mathbf{SMAX}_{i,k} - \mathbf{AMAX}_{i,k-1}) \delta_{i,k,w}, \quad \forall i, s, w. \quad (24)$$

where $\mathbf{SMAX}_{i,k}$ is the maximum stock level after refueling in the outage following production campaign k at plant i .

$$f_{i,s,w} \leq \sum_{k \neq -1} (r_{i,k,w} + a_{i,s,w}), \quad \forall i, s, w. \quad (25)$$

where $r_{i,k,w}$ is the fuel reload in week w if the current outage is that of cycle k . And $a_{i,s,w}$ is the adjustment in fuel level in week w under scenario s during refueling.

$$r_{i,k,w} \leq \mathbf{RMAX}_{i,k} \delta_{i,k,w}, \quad \forall i, k \neq -1, w. \quad (26)$$

where $\mathbf{RMAX}_{i,k}$ is the maximum refueling for type II plant in campaign k .

$$a_{i,s,w} = \left(\mathbf{BO}_{i,k} - \frac{(\mathbf{Q}_{i,k} - 1)}{\mathbf{Q}_{i,k}} \mathbf{BO}_{i,k-1} \right) \delta_{i,k,w} - \frac{1}{\mathbf{Q}_{i,k}} x_{i,s,t_w}, \quad \forall i, s, w. \quad (27)$$

where $\mathbf{Q}_{i,k}$ is the refueling coefficient during the outage k at plant i and $\mathbf{BO}_{i,k}$ is the threshold above which the fuel level is considered as high at plant i during production campaign k . However, for every plant i , instances show that $\mathbf{Q}_{i,k}$ and $\mathbf{BO}_{i,k}$ are constants for all campaign.

CT11: Bounded Fuel Stock Levels

$$0 \leq x_{i,s,t} \leq \begin{cases} \sum_k \mathbf{SMAX}_{i,k} (d_{i,k,w_t} - d_{i,k+1,w_t}) - \sum_{k \neq -1} (\mathbf{SMAX}_{i,k} - \mathbf{AMAX}_{i,k}) \delta_{i,k,w_t}, & \forall i, s, t = t_w, \\ \sum_k \mathbf{SMAX}_{i,k} (d_{i,k,w_t} - d_{i,k+1,w_t}), & \forall i, s, t \neq t_w. \end{cases} \quad (28)$$

where $\mathbf{AMAX}_{i,k}$ is the maximum stock level before refueling in the outage following production campaign k at plant i and $\mathbf{SMAX}_{i,k}$ is the maximum stock level during production campaign k at plant i .

CT12: Maximum modulation over a cycle

$$v_{i,k,s,t} \geq 0, \quad \forall i, k, s, t. \quad (29)$$

$$v_{i,k,s,t} \geq \mathbf{PMAX}_i^t z_{i,k,0,s,t} - p_{i,s,t}, \quad \forall i, k, s. \quad (30)$$

$$\sum_t \mathbf{D}^t v_{i,k,s,t} \leq \mathbf{MMAX}_{i,k}, \quad \forall i, k, s. \quad (31)$$

where $\mathbf{D}^t$ is the duration (length in hours) of time period t and $\mathbf{MMAX}_{i,k}$ is the maximum deviation (modulation) from the maximum power production of power plant i in any period of campaign k .

CT13: Time windows for shutdowns (outages)

Variables $d_{i,k,w}$ are true variables for $w \in [\mathbf{TO}_{i,k}, \mathbf{TA}_{i,k}]$ where $\mathbf{TO}_{i,k}$ is the first possible week for the start of outage k and $\mathbf{TA}_{i,k}$ is the last possible week for the start of outage k . While there are fixed outside this interval:

$$d_{i,k,w} \leq 0, \quad \forall i, k \neq -1, w < \mathbf{TO}_{i,k}. \quad (32)$$

$$d_{i,k,w} \geq 1, \quad \forall i, k \neq -1, w \geq \mathbf{TA}_{i,k}. \quad (33)$$

CT14: Minimum time space between outages

$$\overbrace{\sum_w w (d_{i',k',w} - d_{i',k',w-1})}^{ha_{i',k'}} \geq \overbrace{\sum_w w (d_{i,k,w} - d_{i,k,w-1})}^{ha_{i,k}} + \mathbf{DA}_{i,k} + \mathbf{Se14} - M(1 - g_{i,k,k',i'}), \quad (34)$$

$$\forall (i, k), (i', k') \in \mathbf{A14} \times \mathbf{A14}.$$

$$g_{i,k,k',i'} + g_{i',k',k,i} = 1, \quad \forall (i, k), (i', k') \in \mathbf{A14} \times \mathbf{A14}. \quad (35)$$

where $\mathbf{A14}$ is the set of outages (i, k) that are concerned by constraint 14, $\mathbf{Se14}$ is the time space required between 2 outages in constraint 14 (a negative value corresponds to an authorized overlapping), and $\mathbf{DA}_{i,k}$ is the length (duration in weeks) of outage k at plant i .

An alternative model consists in considering a fictive resource 14 of capacity 1, while all outages $(i, k) \in \mathbf{A14}$ consume 1 unit of this resource during $(\mathbf{DA}_{i,k} + \mathbf{Se14})^+$ weeks. Then, the constraint takes the form:

$$\sum_{(i,k) \in \mathbf{A14}} (d_{i,k,w} - d_{i,k,w-(\mathbf{DA}_{i,k} + \mathbf{Se14})^+}) \leq 1, \quad \forall w. \quad (36)$$

In the data, the set seems to be given as set of power plant : $\mathbf{A14} \subset \mathcal{I}$, with implicitly all outage of of this power plant being concerned by the constraint.

CT15: Minimum time space between outages arising within a specific time window

$\mathbf{A15}$ is the set of outages (i, k) that are concerned by constraint 15, $\mathbf{Se15}$ is the time space required between 2 outages in constraint 15 (a negative value corresponds to an authorized overlapping)

One option is to model constraint 15 as:

$$\overbrace{\sum_w w (d_{i',k',w} - d_{i',k',w-1})}^{ha_{i',k'}} \geq \overbrace{\sum_w w (d_{i,k,w} - d_{i,k,w-1})}^{ha_{i,k}} + \mathbf{DA}_{i,k} + \mathbf{Se15} - M(1 - g_{i,k,k',i'}), \quad (37)$$

$$\forall (i, k), (i', k') \in \mathbf{A15} \times \mathbf{A15}.$$

$$g_{i,k,k',i'} + g_{i',k',k,i} \geq (d_{i,k,\mathbf{IF15}} - d_{i,k,\mathbf{ID15}-\mathbf{DA}_{i,k}}) + (d_{i',k',\mathbf{IF15}} - d_{i',k',\mathbf{ID15}-\mathbf{DA}_{i,k}}) - 1, \quad (38)$$

$$\forall (i, k), (i', k') \in \mathbf{A15} \times \mathbf{A15}.$$

where $[\mathbf{ID15}, \mathbf{IF15}]$ is the time window associated with constraint 15.

An alternative model consists in considering a fictive resource 15 of capacity 1, while all outages $(i, k) \in \mathbf{A15}$ consume 1 unit of this resource during the $(\mathbf{DA}_{i,k} + \mathbf{Se15})^+$ weeks that intersect time window $[\mathbf{ID15}, \mathbf{IF15}]$. Then, the constraint takes the form:

$$\sum_{(i,k) \in \mathbf{A15}} (d_{i,k,w} - d_{i,k,w-(\mathbf{DA}_{i,k} + \mathbf{Se15})^+}) \leq 1, \quad \forall w \in [\mathbf{ID15}, \mathbf{IF15}]. \quad (39)$$

In the data, the set seems to be given as set of power plant : $\mathbf{A15} \subset \mathcal{I}$, with implicitly all outage of of this power plant being concerned by the constraint.

CT16: Minimum time space between dates of decoupling

Considering a fictive resource 16 of capacity 1, while all outages $(i, k) \in \mathbf{A16}$ consume 1 unit of this resource during $\mathbf{Se16} > 0$ weeks, the constraint takes the form:

$$\sum_{(i,k) \in \mathbf{A16}} (d_{i,k,w} - d_{i,k,w-\mathbf{Se16}}) \leq 1, \quad \forall w. \quad (40)$$

In the data, the set seems to be given as set of power plant : $\mathbf{A16} \subset \mathcal{I}$, with implicitly all outage of of this power plant being concerned by the constraint.

CT17: Minimum time space between dates of coupling

Considering a fictive resource 17 of capacity 1, while all campaigns $(i, k) \in \mathbf{A17}$ consume 1 unit of this resource during $\mathbf{Se17} > 0$ weeks, the constraint takes the form:

$$\sum_{(i,k) \in \mathbf{A17}} (d_{i,k,w-\mathbf{DA}_{i,k}} - d_{i,k,w-\mathbf{DA}_{i,k}-\mathbf{Se17}}) \leq 1, \quad \forall w \text{ corresponding to end week.} \quad (41)$$

In the data set seems to be given as set of power plant : $\mathbf{A17} \subset \mathcal{I}$, with implicitly all outage of of this power plant being concerned by the constraint.

CT18: Minimum time space between dates of coupling and decoupling

Considering a fictive resource 18 of capacity 1, while all maintenances $(i, k) \in \mathbf{A18}$ consume 1 unit of this resource during $\mathbf{Se18} > 0$ weeks for each decoupling and coupling, the constraint takes the form:

$$\sum_{(i,k) \in \mathbf{A18}} \overbrace{(d_{i,k,w} - d_{i,k,w-\mathbf{Se18}})}^{\text{decoupling cons}} + \overbrace{(d_{i,k,w-\mathbf{DA}_{i,k}} - d_{i,k,w-\mathbf{DA}_{i,k}-\mathbf{Se18}})}^{\text{coupling cons}} \leq 1, \quad \forall w. \quad (42)$$

In the data set seems to be given as set of power plant : $\mathbf{A18} \subset \mathcal{I}$, with implicitly all outage of of this power plant being concerned by the constraint.

NOTE: this requires $\mathbf{Se18} \leq \mathbf{DA}_{i,k} \forall (i, k) \in \mathbf{A18}$; otherwise the constraint links coupling and decoupling at the same plant.

CT19: Limited Resource for Maintenance

Considering a maintenance resource of capacity $\mathbf{Q19}$, while all campaigns $(i, k) \in \mathbf{A19}$ consume 1 unit of this resource during $\mathbf{TU19}_{i,k}$ weeks starting from the date of outage with a shift of $\mathbf{L19}_{i,k} \in \{0, \dots, \mathbf{DA}_{i,k} - 1\}$, with $(\mathbf{L19}_{i,k} + \mathbf{TU19}_{i,k}) \leq \mathbf{DA}_{i,k}$. The constraint takes the form:

$$\sum_{(i,k) \in \mathbf{A19}} (d_{i,k,w-\mathbf{L19}_{i,k}} - d_{i,k,w-\mathbf{L19}_{i,k}-\mathbf{TU19}_{i,k}}) \leq \mathbf{Q19}, \quad \forall w. \quad (43)$$

In the data set seems to be given as set of power plant : $\mathbf{A19} \subset \mathcal{I}$, with implicitly all outage of of this power plant being concerned by the constraint.

CT20: Limited Number of Concomitant Maintenance

Considering a counting resource of capacity $\mathbf{N20}_w$, while all maintenance $(i, k) \in$

$\mathbf{A20}_w$ count for 1 unit of this counting ressource during $\mathbf{DA}_{i,k}$ weeks, the constraint takes the form:

$$\sum_{(i,k) \in \mathbf{A20}_w} (d_{i,k,w} - d_{i,k,w-\mathbf{DA}_{i,k}}) \leq \mathbf{N20}_w, \quad \forall w. \quad (44)$$

In the data, the set seems to be the set of all power plants : $\mathbf{A20} = \mathcal{I}$.

Common Format to Scheduling Constraints CT14-CT20

Each scheduling and spacing constraint c is a special case of a common format. Let

- $\mathbf{L}_{i,k,c}$ be the number of weeks after the start of outage job i after which resource is used.
- $\mathbf{TU}_{i,k,c}$ be the number of weeks during which resource is used by the outage at plant i , meaning that job uses resource during the interval

$$\left[\sum_{w=0}^{W-1} w \delta_{i,k,w} + \mathbf{L}_{i,k,c} ; \sum_{w=0}^{W-1} w \delta_{i,k,w} + \mathbf{L}_{i,k,c} + \mathbf{TU}_{i,k,c} \right].$$

- $\mathbf{Q}_c$ be the available number of units of the resource,
- $\mathbf{ID}_c, \mathbf{IF}_c$ be the first and the last week of the time interval for which the constraint is defined,
- $\mathbf{A}_c$ be the set of plants for which the constraint is defined.

then, the constraint takes the form:

$$\sum_{i \in \mathbf{A}_c, k} \left(\sum_{\tilde{w} \in [w - \mathbf{L}_{i,k,c} - \mathbf{TU}_{i,k,c}, w - \mathbf{L}_{i,k,c}]} \delta_{i,k,\tilde{w}} \right) \leq \mathbf{Q}_c, \quad \forall c, w \in [\mathbf{ID}_c, \mathbf{IF}_c]. \quad (45)$$

which can be written as flow conservation constraints:

$$\sum_{i \in \mathbf{A}_c, k} \left(\sum_{w = \mathbf{ID}_c - \mathbf{L}_{i,k,c} - \mathbf{TU}_{i,k,c} + 1}^{\mathbf{ID}_c - \mathbf{L}_{i,k,c}} \delta_{i,k,w} \right) + \text{slack}_{c, \mathbf{ID}_c} = \mathbf{Q}_c, \quad \forall c. \quad (46)$$

$$\sum_{i \in \mathbf{A}_c, k} \delta_{i,k,w - \mathbf{L}_{i,k,c} - \mathbf{TU}_{i,k,c}} + \text{slack}_{c,w-1} = \sum_{i \in \mathbf{A}_c, k} \delta_{i,k,w - \mathbf{L}_{i,k,c}} + \text{slack}_{c,w}, \quad \forall c, w \in (\mathbf{ID}_c, \mathbf{IF}_c]. \quad (47)$$

CT21: Bounded off-line production capacity

Given a set of plant $\mathbf{C21}$, a time interval (in weeks) $\mathbf{IT21} = [\mathbf{ID21}; \mathbf{IF21}]$ and a maximum off-line production capacity $\mathbf{IMAX21}$, the constraint takes the form:

$$\sum_{i \in \mathbf{C21}} \left(\left(\sum_{t=t_w}^{t_{w+1}-1} \mathbf{PMAX}_i^t \right) \sum_k (d_{i,k,w} - d_{i,k,w-\mathbf{DA}_{i,k}}) \right) \leq \mathbf{IMAX21}, \quad \forall w \in \mathbf{IT21}. \quad (48)$$

In the data, the set seems to be the set of all power plants : $\mathbf{A21} = \mathcal{I}$.

1.4 Objective

$$\sum_{i,k,w} \mathbf{C}_{i,k} r_{i,k,w} + \frac{1}{\mathbf{S}} \sum_{j,s,t} \mathbf{C}_{j,t}^s \mathbf{D}^t p_{j,s,t} - \sum_i \mathbf{C}_i x_{i,T+1}^s. \quad (49)$$

Where $\mathbf{C}_{i,k}$ is the refueling cost, $\mathbf{C}_{j,t}^s$ is the type I plant cost and $\mathbf{C}_i$ is the type II plant cost for the last period time.

2 A 2-Stage Hierarchical Optimisation Approach

We consider a 2 stage solution approach

In stage 1: we solve the aggregate planning problem in weekly period based on an average scenario.

In stage 2: Having fixed the linking constraints/variables (δ^*, y^*, r^*) , we consider one problem for each scenario $s \in \mathcal{S}$.

2.1 Stage 1 - phase A: a Dantzig-Wolfe decomposition approach

Let us consider an hierarchical optimisation approach: in the first stage we fix type II plant production campaign scheduling and refueling, in the second stage we decide on power production and resulting fuel stock dynamic under each scenarios. However, we could solve the second stage problem for several “good” first stage solution and consider local search procedure that induce shift in first stage decisions.

The following approach makes use of two observations:

1. As resource constraint tend to concern different subsets of plants or different week intervals, aggregating these into a single cumulative surrogate resource could be a good approximation.
2. The main decision concerning type II plant are campaign scheduling and refueling. They do depend on the scenarios. The adjustment in power production made to follow the scenarios can be seen as marginal (they are restricted by production lower bounds derived from the upper bounds). Hence, the main issue is to determine production capacity (upper bounds), which can be seen as a **production profile** over the whole time horizon. Production variation will however generate variation in fuel consumption and fuel lost in adjustments that can be seen as the cost of power variations for the second stage problem.

2.1.1 The master

Given a set of feasible production profiles G^i for type I central i , each of which can be summarized by a fuel cost c^g , a production capacity p_w^g for each week w , and a the resource utilisation $u_{c,w}^g$ for capacity type c for week w , the problem can be reformulated as:

$$\min \sum_{g \in G^i} c^g \lambda_g + \sum_{jw} C_{j,w} p_{j,w}, \quad (50)$$

$$\sum_i \sum_{g \in G^i} p_{i,w}^g \lambda_g + \sum_j p_{j,w} \geq \mathbf{D}^w, \quad \forall w, \quad (\pi_w) \quad (51)$$

$$\mathbf{PMIN}_j^w \leq p_{j,w} \leq \mathbf{PMAX}_j^w, \quad \forall j, w, \quad (52)$$

$$\sum_{i \in \mathbf{A}_c} \sum_{g \in G^i} u_{i,c,w}^g \lambda_g \leq \mathbf{Q}_c, \quad \forall c, w, \quad (\rho_{c,w}) \quad (53)$$

$$\sum_{g \in G^i} \lambda_g \leq 1, \quad \forall i, \quad (54)$$

$$\lambda_g \geq 0, \quad \forall i, g \in G^i. \quad (55)$$

Remark $\mathbf{D}^w$ could be initialized with the average demand. Then, one could identified what are the most violated scenarios, in terms of demand covering in (52) or power production bound in (52), and add cuts for these. $\blacklozenge$

Remark The modulation of power at type II central and the resulting cost could be introduced in this master for a finer formulation. $\blacklozenge$

2.1.2 The pricing subproblem

For type II central, the pricing problem over set G^i takes the form:

$$\min \sum_w \left(\sum_c \rho_{i,c,w} u_{i,c,w} - \pi_w p_{i,w} \right) + \sum_k \mathbf{C}_{i,k} r_{i,k}, \quad (56)$$

$$x_{i,k} + r_{i,k} + a_{i,k} = p_{i,k} + x_{i,k+1}, \quad \forall k, \quad (57)$$

$$\text{combining LS \& Scheduling a central under resource constraint,} \quad (58)$$

$$\text{switching from On to Off to Maintenance in cycles,} \quad (59)$$

$$\text{flow with resource constraints,} \quad (60)$$

We could derive a **dynamic programming** approach implementing a shortest path computation in a network corresponding to arrow going south-east in the Table 1. The possible transition and their fuel used, power production and resource used are reported in Table 2 for given initial stock for $k = -1$ or reload and stock bounds for $k > 0$. For every transition, $\mathbf{pp}$ is the power production and $\mathbf{D} \cdot \mathbf{pp}$ is the fuel used. To fuel cost, one must add fixed cost for resource usage and subtract reward for power production. Note that for each cycle there is a **Fixed minimum Fuel Reload (FFR)**:

For $k = -1$,

$$\mathbf{FFR}_{-1} = \mathbf{X}\mathbf{I}_i.$$

For $k = 0, \dots, K-1$,

$$\mathbf{FFR}_k = \left(\mathbf{RMIN}_{i,k} + \mathbf{BO}_{i,k} - \frac{(\mathbf{Q}_{i,k} - 1)}{\mathbf{Q}_{i,k}} \mathbf{BO}_{i,k-1} \right).$$

A more accurate model would associate a state $(C; X)$ to each nodes in the network of Table 1, where C is the accumulated cost while X is the current stock level. Then the dynamic program would become pseudopolynomial, f.i., in a transition $\text{FP}(k, w_1) \rightarrow \text{DP}^l(k, w_2)$:

$$(C; X) \rightarrow (C - \sum_w \pi_w p_w + \mathbf{C}_{i,k} r; X - \mathbf{D} \cdot \mathbf{pp} + \mathbf{FFR}_k + r).$$

Observe that only then, the master LP bound can become stronger than the LP bound that could be obtained by translating the network flow formulation underlying the DP in the original formulation.

A more accurate fuel cost estimation can be obtained by keeping track of the cumulative power production $\mathbf{CPP}$ and the cumulatif fixed reload $\mathbf{CFR}_k$ up to cycle k . The later is a constant, while the former must be record in the state but it is fixed for a given partial path. Hence, for each node in the network, we record a state $(C, \mathbf{CPP})$ that is update at each transition. For instance, a transition $\text{FP}(k, w_1) \rightarrow \text{DP}^l(k, w_2)$ gives rise to

$$\begin{aligned} & (C; \mathbf{CPP}) \\ & \quad \downarrow \\ & \left(C - \sum_w \pi_w p_w + \mathbf{C}_{i,k} \left((\mathbf{D} \mathbf{CPP}' - \mathbf{CFR}_k)^+ + \frac{1}{\mathbf{Q}_{i,k}} (\mathbf{CFR}_k - \mathbf{D} \mathbf{CPP}')^+ \right); \underbrace{\mathbf{CPP} + \sum_w p_w}_{\mathbf{CPP}'} \right). \end{aligned}$$

Cycle	States	Weeks														
		0	1	2	3	4	5	6	7	8	9	10	11	12	...	W-1
$k = -1$	INIT	.													...	
	FP	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	DP ¹	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	⋮	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	DP ⁷	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	OFF	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
$k = 0$	OUT	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	FP	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	DP ¹	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	⋮	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	DP ⁷	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	OFF	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
$k = 1$	OUT	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	FP	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	DP ¹	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	⋮	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	DP ⁷	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	OFF	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
⋮		⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
$k = K - 1$	OUT	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	FP	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	DP ¹	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	⋮	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	DP ⁷	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	OFF	.	.	.	.	.	.	.	.	.	.	.	.	.	...	.
	END															

Table 1: Going from the initial state INIT, successively through production at full power (FP), Decreasing Profile for 1 upto 7 weeks (DP¹, ..., DP⁷), off-production (OFF) and Outage (OUT) in cycle $k = 0, \dots, K$ and the END state with fuel resale price. In all event, the week index marks the beginning of the event.

Transition	power production ($\mathbf{pp} = \sum_w p_w$)	Fuel Cost Estim.	power p reward	resource cost
INIT $\rightarrow$ FP($-1, 0$)	0	0	0	0
INIT $\rightarrow$ DP ^{<i>l</i>} ($-1, 0$)	0	0	0	0
INIT $\rightarrow$ OFF($-1, 0$)	0	0	0	0
FP(k, w_1) $\rightarrow$ DP ^{<i>l</i>} (k, w_2)	$p_w = \mathbf{PMAX}_i^w$ for $w \in [w_1, w_2[$	$\mathbf{C}_{i,k}(\mathbf{D} \cdot \mathbf{pp} - \mathbf{FFR}_k)^+$	$\sum_w \pi_w p_w$	0
FP(k, w_1) $\rightarrow$ OFF(k, w_2)	$p_w = \mathbf{PMAX}_i^w$ for $w \in [w_1, w_2[$	$\mathbf{C}_{i,k}(\mathbf{D} \cdot \mathbf{pp} - \mathbf{FFR}_k)^+$	$\sum_w \pi_w p_w$	0
FP(k, w_1) $\rightarrow$ OUT($k + 1, w_2$)	$p_w = \mathbf{PMAX}_i^w$ for $w \in [w_1, w_2[$	$\mathbf{C}_{i,k}(\mathbf{D} \cdot \mathbf{pp} - \mathbf{FFR}_k)^+$	$\sum_w \pi_w p_w$	0
FP(k, w_1) $\rightarrow$ END	$p_w = \mathbf{PMAX}_i^w$ for $w \in [w_1, w_2[$	$\mathbf{C}_{i,k}(\mathbf{D} \cdot \mathbf{pp} - \mathbf{FFR}_k)^+$	$\sum_w \pi_w p_w$	0
DP ^{<i>l</i>} (k, w_1) $\rightarrow$ OFF($k, w_1 + l$)	$p_w = \mathbf{PMAX}_i^w \frac{(\mathbf{C}_{i,k,\tau} + \mathbf{C}_{i,k,\tau+1})}{2}$ for $w \in [w_1; w_1 + l[$	$\mathbf{C}_{i,k}(\mathbf{D} \cdot \mathbf{pp} - \mathbf{FFR}_k)^+$	$\sum_w \pi_w p_w$	0
DP ^{<i>l</i>} (k, w_1) $\rightarrow$ OUT($k + 1, w_1 + l$)	$p_w = \mathbf{PMAX}_i^w \frac{(\mathbf{C}_{i,k,\tau} + \mathbf{C}_{i,k,\tau+1})}{2}$ for $w \in [w_1; w_1 + l[$	$\mathbf{C}_{i,k}(\mathbf{D} \cdot \mathbf{pp} - \mathbf{FFR}_k)^+$	$\sum_w \pi_w p_w$	0
DP ^{<i>l</i>} (k, w_1) $\rightarrow$ END	$p_w = \mathbf{PMAX}_i^w \frac{(\mathbf{C}_{i,k,\tau} + \mathbf{C}_{i,k,\tau+1})}{2}$ for $w \in [w_1; w_1 + l[$	$\mathbf{C}_{i,k}(\mathbf{D} \cdot \mathbf{pp} - \mathbf{FFR}_k)^+$	$\sum_w \pi_w p_w$	0
OFF(k, w_1) $\rightarrow$ OUT($k + 1, w_2$)	0	0	0	0
OFF(k, w_1) $\rightarrow$ END	0	0	0	0
OUT(k, w_1) $\rightarrow$ FP($k, w_1 + \mathbf{DA}_{i,k}$)	0	0	0	$\sum_{c,w} \rho_{c,w} \delta_{c,w}(w_1)$
OUT(k, w_1) $\rightarrow$ DP ^{<i>l</i>} ($k, w_1 + \mathbf{DA}_{i,k}$)	0	0	0	$\sum_{c,w} \rho_{c,w} \delta_{c,w}(w_1)$
OUT(k, w_1) $\rightarrow$ OFF($k, w_1 + \mathbf{DA}_{i,k}$)	0	0	0	$\sum_{c,w} \rho_{c,w} \delta_{c,w}(w_1)$
OUT(k, w_1) $\rightarrow$ END	0	0	0	$\sum_{c,w} \rho_{c,w} \delta_{c,w}(w_1)$

Table 2: Feasible Transitions & Costs

2.1.3 Alternative Column Definition

A stronger model can be obtained by defining columns corresponding to a schedule for a set of plant in the same region (subject to CT14 type of scheduling constraints).

2.2 Stage 1: an extended formulation

Here we propose a reformulation of an approximation of the stage-1 problem that rely of variables modeling macroscopic event. Preprocessing can have more impact on macro event (such as full production campaign). Moreover the definition of variables associated to macro event allows to build in these definitions some of the problem constraints. We assume for now that we only have scheduling constraint associated to resource consumption.

2.2.1 State and Transition Variables

We denotes by **ES** and **LS** (resp. **EE** and **LE**) the earliest and the latest start (resp. end) week for every state.

$PC_{i,k,l}(w_1, w_2) = 1$ if in the planning of plant i the production campaign of cycle k starts at the beginning of week w_1 and ends on week w_2 , with the last l weeks being spent in the decreasing profile, with $l \in [0, w_2 - w_1]$.

Existence

- $w_1 \in [\mathbf{ES}_{i,k}^{fp}; \mathbf{LS}_{i,k}^{fp}]$; $w_2 - l \in [\mathbf{EE}_{i,k}^{fp}; \mathbf{LE}_{i,k}^{fp}] = [\mathbf{ES}_{i,k}^{dp}; \mathbf{LS}_{i,k}^{dp}]$; $w_2 \in [\mathbf{EE}_{i,k}^{dp}; \mathbf{LE}_{i,k}^{dp}]$,
- $\mathbf{ES}_{i,k}^{pc} = \mathbf{ES}_{i,k}^{fp}$; $\mathbf{ES}_{i,k}^{pc} = \mathbf{LS}_{i,k}^{fp}$; $\mathbf{EE}_{i,k}^{pc} = \mathbf{EE}_{i,k}^{dp}$ and $\mathbf{EE}_{i,k}^{pc} = \mathbf{LE}_{i,k}^{dp}$,
- $w_2 - l - w_1 \in [\mathbf{L}_{i,k}^{\min}(\text{FP}), \mathbf{L}_{i,k}^{\max}(\text{FP})]$ and $l \in [\mathbf{L}_{i,k}^{\min}(\text{DP}), \mathbf{L}_{i,k}^{\max}(\text{DP})]$.

Associated Micro Variable Setting

- $y_{i,k,w} = 1$ for $w \in [w_1, w_2[$,
- $\mathbf{p}_{i,k,w} = \mathbf{PMAX}_i^w$ for $w \in [w_1, w_2 - l[$,
- $\mathbf{p}_{i,k,w} = \mathbf{PMAX}_i^w \frac{(\mathbf{C}_{i,k,\tau} + \mathbf{C}_{i,k,\tau+1})}{2}$, for $w \in [w_2 - l, w_2[$,
- $\mathbf{pp}_{i,k,l}(w_1, w_2) = \sum_{w=w_1}^{w_2-1} \mathbf{p}_{i,k,w}$.

Associated Cost

- first estimation: $\mathbf{C}_{i,k}(\mathbf{D} \cdot \mathbf{pp}_{i,k,l}(w_1, w_2) - \mathbf{FFR}_k)^+$,
- finer estimation (lower bound) using cumulative power production so far **CPP** and cumulative fuel minimum reload **CFR**:

$$\mathbf{C}_{i,k} \left((\mathbf{D} \cdot \mathbf{CPP} - \mathbf{CFR})^+ + \frac{1}{\mathbf{Q}_{i,k}} (\mathbf{CFR} - \mathbf{D} \cdot \mathbf{CPP})^+ \right).$$

$\text{OUT}_{i,k}(w, w + \mathbf{DA}_{i,k}) = \delta_{i,k,w} = 1$ if in the planning of plant i the outage of cycle k starts at the beginning of week w and ends on week $w + \mathbf{DA}_{i,k}$. The outage can be preceeded by weeks being spent in an OFF-mode ending cycle $k - 1$ while waiting for the outage starting cycle k .

Existence

- $w \in [\mathbf{ES}_{i,k}^{\text{out}}, \mathbf{LS}_{i,k}^{\text{out}}]$ with $\mathbf{TO}_{i,k} \leq \mathbf{ES}_{i,k}^{\text{out}} \leq \mathbf{LS}_{i,k}^{\text{out}} \leq \mathbf{TA}_{i,k}$,
- $\mathbf{EE}_{i,k}^{\text{out}} = \mathbf{ES}_{i,k}^{\text{out}} + \mathbf{DA}_{i,k}$ and $\mathbf{LE}_{i,k}^{\text{out}} = \mathbf{LS}_{i,k}^{\text{out}} + \mathbf{DA}_{i,k}$.

Associated Micro Variable Setting

- $\delta_{i,k,w} = 1$,

- u_{c,w_a,w_b} is the number of resource capacity c used during weeks $[w_a, w_b[$ given that an outage starts in w . For CT14, ..., CT17, CT19, ..., CT20, $u_{c,w_a,w_b} = 1$ for $w_a = w + \mathbf{L}_{i,c}$ and $w_b = w + \mathbf{L}_{i,c} + \mathbf{TU}_{i,c}$.

Associated Cost 0.

$\text{OFF}_{i,k,w} = 1$ iff plant i is off line in week w of cycle k .

Existence $w \in [\mathbf{ES}_{i,k}^{\text{off}}, \mathbf{EE}_{i,k}^{\text{off}}]$,

Associated Cost 0.

$\text{END}_{i,w} = 1$ iff the planning pf plant i ends in week w .

Associated Cost $-\text{CI}_i(\mathbf{CFR} - \mathbf{D CPP})^+$.

$\text{slack}_{c,w}$ is the slack resource capacity c that is unused during week w .

2.2.2 Formulation

Fuel stock at the end of previous production campaign

$$0 \leq x_{i,k}^{\text{epc}} \leq \mathbf{AMAX}_{i,k}, \quad \forall i, k = 0, \dots, K. \quad (61)$$

Fuel stock after a reload

$$0 \leq x_{i,k}^{\text{reload}} \leq \mathbf{SMAX}_{i,k}, \quad \forall i, k \neq -1. \quad (62)$$

$$x_{i,-1}^{\text{reload}} = \mathbf{XI}_i, \quad \forall i. \quad (63)$$

Variable fuel reload

$$0 \leq r_{i,k}^{\text{var}} \leq \mathbf{RMAX}_{i,k} - \mathbf{RMIN}_{i,k}, \quad \forall i, k \geq 0. \quad (64)$$

Fixed fuel reload

$$0 \leq r_{i,k}^{\text{fixed}} \leq \mathbf{RMIN}_{i,k} + \mathbf{BO}_{i,k} - \frac{(\mathbf{Q}_{i,k} - 1)}{\mathbf{Q}_{i,k}} \mathbf{BO}_{i,k-1}, \quad \forall i, k \geq 0. \quad (65)$$

Carry-over from previous cycle

$$0 \leq z_{i,k} \leq \frac{(\mathbf{Q}_{i,k} - 1)}{\mathbf{Q}_{i,k}} \mathbf{AMAX}_{i,k}, \quad \forall i, k \geq 0. \quad (66)$$

Binary transitions

$$\overbrace{\text{OUT}_{i,k}(w, w + \mathbf{DA}_{i,k})}^{\delta_{i,k,w}} \in \{0, 1\}, \quad \forall i, k, w \in [\mathbf{ES}_{i,k}^{\text{out}}, \mathbf{LS}_{i,k}^{\text{out}}], \quad (67)$$

$$\text{PC}_{i,k,l}(w_1, w_2) \in \{0, 1\}, \quad \forall i, k, l, w_1 \in [\mathbf{ES}_{i,k}^{\text{pc}}, \mathbf{LS}_{i,k}^{\text{pc}}], w_2 \in [\mathbf{EE}_{i,k}^{\text{pc}}, \mathbf{LE}_{i,k}^{\text{pc}}], \quad (68)$$

$$\text{OFF}_{i,k,w} \in \{0, 1\}, \quad \forall i, k, w \in [\mathbf{ES}_{i,k}^{\text{off}}, \mathbf{LE}_{i,k}^{\text{off}}], \quad (69)$$

$$\text{END}_{i,w} \in \{0, 1\}, \quad \forall i, w. \quad (70)$$

Integer resource slack

$$\text{slack}_{c,w} \in \{0, \dots, \mathbf{Q}_c\}, \quad \forall c, w \in [\mathbf{ID}_c, \mathbf{IF}_c]. \quad (71)$$

Path defining a plant i planning (flow balance)

$$1 = \sum_{l,w} \text{PC}_{i,k=-1,l}(0, w) \quad \forall i, \quad (72)$$

$$\sum_{l,w'} \text{PC}_{i,k-1,l}(w', w) + \text{OFF}_{i,k-1,w-1} = \text{OFF}_{i,k-1,w} + \delta_{i,k,w}, \quad \forall i, k \geq 0, w \in [\underbrace{\mathbf{ES}_{i,k-1}^{off}, \mathbf{LE}_{i,k-1}^{off}}_{\mathbf{EE}_{i,k-1}^{pc}}]. \quad (73)$$

$$\sum_{l,w} \text{PC}_{i,k,l}(w, \mathbf{W}) + \text{OFF}_{i,k,\mathbf{W}-1} = \text{END}_{i,\mathbf{W}}, \quad \forall i, k \geq 0. \quad (74)$$

$$\delta_{i,k,w} = \text{END}_{i,\min(\mathbf{W}, w + \mathbf{DA}_{i,k})} + \sum_{l, w_2 \geq w + \mathbf{DA}_{i,k}} \text{PC}_{i,k,l}(w + \mathbf{DA}_{i,k}, w_2), \quad \forall i, k \geq 0, w \in [\mathbf{ES}_{i,k}^{out}, \mathbf{LS}_{i,k}^{out}]. \quad (75)$$

$$\sum_{\substack{k,w \\ w + \mathbf{DA}_{i,k} \geq \mathbf{W}}} \delta_{i,k,w} + \sum_{w=0}^{\mathbf{W}} \text{END}_{i,w} = 1, \quad \forall i. \quad (76)$$

Path defining a resource c use (Flow Balance for constraints CT14 to CT20)

$$\sum_{i \in \mathbf{A}_c, k} \left(\sum_{w = \mathbf{ID}_c - \mathbf{L}_{i,k,c} - \mathbf{TU}_{i,k,c} + 1}^{\mathbf{ID}_c - \mathbf{L}_{i,k,c}} \delta_{i,k,w} \right) + \text{slack}_{c, \mathbf{ID}_c} = \mathbf{Q}_c, \quad \forall c. \quad (77)$$

$$\sum_{i \in \mathbf{A}_c, k} \delta_{i,k, w - \mathbf{L}_{i,k,c} - \mathbf{TU}_{i,k,c}} + \text{slack}_{c, w-1} = \sum_{i \in \mathbf{A}_c, k} \delta_{i,k, w - \mathbf{L}_{i,k,c}} + \text{slack}_{c, w}, \quad \forall c, w \in (\mathbf{ID}_c, \mathbf{IF}_c]. \quad (78)$$

Knapsack constraints (Off-line power)

$$\sum_{i \in \mathbf{C21}} \mathbf{PMAX}_i^w \sum_k \sum_{\bar{w} = w - \mathbf{DA}_{i,k} + 1}^w \delta_{i,k, \bar{w}} \leq \mathbf{IMAX21}, \quad \forall w \in \mathbf{IT21}. \quad (79)$$

Demand covering

$$\sum_{i,k,l,w_1,w_2} \mathbf{PP}_{i,k,l}(w_1, w_2) \text{PC}_{i,k,l}(w_1, w_2) + \sum_j p_{j,w} \geq \mathbf{D}_w, \quad \forall w. \quad (80)$$

Fuel Dynamic: end of campaign stock

$$x_{i,k+1}^{epc} = x_{i,k}^{reload} - \sum_{l,w_1,w_2} \mathbf{D}^t \left(\sum_{w=w_1}^{w_2-1} \mathbf{PP}_{i,k,l}(w_1, w_2) \text{PC}_{i,k,l}(w_1, w_2) \right), \quad \forall i, k \geq -1. \quad (81)$$

Fuel Dynamic: reload

$$x_{i,k}^{reload} = r_{i,k}^{var} + r_{i,k}^{fixed} + \frac{(\mathbf{Q}_{i,k} - 1)}{\mathbf{Q}_{i,k}} x_{i,k}^{epc} + z_{i,k}, \quad \forall i, k \geq 0. \quad (82)$$

Fuel Dynamic: fixed reload

$$r_{i,k}^{fixed} = \left(\mathbf{RMIN}_{i,k} + \mathbf{BO}_{i,k} - \frac{(\mathbf{Q}_{i,k} - 1)}{\mathbf{Q}_{i,k}} \mathbf{BO}_{i,k-1} \right) \left(\sum_{w = \mathbf{ES}_{i,k}^{out}}^{\mathbf{LS}_{i,k}^{out}} \delta_{i,k,w} \right), \quad \forall i, k \geq 0. \quad (83)$$

Fuel Dynamic: variable reload

$$r_{i,k}^{var} \leq \left(\mathbf{RMAX}_{i,k} - \mathbf{RMIN}_{i,k} \right) \left(\sum_{w=\mathbf{ES}_{i,k}^{out}}^{\mathbf{LS}_{i,k}^{out}} \delta_{i,k,w} \right), \quad \forall i, k \geq 0. \quad (84)$$

Fuel Dynamic: carry over

$$z_{i,k} \leq \left(\frac{1}{\mathbf{Q}_{i,k}} \mathbf{AMAX}_{i,k} \right) \left(1 - \sum_{w=\mathbf{ES}_{i,k}^{out}}^{\mathbf{LS}_{i,k}^{out}} \delta_{i,k,w} \right), \quad \forall i, k \geq 0. \quad (85)$$

$$z_{i,k} \leq \frac{1}{\mathbf{Q}_{i,k}} x_{i,k}^{epc}, \quad \forall i, k \geq 0. \quad (86)$$

Objective

$$\min \sum_{i,k} \mathbf{C}_{i,k} r_{i,k}^{var} + \sum_{j,w} \mathbf{C}_{j,w} \mathbf{D}^t p_{j,w} - \sum_i C_{i,T+1} x_{i,K}^{epc}. \quad (87)$$

For a tighter estimation of fuel costs, one can keep track of accumulated power production: “cumulative production **CPP**, and cumulative fuel minimum reload **CFR**”.

$$\mathbf{CPP}_{i,w_2} \geq \mathbf{CPP}_{i,w_1} + \sum_l \mathbf{PC}_{i,k,l}(w_1, w_2) \mathbf{PC}_{i,k,l}(w_1, w_2) - M(1 - \sum_l \mathbf{PC}_{i,k,l}(w_1, w_2)). \quad (88)$$

$$\mathbf{CFR}_{i,w_2} \geq \mathbf{CFR}_{i,w_1} + \mathbf{FFR}_{i,k} \mathbf{OUT}_{i,k}(w_1, w_2) - M(1 - \mathbf{OUT}_{i,k}(w_1, w_2)). \quad (89)$$

2.3 Stage 2: in one pass

2.3.1 Variables

For a given scenario s and a fixed stage 1 solution: $\hat{\delta}_{i,k,w}$, and $\hat{r}_{i,k} = r_{i,k}^{var} + r_{i,k}^{fixed}$, one can solve a MIP to get solution $x_{i,s,t}$, $p_{i,s,t}$, $p_{j,s,t}$. We denote by $[\mathbf{S}_{i,k}^{pc}, \mathbf{E}_{i,k}^{pc})$ the interval of period in which the k -th production campaign is taking place as induced by the $\hat{\delta}_{i,k,w}$. The formulation is based on generating feasible decreasing profile a priori for all feasible interval $[t_1, t_2)$ **under the assumption that the fuel stock is at level $\hat{\mathbf{BO}}_{i,k}$ when entering $\mathbf{DP}_{i,k}$** , where $\hat{\mathbf{BO}}_{i,k}$ is an estimation to fuel level to start the decreasing profile.

$\mathbf{DP}_{i,k}(t_1, t_2) = 1$ if in the planning of plant i the decreasing profile of cycle k starts at the beginning of period t_1 , with $t_1 \in [\mathbf{ES}_{i,k}^{dp}, \mathbf{LS}_{i,k}^{dp}]$, and ends on period $t_2 - 1$, with $t_2 \in [\mathbf{EE}_{i,k}^{dp}(t_1), \mathbf{LE}_{i,k}^{dp}(t_1)]$. To reduce the number of variables, one could consider only the value $t_2 = \mathbf{LE}_{i,k}^{dp}(t_1)$, i.e. carrying the decreasing profile to its end without going OFF line.

Associated Micro Variable Setting

- $x_{i,t_1}^{dp} = \hat{\mathbf{BO}}_{i,k}$ and $x_{i,t+1}^{dp} = \max(0, x_{i,t}^{dp} - \mathbf{D}^t \mathbf{p}_{i,t}^{dp})$ for $t = t_1 + 1, \dots, t_2$.
- $\mathbf{p}_{i,t}^{dp}$ is the value computed as a function of $x_{i,t}^{dp}$.

2.3.2 Formulation

Fuel stock

$$0 \leq x_{i,k,t} \leq \mathbf{SMAX}_{i,k}, \quad \forall i, k, t \in [\mathbf{S}_{i,k}^{pc}, \mathbf{E}_{i,k}^{pc}). \quad (90)$$

$$0 \leq x_{i,k,t} \leq \mathbf{AMAX}_{i,k+1}, \quad \forall i, k, t = \mathbf{E}_{i,k}^{pc}. \quad (91)$$

Type 2 Power production during Full Power

$$0 \leq pf_{i,k,t} \leq \mathbf{PMAX}_{i,t}, \quad \forall i, k, t \in [\mathbf{S}_{i,k}^{pc}, \mathbf{LE}_{i,k}^{fp} = \mathbf{ES}_{i,k}^{dp}). \quad (92)$$

Type 2 Power production during Decreasing Profile

$$0 \leq pd_{i,k,t} \leq \mathbf{PMAX}_{i,t}, \quad \forall i, k, t \in [\mathbf{ES}_{i,k}^{dp}, \mathbf{E}_{i,k}^{pc}). \quad (93)$$

Type 1 Power production

$$\mathbf{PMIN}_{j,t}^s \leq p_{j,t} \leq \mathbf{PMAX}_{j,t}^s, \quad \forall j, t. \quad (94)$$

Type 2 Modulation

$$0 \leq v_{i,k,t} \leq \mathbf{PMAX}_{i,t}, \quad \forall i, k, t \in [\mathbf{S}_{i,k}^{pc}, \mathbf{LE}_{i,k}^{fp} = \mathbf{ES}_{i,k}^{dp}). \quad (95)$$

Date of End of Full Power and Start of Decreasing Profile

$$z_{i,k,t} \in \{0, 1\}, \quad \forall i, k, t \in [\mathbf{ES}_{i,k}^{dp}, \mathbf{LS}_{i,k}^{dp}]. \quad (96)$$

$z_{i,k,t} = 1$ if i ends full power at the outset of period t ; the plant must then go into decreasing profile.

Decreasing Profile Transition

$$\mathbf{DP}_{i,k}(t_1, t_2) \in \{0, 1\}, \quad \forall i, t_1 \in [\mathbf{ES}_{i,k}^{dp}, \mathbf{LS}_{i,k}^{dp}] : t_1 < T, t_2 = \min(\mathbf{LE}_{i,k}^{dp}(t_1), \mathbf{E}_{i,k}^{pc}). \quad (97)$$

Demand satisfaction

$$\sum_j p_{j,t} + \sum_{i,k} pf_{i,k,t} + \sum_{i,k} pd_{i,k,t} = \mathbf{DEM}^{t,s}, \quad \forall t. \quad (98)$$

Selection of a date of end of full power

$$\sum_{t=\mathbf{ES}_{i,k}^{dp}}^{\mathbf{LS}_{i,k}^{dp}} z_{i,k,t} = 1, \quad \forall i, k. \quad (99)$$

Selection of a decreasing profile transition

$$\sum_{t_2} \mathbf{DP}_{i,k}(t, t_2) = z_{i,k,t}, \quad \forall i, k, t. \quad (100)$$

Variable bound on power production defining modulation

$$pf_{i,k,t} + v_{i,k,t} = \mathbf{PMAX}_{i,t} \left(1 - \sum_{\tau=\mathbf{ES}_{i,k}^{dp}}^t z_{i,k,\tau} \right), \quad \forall i, k, t \in [\mathbf{S}_{i,k}^{pc}, \mathbf{LS}_{i,k}^{dp}). \quad (101)$$

Maximum modulation

$$\sum_{t=\mathbf{S}_{i,k}^{pc}}^{\mathbf{E}_{i,k}^{pc}-1} v_{i,k,t} \leq \frac{\mathbf{MMAX}_{i,k}}{\mathbf{D}^t}, \quad \forall i, k. \quad (102)$$

Fuel stock initialisation at the ouset of a campaign

$$x_{i,-1,0} = \mathbf{X}\mathbf{I}_i, \quad \forall i. \quad (103)$$

$$x_{i,k,t_2} = \frac{(\mathbf{Q}_{i,k} - 1)}{\mathbf{Q}_{i,k}} x_{i,k-1,t_1} + \hat{r}_{i,k} + \mathbf{B}\mathbf{O}_{i,k} - \frac{(\mathbf{Q}_{i,k} - 1)}{\mathbf{Q}_{i,k}} \mathbf{B}\mathbf{O}_{i,k-1}, \quad \forall i, k \geq 0, t_1 = \mathbf{E}_{i,k-1}^{pc}, t_2 = \mathbf{S}_{i,k}^{pc}. \quad (104)$$

Fuel stock consumption dynamic

$$x_{i,k,t+1} = x_{i,k,t} - \mathbf{D}^t (pf_{i,k,t} + pd_{i,k,t}), \quad \forall i, k, t \in [\mathbf{S}_{i,k}^{pc}, \mathbf{E}_{i,k}^{pc}]. \quad (105)$$

Decreasing Profile Power Generation

$$pd_{i,k,t} = \mathbf{P}\mathbf{M}\mathbf{A}\mathbf{X}_{i,t} \left(\sum_{\tau=\mathbf{E}\mathbf{S}_{i,k}^{dp}}^t PB(\tau, t) z_{i,k,t} \right), \quad \forall i, k, t \in [\mathbf{E}\mathbf{S}_{i,k}^{dp}, \mathbf{E}_{i,k}^{pc}]. \quad (106)$$

Fuel stock lower bound at the ouset of the last period of full power

$$x_{i,k,t-1} \geq \mathbf{B}\mathbf{O}_{i,k} z_{i,k,t}, \quad \forall i, k, t \in (\mathbf{S}_{i,k}^{pc}, \mathbf{L}\mathbf{S}_{i,k}^{dp}]. \quad (107)$$

Fuel stock lower bound at the outset of decreasing profile

$$x_{i,k,t} \geq \hat{\mathbf{B}}\mathbf{O}_{i,k} z_{i,k,t}, \quad \forall i, k, t \in [\mathbf{E}\mathbf{S}_{i,k}^{dp}, \mathbf{L}\mathbf{S}_{i,k}^{dp}] : t < \mathbf{E}^{PC}. \quad (108)$$

where $\hat{\mathbf{B}}\mathbf{O}_{i,k}$ is an estimation to $\mathbf{B}\mathbf{O}_{i,k}$ (fuel level to start the decreasing profile).

Fuel stock lower bound at the outset of decreasing profile

$$x_{i,k,t} \leq \mathbf{S}\mathbf{M}\mathbf{A}\mathbf{X}_{i,k} + (\hat{\mathbf{B}}\mathbf{O}_{i,k} - \mathbf{S}\mathbf{M}\mathbf{A}\mathbf{X}_{i,k}) z_{i,k,t}, \quad \forall i, k, t \in [\mathbf{E}\mathbf{S}_{i,k}^{dp}, \mathbf{L}\mathbf{S}_{i,k}^{dp}]. \quad (109)$$

Objective

$$\min \sum_{j,t} \mathbf{C}_{j,t}^s \mathbf{D}^t p_{j,t} - \sum_i \mathbf{C}_{i,T} x_{i,K,T}. \quad (110)$$

The model will be updated for each scenario: $\mathbf{P}\mathbf{M}\mathbf{I}\mathbf{N}_{j,t}^s$, $\mathbf{P}\mathbf{M}\mathbf{A}\mathbf{X}_{j,t}^s$, $\mathbf{D}\mathbf{E}\mathbf{M}_{j,t}^{s,t}$, $\mathbf{C}_{j,t}^s$.

3 Selected Approach

For the class A instances of the Roadef Challenge, we implemented the global approach consisting in the phase 1 in one pass and the phase 2 in one pass. We obtain these results:

bench.	data0	data1	data2
OPT	8 730 985 367 093	169 538 386 903	146 048 433 973
sol.	8 740 938 710 031	170 150 546 138	146 953 050 287
gap	0.11%	0.36%	0.62%
time	00:00:01	00:02:16	00:06:06
bench.	data3	data4	data5
OPT	154 429 888 940	111 591 350 516	125 822 236 385
sol.	154 654 652 039	112 337 301 655	*
gap	0.15%	0.67%	*
time	00:07:18	00:24:24	*

Table 3: Computational results on 6 instances of the qualification phase

*
* *

Sommaire

1	Introduction	168
2	An overview of column generation based heuristics	168
3	Diving heuristics	169
4	Computational results	171
	Bibliography	172

ANNEXE B

Column Generation based Primal Heuristics

C. Joncour, S. Michel, R. Sadykov, D. Sverdlov, F. Vanderbeck

In the past decade, significant progress has been achieved in developing generic primal heuristics that made their way into commercial mixed integer programming (MIP) solver. Extensions to the context of a column generation solution approach are not straightforward. The Dantzig-Wolfe decomposition principle can indeed be exploited in greedy, local search, rounding or truncated exact methods. The price coordination mechanism can bring a global view that may be lacking in some “myopic” approaches based on a compact formulation. However, the dynamic generation of variables requires specific adaptation of heuristic paradigms. The column generation literature reports many application specific studies where primal heuristics are a key to success. There remains to extract generic methods that could be seen as black-box primal heuristics for use across applications. In this paper we review generic classes of column generation based primal heuristics. We then focus on a so-called “diving” method in which we introduce diversification based on Limited Discrepancy Search. While being a general purpose approach, the implementation of our heuristic illustrates the technicalities specific to column generation. The method is numerically tested on variants of the cutting stock and vehicle routing problems.

1 Introduction

Heuristics are algorithms that attempts to derive “good” primal feasible solutions to a combinatorial optimization problem. They include constructive methods that build a solution and improvement methods such as local search procedure that starts with an incumbent. The term “primal heuristic” generally refers to methods based on the tools of exact optimization, truncating an exact procedure or constructing solutions from the relaxation on which the exact approach relies: techniques range from greedy constructive procedures to rounding a solution of the linear programming (LP) relaxation, using the LP solution to define a target, or simply exploiting dual information for pricing choices. Alternatively, exact solvers can be used as subroutines in building heuristic solutions, for instance to explore a neighborhood in a local search procedure. Today’s MIP solvers rely heavily on generic primal heuristics: high quality primal values help pruning the enumeration by bound and preprocessing; they are also essential in tackling large scale real-life applications where the exact solver is given limited running time.

Heuristics based on exact methods have found a new breath in the recent literature. The latest developments are reviewed in [B.5]. Let us just mention a few landmarks: the *Large Scale Neighborhood Search* [B.3], the *Relaxation Induced Neighborhood Search* [B.8], the *local branching* heuristic [B.12], the *feasibility pump algorithm* [B.1, B.11]. Meta-heuristic paradigms such as oscillation between intensification and diversification of the search, and use of historical memory have also inspired progress in primal heuristics. Recently such work has been extended from the context of binary integer programs to general integer programs. Our purpose is to examine possible extensions to the case where one works with a Dantzig-Wolfe reformulation of the problem involving an exponential number of columns. The above mentioned landmark heuristics have not been applied directly to the Dantzig-Wolfe reformulation because setting bounds on column values can hardly been implemented in a context of dynamic column generation (the pricing problem ignores such bounds; modifying it to enforce such bounds typically induces an harder to solve subproblem). Alternatively, one could potentially develop an implementation of classic primal heuristics based on projecting the master solution in the original variable space. But, in some applications, there might not exist a bijective relation between compact formulation and Dantzig-Wolfe reformulation. Here, we consider how one can develop constructive or improvement heuristics specifically for the column generation context.

2 An overview of column generation based heuristics

Assume a mixed integer program (IP):

$$[P] \quad \min\{cx : Ax \geq a, \underbrace{Bx \geq b, x \in \mathbb{R}_+^n \times \mathbb{Z}_+^p}_{x \in X}\}$$

where a subset of constraints $Bx \geq b$ defines a subsystem X over which optimization is “relatively easy” while $Ax \geq a$ represent “complicating constraints”. For many structured applications, $Bx \geq b$ has a *block diagonal* structure with identical blocks. The structure of $[P]$ can exploited by reformulating it as a master program:

$$[M] \quad \min\{\sum_{g \in G} (cx^g)\lambda_g : \sum_{g \in G} (Ax^g)\lambda_g \geq a; \sum_{g \in G} \lambda_g = K \quad \forall k; \lambda_g \in \mathbb{N} \quad \forall g\}$$

where G is the set of *generators* of X and K is the number of identical blocks in $Bx \geq b$. We assume a bounded subsystem, thus G is an enumeration of the feasible solutions to X , i.e. $X = \{x^g\}_{g \in G}$. $|G|$ is typically exponential in the input size. Reformulation $[M]$ is solved by branch-and-price: at each node of the branch-and-bound tree the linear relaxation of $[M]$ is solved by column generation with a pricing problem of the form: $\min_{x \in X} \{(c - \pi A) x\}$, where π is the dual solution associated with constraints $Ax \geq a$ in $[M]$.

The most commonly used generic primal heuristic in this column generation context is the so-called *restricted master heuristic*. The column generation formulation is restricted to a subset of generators $\bar{G}$ and associated variables, and it is solved as a static IP. The restricted set $\bar{G}$ can either be generated heuristically, or be made of the columns generated during the master LP solution, or a mixture of both. The main drawback of this approach is that the resulting restricted master integer problem is often infeasible (the columns of $\bar{G}$ – typically defined by the LP solution – may not be combined to an integer solution). In an application specific context, an ad-hoc recourse can be designed to “repair” infeasibility. Such implementation has been developed for network design [B.6], vehicle routing [B.2, B.7, B.20] problems.

Simple *greedy heuristic* strategies have also been used: one iteratively adds a greedy selected column (the one with the best so-called “pseudo-cost”) to the partial solution until feasibility is reached. Column selection criteria can make use of the pricing procedure: for instance selecting columns that have the smallest ratio of reduced cost per unit of constraint satisfaction (based on dual price estimates that can be re-evaluated in the course of the algorithm). One can also use the master LP solution as a base for column selection. This gives rise to so-called *rounding heuristics*. A standard rounding strategy for the common set covering type master formulation consists of 3 steps: (i) an initial partial solution is obtained by rounding downwards the master LP solution, (ii) columns whose LP values are closest to their rounded part are then considered for round up while feasible, (iii) an ad-hoc constructive procedure is used to complete the solution. *Local search* can be used to improve the solutions while implementing some form of diversification. Typically one will remove some of the columns selected in the primal solution and reconstruct a solution with one of the above techniques. Greedy and rounding heuristics (sometimes coupled with local search) have been successfully applied (f.i in [B.4, B.19]). However, reaching feasibility remains a difficult issue that is often handled in an application specific manner along with greedy selection of columns, rounding directions or neighborhood definition. Deriving general purpose approaches is difficult. A generic way to complete a partial solution in search for feasibility is to generate further columns through pricing. This is precisely the strength of *diving heuristics*.

3 Diving heuristics

A *diving heuristic* is a depth-first heuristic search in the branch-and-price tree. At each node, a branch is heuristically selected based on greedy or rounding strategies. After branching the master LP is re-optimized (exactly or approximately). The branching rule used in such context is typically quite different from the one used in an exact approach: in a diving heuristic, one is not concerned with balancing the tree and one can overcome issues of variable fixing that are not compatible with column generation. Generating new columns in the process of re-optimization is an important feature for the success of the approach as it allow to construct feasible solutions. Observe that fixing a partial solution

does not impair column generation: the residual master can be tackled in the same way as the original master LP as no specific bounds on master variables have been set. In particular, columns previously selected in the partial solution remain in the formulation and may be selected again at a further stage.

Diving heuristics admit many possible implementation variants as illustrated in previous application specific studies: they were used on vehicle routing [B.15, B.18], crew rostering [B.14, B.13], cutting stock [B.9, B.19], and lot-sizing [B.9] problems. A key feature is the column selection mode that drives the heuristic: using greedy, random or rounding strategies (rounding down, up, to the closest integer, or based on a threshold [B.9, B.13, B.15, B.18]). Note that several columns can be taken into the solution simultaneously, and a constructive heuristic can be applied in an attempt to complete the solution at any stage [B.4, B.17]. A template of such diving procedure is given in Table 1.

Algorithm 1

Step 1: Solve the current master LP.

Step 2: Select columns into the current solution at heuristically set values.

Step 3: If the partial solution defines a complete primal solution, record this solution and goto *Step 6*.

Step 4: Update the master (right hand side) and the pricing problems (setting new bounds on subproblem variables to generate proper columns).

Step 5: If residual master problem is shown infeasible through preprocessing, goto *Step 6*. Else, return to *Step 1*.

Step 6: Stop.

Table 1: Depth first search diving heuristic

In our implementation, columns are selected one at the time in Step 2, selecting in the current master LP solution the λ_g variable value closest to integer (among roundings that are feasible for the master program). The selected column g is taken in the solution at a value equal to the closest integer to λ_g . An original feature not found in the above mentioned references is a limited backtracking that implements a diversification mechanism. The solution obtained through the initial depth-first exploration of the tree is considered as a reference solution around which we implement a *Limited Discrepancy Search (LDS)* [B.16]. The latter is controlled by two parameters: **maxDepth** and **maxDiscrepancy**: up to depth **maxDepth**, we considered to deviate from the reference solution in up **maxDiscrepancy** ways. Specifically, in Step 2, we avoid selecting columns present in a tabu list (of size $\leq$ **maxDiscrepancy**) that consists of columns selected in previous branches from which we wish to diversify the search. In step 6, we backtrack, while the current depth is greater than **maxDepth** or the current Discrepancy level = **maxDiscrepancy**. If such backtracking is not possible, we stop. Otherwise, we create a new branch defined by a tabu list made of columns that were tabu at the ancestor node or were selected in previous child nodes of the ancestor node. The resulting exploration tree is illustrated in Figure 1 for two parameter settings.

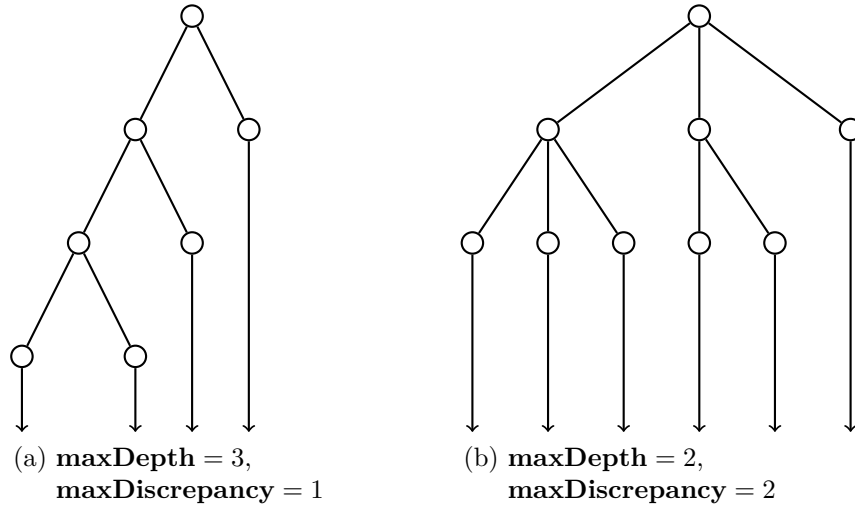


Figure 1: Illustration of a tree search with bounded depth and discrepancy

4 Computational results

We have build our diving heuristic into *BaPCod* [B.21], a generic Branch-and-Price Code that we developed. To test its usefulness, we compare the exact branch-and-price algorithm with and without the use of the primal heuristic at the root node, and the use of the diving heuristic only. Table 2 presents computational results on standard instances for the cutting stock problem (CSP) [B.22], bin packing problem with conflicts (BPWC) [B.10] (considering “hard instances” with a conflict graph of density of 10 to 40%), the graph vertex coloring problem (VCP) and classical vehicle routing problem (VRP) using the class A instances of <http://neo.lcc.uma.es/radi-aeb/webvrp/>. For **maxDepth** = 3 and **maxDiscrepancy** = 2, Table 2 reports the problem class, the instances size, the number of instances tested (#inst.), the number of unsolved instances within 30 minutes (#unsolv.), the average size of the branch-and-price tree (#nodes, using the branching scheme of [B.23]), the average solution time (in seconds). For the pure heuristic (DH only), we report the number of instances for which the solution found is not optimal (#unsolv.) and the average gap of the solution found with the optimum. Even when the DH solution has no gap with the optimum, it can have a gap with the column generation dual bound and hence no prove of optimality is known, which induces branching.

problem	size	#inst.	pure B-a-P			DH + B-a-P			DH only		
			#unsolv.	#nodes	time	#unsolv.	#nodes	time	#unsolv.	gap	time
CSP	50	10	0	245	26				0	0%	2
CSP	80	10	0	417	192				0	0%	14
BPWC	60–500	280	9	148	167	0	2.9	102	2	0.02%	75
VCP	11–211	18	0	17	116	0	1.2	34	0	0%	26
VRP	An32-80	27							17	5.25%	70

Table 2: Comparing branch-and-price with and without our diving heuristic (DH).

Bibliography

- [B.1] Tobias Achterberg and Timo Berthold. Improving the feasibility pump. *Discrete Optim.*, 4(1):77–86, 2007.
- [B.2] Y. Agarwal, K. Mathur, and H. M. Salkin. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks*, 19(7):731–749, 1989.
- [B.3] Ravindra K. Ahuja, Özlem Ergun, James B. Orlin, and Abraham P. Punnen. A survey of very large-scale neighborhood search techniques. *Discrete Appl. Math.*, 123(1-3):75–102, 2002. Workshop on Discrete Optimization, DO’99 (Piscataway, NJ).
- [B.4] G. Belov and G. Scheithauer. A cutting plane algorithm for the one-dimensional cutting stock problem with multiple stock lengths. *European J. Oper. Res.*, 141(2):274–294, 2002. Cutting and packing.
- [B.5] Timo Berthold. Primal Heuristics for Mixed Integer Programs. Master’s thesis, Technische Universität Berlin, 2006.
- [B.6] A. Chabrier. Heuristic branch-and-price-and-cut to solve a network design problem. In *Proceedings CPAIOR*, Montreal, Canada, may 2003.
- [B.7] A. Chabrier, E. Danna, and C. Le Pape. Coopération entre génération de colonnes et recherche locale appliquées au problème de routage de véhicules. In *Huitièmes Journées Nationales sur la résolution de Problèmes NP-Complets (JNPC)*, pages 83–97, Nice, France, may 2002.
- [B.8] Emilie Danna, Edward Rothberg, and Claude Le Pape. Exploring relaxation induced neighborhoods to improve MIP solutions. *Math. Program.*, 102(1, Ser. A):71–90, 2005.
- [B.9] Zeger Degraeve and Raf Jans. A new Dantzig-Wolfe reformulation and branch-and-price algorithm for the capacitated lot-sizing problem with setup times. *Oper. Res.*, 55(5):909–920, 2007.
- [B.10] Albert E. Fernandes Murtitaba, Manuel Iori, Enrico Malaguti, and Paolo Toth. Algorithms for the bin packing problem with conflicts. *INFORMS Journal on Computing*, page ijoc.1090.0355, 2009.
- [B.11] Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. *Math. Program.*, 104(1, Ser. A):91–104, 2005.
- [B.12] Matteo Fischetti and Andrea Lodi. Local branching. *Math. Program.*, 98(1-3, Ser. B):23–47, 2003. Integer programming (Pittsburgh, PA, 2002).
- [B.13] Michel Gamache, François Soumis, Gérald Marquis, and Jacques Desrosiers. A column generation approach for large-scale aircrew rostering problems. *Oper. Res.*, 47(2):247–263, 1999.
- [B.14] Martin Grötschel, Ralf Borndörfer, and Andreas Löbel. Duty scheduling in public transit. Jäger, Willi (ed.) et al., Mathematics—key technology for the future. Joint projects between universities and industry. Berlin: Springer. 653-674 (2003)., 2003.
- [B.15] Oktay Gunluk, Tracy Kimbrel, Laszlo Ladanyi, Baruch Schieber, and Gregory B. Sorkin. Vehicle Routing and Staffing for Sedan Service. *Transportation Science*, 40(3):313–326, 2006.
- [B.16] William D. Harvey and Matthew L. Ginsberg. Limited discrepancy search. In *Proc. IJCAI-95, Montreal, Quebec*, pages 607–613. Morgan Kaufmann, 1995.

- [B.17] Krzysztof C. Kiwiel. An Inexact Bundle Approach to Cutting-Stock Problems. *INFORMS Journal on Computing*, page ijoc.1090.0326, 2009.
- [B.18] Sophie Michel. *Optimisation des tournées de véhicules combinées à la gestion de stock*. PhD thesis, Université Bordeaux 1, France, 2006.
- [B.19] Nancy Perrot. *Integer Programming Column Generation Strategies for the Cutting Stock Problem and its Variants*. PhD thesis, Université Bordeaux 1, France, 2005.
- [B.20] É. D. Taillard. A heuristic column generation method for the heterogeneous fleet VRP. *RO Oper. Res.*, 33(1):1–14, 1999.
- [B.21] F. Vanderbeck. Bapcod - a generic branch-and-price code, 2008. <http://wiki.bordeaux.inria.fr/realopt/>.
- [B.22] François Vanderbeck. Computational study of a column generation algorithm for bin packing and cutting stock problems. *Math. Program.*, 86(3, Ser. A):565–594, 1999.
- [B.23] François Vanderbeck. Branching in branch-and-price: a generic scheme. *Mathematical Programming*, 2010. doi 10.1007/s10107-009-0334-1.

*
* *

Problèmes de placement 2D et application à l'ordonnancement : modélisation par la théorie des graphes et approches de programmation mathématique

Résumé :

Le problème de placement sur deux dimensions consiste à décider s'il existe un rangement d'objets rectangulaires dans une boîte donnée. C'est un problème combinatoire difficile (à la complexité du respect des capacités s'ajoute celle du positionnement des objets). Nous considérons les variantes sans rotation des objets et avec ou sans optimisation de la valeur des objets placés. Nous menons une étude exploratoire des méthodologies qui peuvent être développées à l'interface de la programmation mathématique, de l'optimisation combinatoire et de la théorie des graphes. Nous comparons les formulations de la littérature et en proposons de nouvelles. Nous développons et testons deux approches de résolution innovantes. L'une est basée sur la décomposition de Dantzig-Wolfe (avec un branchement sur les contraintes disjonctives de non recouvrement des objets). L'autre constitue en une approche combinatoire basée sur diverses caractérisations des graphes d'intervalles (modélisant le chevauchement des objets selon chaque axe).

Mots-clefs : problèmes de placement, théorie des graphes, génération de colonnes, modèles mathématiques, étude de branchement, graphes d'intervalles

2D-orthogonal packing and scheduling problems: modelling by graph theory and mathematical programming approaches

Abstract :

The two dimensional orthogonal packing problem consists in deciding whether there exists a packing of rectangular items in a given bin. This is a hard combinatorial problem (In addition to capacity constraints add the items positioning complexity). In this thesis, we consider the case where item rotation is not allowed and with or without packing value optimization. In this study, we compare literature formulations and we propose new ones. We develop and test two resolution approaches. The first is based on Dantzig-Wolfe decomposition associated with a branching on no-overlapping disjunctive constraints. The second establish a combinatorial approach based on multiple interval graph characterization (modelling the item no-overlapping according to each axis).

Keywords : orthogonal packing problems, graph theory, column generation, mathematical models, branching study, interval graph
