

Titre: Un modèle biniveau pour la gestion du revenu dans l'industrie des médias
Title:

Auteur: François Guay
Author:

Date: 2010

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Guay, F. (2010). Un modèle biniveau pour la gestion du revenu dans l'industrie des médias [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.
Citation: <https://publications.polymtl.ca/358/>

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

URL de PolyPublie: <https://publications.polymtl.ca/358/>
PolyPublie URL:

Directeurs de recherche: Gilles Savard, & Patrice Marcotte
Advisors:

Programme: Mathématiques appliquées
Program:

UNIVERSITÉ DE MONTRÉAL

UN MODÈLE BINIVEAU POUR LA GESTION DU REVENU DANS
L'INDUSTRIE DES MÉDIAS

FRANÇOIS GUAY
DÉPARTEMENT DE GÉNIE INDUSTRIEL ET MATHÉMATIQUES
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE
MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE INDUSTRIEL ET MATHÉMATIQUES)

JUIN 2010

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

UN MODÈLE BINIVEAU POUR LA GESTION DU REVENU DANS
L'INDUSTRIE DES MÉDIAS

présenté par : M. GUAY François, B.Ing.

en vue de l'obtention du diplôme de : Maîtrise ès Sciences Appliquées

a été dûment accepté par le jury constitué de :

M. LABIB Richard, Ph.D., président.

M. SAVARD Gilles, Ph.D., membre et directeur de recherche.

M. MARCOTTE Patrice, Ph.D., membre et co-directeur de recherche.

M. ROUSSEAU Louis-Martin, Ph.D., membre.

*À mes parents ainsi que toute ma famille,
ma petite amie, et tous mes amis.*

Remerciements

Je veux dans un premier temps témoigner de toute ma reconnaissance envers mes directeurs de recherche, Gilles Savard et Patrice Marcotte. Je les remercie pour leurs conseils, leurs idées, leur disponibilité, ainsi que leur passion pour la recherche qu'ils m'ont transmise.

Je veux également remercier Richard Labib et Louis-Martin Rousseau pour avoir relu ce mémoire, et m'avoir aidé à le rendre plus clair.

J'aimerais remercier mes collègues du GERAD, Aimé Kamgaing Kuiteing et Yousef Maknoon, pour l'aide technique qu'ils m'ont apportée, que ce soit pour mes travaux de recherche, ou pour la rédaction de ce mémoire. Un grand merci aussi à Nicolas Chan, pour son temps et sa patience.

Je remercie Anne-Sophie Touret et Mathilde Touret pour avoir relu ce mémoire et l'avoir rendu plus intelligible.

Enfin, je voudrais remercier l'École Polytechnique de Montréal, sans qui ce mémoire n'aurait jamais vu le jour, et le GERAD, pour ses locaux mis à disposition ainsi que son personnel toujours prêt à aider.

Résumé

Dans ce mémoire, nous nous intéressons à l'industrie des médias, et plus particulièrement à l'allocation de l'espace publicitaire et au problème de tarification qui y est rattaché.

Ce sujet est complexe : les chaînes de télévision font face à plusieurs difficultés lors de la vente de leur espace publicitaire à des agences de publicité. D'une part, les chaînes doivent savoir quels clients choisir lors de la vente de leur inventaire. D'autre part, une tarification adéquate de cette espace est inéluctable si celles-ci veulent rester compétitives.

Pour la première fois, ce problème typique de gestion du revenu a été abordé avec la programmation biniveau, mettant en jeu un modèle "M-Spot" dans lequel la chaîne télévisuelle propose et vend aux clients des spots publicitaires. Cependant, le modèle M-Spot présente deux défauts principaux : seules des instances de faible taille peuvent être résolues, et lors des résolutions numériques l'allocation est souvent peu réaliste.

Reprenant ce modèle, nous proposons un autre programme biniveau basé sur la notion de *patron* de spot publicitaire. De même que pour le problème de la découpe des rouleaux, qui utilise des patrons pour découper les rouleaux de tissus, nous découpons l'espace publicitaire des chaînes de télévision à l'aide de patrons de spots. Celles-ci vendent alors des assortiments de spots publicitaires, plutôt que des spots individuels. Nous appelons ce modèle "M-Patron".

Nous reprenons un algorithme développé pour le modèle M-Spot, qui est fondé sur la génération de coupes. Nous l'adaptions à notre modèle pour la résolution numérique, et nous développons une variante de cet algorithme, basée sur une génération additionnelle de coupes dès la première itération. Ces deux algorithmes donnent des résultats satisfaisants, et permettent de programmer et tarifier de manière optimale une soirée

typique de diffusion de spots publicitaires à la télévision.

Mots-clefs : Publicité, gestion du revenu, chaîne de télévision, patron, plans coupants, programmation biniveau, sac-à-dos.

Abstract

In this master's thesis, we are interested in the media industry, particularly in media broadcasting and the associated pricing problem.

TV Channels face many problems when they sell their inventory to media agencies. On one hand, they have to choose which commercials should be aired. On the other hand, they must determine the prices of these commercials, while staying competitive.

For the first time, this revenue management problem was solved thanks to bilevel programming. In this model (called "M-Spot"), the TV channel offers and sells commercial spots to clients. Even if bilevel programming is very convenient as a paradigm for this problem, the model fails to tackle large-scale instances and proposes unrealistic solutions.

Using this model, we present another bilevel program, based on patterns of commercial spots, which we call "M-Pattern". As in the *cutting stock problem*, we use patterns to cut media inventories. This way, a pattern can be considered as a set of commercials, and TV channels sell selection of commercial spots, whereas in the M-Spot they sell single commercials.

We take up an algorithm developed for the M-Spot model, based on cut generation. We adapt this algorithm to our model, and introduce additional cuts in a pre-processive step. Both algorithms allow to optimally schedule and price patterns of commercial spots for a typical prime time on TV.

Keywords: Advertising, revenue management, bilevel programming, commercial, pattern, knapsack, branch-and-cut.

Table des matières

Dédicace	iv
Remerciements	v
Résumé	vi
Abstract	viii
Table des matières	ix
Liste des tableaux	xii
Liste des figures	xiv
Liste des annexes	xv
Liste des sigles et abréviations	xvi
Chapitre 1 INTRODUCTION	1
1.1 L'industrie de la publicité à la télévision	1
1.2 Approche utilisée	5
Chapitre 2 REVUE DE LITTÉRATURE	8
2.1 La gestion du revenu	8
2.2 L'industrie des médias	10
2.3 La programmation biniveau	13
Chapitre 3 MODÉLISATION DU PROBLÈME DU PLACEMENT MÉDIA	18
3.1 Le modèle par spots	18
3.1.1 Généralités sur le modèle	18

3.1.2	Les notations du modèle M-Spot	19
3.1.3	Le programme mathématique M-Spot	21
3.2	Hypothèses du modèle M-Patron	23
3.3	Modèle mathématique	24
3.3.1	Notations relatives au modèle	24
3.3.2	Les fonctions et les contraintes du modèle	26
3.3.3	Le modèle mathématique	28
3.4	Un exemple	28
3.4.1	Ensemble Minimal	29
3.4.2	L'exemple	30
Chapitre 4 ALGORITHMES DE RÉOLUTION		36
4.1	Algorithme initial	36
4.1.1	Cadre de l'algorithme	36
4.1.2	Le problème relaxé initial	37
4.1.3	Le problème relaxé modifié et la génération des coupes	40
4.2	Génération initiale de coupes	43
4.3	Algorithme de relaxation sous-optimale	46
Chapitre 5 RÉOLUTION NUMÉRIQUE		49
5.1	Les paramètres de simulation	49
5.1.1	Choix de génération de données	49
5.1.2	Les paramètres du modèle	53
5.1.3	La publicité à la télévision	57
5.2	Tests numériques initiaux	58
5.2.1	Résultats numériques du modèle M-Spot	58
5.2.2	Le jeu de données initial	60
5.2.3	Commentaires sur le comportement de l'algorithme A0	63
5.2.4	Algorithme A1	64

5.2.5	Algorithme de Relaxation Sous-optimale A_2	66
5.3	Plan d'expérimentation	67
5.3.1	Les paramètres déterminant la capacité du leader	67
5.3.2	Les paramètres associés aux clients	72
5.3.3	Les paramètres associés aux patrons	76
5.3.4	Conclusion sur le plan d'expérimentation	79
5.4	Les pertes de revenu occasionnées par certaines hypothèses du modèle	79
5.5	Simulation d'un <i>prime time</i>	82
Chapitre 6 CONCLUSION		84
Références		87
Annexes		91

Liste des tableaux

TABLEAU 1.1	Les 10 catégories d'annonceurs les plus importantes en 2009 et 2008 (www.tvb.org)	3
TABLEAU 3.1	Audiences et désutilités pour le leader	31
TABLEAU 3.2	Audiences, désutilités et prix des patrons de la concurrence	31
TABLEAU 3.3	Les ensembles minimaux et leurs prix	33
TABLEAU 5.1	$C = 5 - L = 3 - D = 1 - U^1 = 10 - 6$ à 8 spots pour combler l'audimat des clients	59
TABLEAU 5.2	Temps de résolution de l'algorithme A_0 testé sur le jeu de données initial - 2 à 3 patrons pour combler l'audimat des clients	62
TABLEAU 5.3	Temps de résolution pour le M-Spot et le M-Patron (pour des instances différentes).	63
TABLEAU 5.4	Correspondances clients/patrons	65
TABLEAU 5.5	Temps de résolution des algorithmes A_0 et A_1 pour le jeu de données initial - 2 à 3 patrons pour combler l'audience des clients	65
TABLEAU 5.6	Temps de résolution des algorithmes A_0 et A_2 pour le jeu de données initial - 2 à 3 patrons pour combler l'audience des clients	68
TABLEAU 5.7	Paramètres de capacité du leader - Variation de la longueur $l(s)$ des pauses s	70
TABLEAU 5.8	Paramètres de capacité du leader - Variation de U^1	70
TABLEAU 5.9	Paramètres de capacité du leader - Variation de l'indice $I_{s,l}^1$	70
TABLEAU 5.10	Paramètres de capacité du leader - Variation de l	71
TABLEAU 5.11	Paramètres des clients - Variation du nombre de clients	72

TABLEAU 5.12	Paramètres des clients - Variation de l'audimat par client A_c	73
TABLEAU 5.13	Paramètres des clients - Variation des catégories de clients	76
TABLEAU 5.14	Paramètres des patrons - Variation du nombre de patrons P^1 et P^2	77
TABLEAU 5.15	Paramètres des patrons - Variation de P_c^*	78
TABLEAU 5.16	Résultats du plan d'expérimentation	80
TABLEAU 5.17	Perte de revenu sur des très petites instances	81
TABLEAU 5.18	Perte de revenu sur des petites instances	81
TABLEAU 5.19	Comportement des deux algorithmes lors d'un <i>prime time</i> .	83
TABLEAU B.1	4 Instances difficiles résolues avec Cplex et Xpress	94
TABLEAU B.2	Temps de résolution des 10 instances avec Cplex et Xpress .	96
TABLEAU B.3	Moyenne des temps et écarts-type avec Cplex et Xpress . . .	96

Liste des figures

FIGURE 1.1	Pauses et spots publicitaires	3
FIGURE 3.1	Schéma de l'exemple : architecture des patrons	31
FIGURE 3.2	Ensemble des polyèdres d'optimalité pour le M-Patron	35
FIGURE 5.1	Lors du <i>prime time</i> aux États-Unis	58
FIGURE A.1	Ensemble des polyèdres d'optimalité pour le M-Spot	92

Liste des annexes

ANNEXE A	Exemple numérique pour le M-Spot	91
ANNEXE B	Temps de résolution avec Cplex	93

Liste des sigles et abréviations

PIB Produit Intérieur Brut

FCC Federal Communication Commission

KKT Karush-Kuhn-Tucker

MIP Mixed-Integer Program (Programme mixte en nombres entiers)

Chapitre 1

INTRODUCTION

La création du premier poste de télévision en 1926, suivie de la démocratisation de cette invention dans les années 70, a révolutionné le monde de la publicité. Aujourd'hui, l'impact de la publicité à la télévision est très important puisqu'une grande partie de la population mondiale possède un téléviseur.

Ce mémoire considère le problème du placement publicitaire dans les médias qui, une fois résolu, permettra à une chaîne de télévision de gérer son espace publicitaire. Dans cette introduction, nous présentons dans un premier temps l'industrie de la publicité à la télévision. Nous exposons ensuite l'approche que nous utiliserons afin de résoudre le problème du placement média.

1.1 L'industrie de la publicité à la télévision

Les chaînes de télévision consacrent un quart de leur temps de diffusion à la publicité, et la somme d'argent qu'elles perçoivent en échange de la transmission de spots publicitaires leur assure 90% de leur revenu. En effet, les revenus des publicités diffusées pour la coupe du monde de soccer, ou lors de séries à succès (*Friends*, *How I Met Your Mother*) peuvent atteindre des sommes considérables. Par exemple, la chaîne de télévision américaine CBS a généré des revenus de 243,6 millions de dollars pour la soirée du Superbowl qui s'est déroulée le 5 février 2007, allant même jusqu'à vendre 2,7 millions de dollars certains spots publicitaires de 30 secondes (MacMillan et Lehman (2008)).

De ce fait, les montants engagés dans la publicité sont de plus en plus importants.

Aux États-Unis, où les sommes d'argent investies en publicité à la télévision constituent 2% du PIB (Anderson et Gabszewics (2006)), les trois plus gros annonceurs ont déboursé en 2009 respectivement 269 millions (Verizon), 237 millions (AT&T) et 229 millions de dollars (Ford). Le tableau 1.1 présente les sommes investies dans la publicité à la télévision par les dix plus grosses catégories d'annonceurs sur le réseau national entre 2008 et 2009 aux États-Unis¹.

Au final, la totalité de ces montants est de 50 milliards de dollars pour l'année 2009. Bien que ce chiffre ait quelque peu diminué depuis 2008 à cause de la crise financière, il tendra à réaugmenter avec la reprise.

Compte tenu des sommes en jeu, les chaînes de télévision ont à cœur de gérer au mieux leur espace publicitaire, et doivent alors faire face à deux dilemmes. Premièrement, elles sont tenues d'organiser la programmation des spots publicitaires de façon à satisfaire au mieux leurs clients. Deuxièmement, la tarification de ces spots doit maximiser leur revenu. Avant de décrire ce mécanisme, nous définissons deux notions qui seront fréquemment employées dans ce mémoire : la "pause publicitaire" et le "spot publicitaire". Une pause publicitaire est un intervalle de temps voué à la publicité pendant un programme (match de hockey, film) ou entre deux programmes. Un spot publicitaire est alors un espace de temps de la pause publicitaire alloué à un client. La figure 1.1 schématise ces deux notions, où les spots publicitaires "C" (constructeurs automobiles) et "T" (télécommunications) sont représentés respectivement en bleu et en rouge.

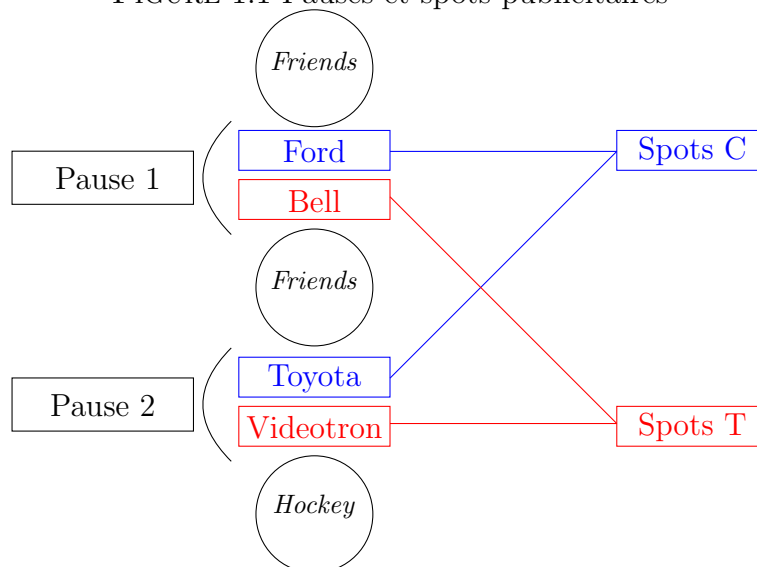
Les chaînes de télévision mettent en vente et négocient leur espace publicitaire chaque année. Aux États-Unis, l'espace temps réservé aux publicités durant la saison à venir est annoncé début mai, à l'occasion de la diffusion des grilles de programmes effectives à partir de septembre. Peu de temps après l'annonce, la vente des spots pu-

¹Toutes ces données sont disponibles sur le site internet www.tvb.org (Television Bureau Advertising), site qui regroupe plusieurs types de données concernant la publicité à la télévision aux États-Unis.

TABLEAU 1.1 Les 10 catégories d'annonceurs les plus importantes en 2009 et 2008
(www.tvb.org)

Rang	Catégorie	Année 2009 \$ millions	Année 2008 \$ millions
1	Automobile	1711,9	2926,2
2	Communications	1322,3	1463,1
3	Restaurant	1132,3	1291,2
4	Vendeur de voitures	539,6	804,9
5	Vendeur de mobilier	515,7	649,6
6	Assurances	495,7	616,2
7	Écoles et universités	446,7	481,5
8	Banques	440,6	543,8
9	Agroalimentaire	433,4	410,2
10	Services juridiques	411,3	410,9

FIGURE 1.1 Pauses et spots publicitaires



bilitaires commence, et elle constituera 90% du temps de diffusion de publicité total. Cette période dure généralement deux à trois semaines, et est plus communément appelée *Up-market*. Le temps de diffusion restant sera alors vendu dans le courant de l'année suivante, et ce marché est connu sous le nom de *Scatter-market*. Dans ce mémoire, nous ne considérerons que l'*Up-market*.

Lors de l'*Up-market*, les agences de publicité négocient avec les chaînes de télévision pour leur acheter du temps de diffusion (*audimat*) consacré à leurs clients. La demande de contrat ainsi établie inclut le prix, la catégorie démographique visée (par exemple, les compagnies automobiles se concentrent principalement sur des adultes entre 25 et 49 ans), et le nombre de personnes ciblées. La chaîne de télévision s'emploie à prédire l'audience pour chaque créneau horaire de la journée, en se basant sur les types d'émissions diffusées et sur les données historiques. Par exemple, l'audience et la catégorie démographique ne seront pas les mêmes un samedi matin lors des dessins animés, et un dimanche soir lors d'un match de hockey. Après avoir satisfait à tous ces paramètres, la chaîne télévisuelle doit planifier son allocation publicitaire afin de répondre aux critères et exigences de chacun de ses clients, tout en maximisant son revenu.

Cependant, cette planification peut s'avérer compliquée à mettre en œuvre. En effet, plusieurs contraintes entrent en jeu : la chaîne doit satisfaire l'*audimat* requis pour chaque client, tout en diffusant les spots publicitaires de manière à ne pas interférer avec d'autres spots publicitaires concurrents. Il est rare de constater deux spots relatifs à deux constructeurs automobiles dans la même pause.

À cela viennent s'ajouter des contraintes de temps, car chaque pause publicitaire a une durée bien déterminée. Les grilles de programmation étant définies à l'avance, les temps des pauses restent fixes. De plus, une compensation financière est assurée aux clients non satisfaits lorsque les spots qui leur sont vendus ne correspondent pas exactement à leurs attentes, ou lorsque leurs objectifs d'audience n'ont pas été

réalisés. Finalement, il est primordial de bien tarifier le prix des spots. Si le prix du spot est trop bas, la chaîne perd de l'argent. Si le prix est trop élevé, le client achètera alors un spot à la concurrence.

Traditionnellement (et encore maintenant), la planification était faite manuellement. Les employés préparaient un plan préliminaire avant le début des négociations, puis le révisaient des centaines de fois avant d'aboutir au plan définitif. Cette tâche mobilisait tout le personnel de la chaîne pendant deux à trois semaines, et ce des dizaines d'heures par jour. Maintenant certaines compagnies utilisent des outils de planification pour résoudre cet épineux problème. NBC Universal a mis au point un système qui lui permet d'économiser désormais 50 millions de dollars par année (Bollapragada *et al.* (2002)). Il est donc intéressant de s'intéresser à un système permettant d'allouer l'espace publicitaire tout en fixant des prix optimaux, et c'est dans cette optique que nous avons rédigé ce mémoire.

1.2 Approche utilisée

Chan (2008) ont été les premiers à aborder le problème du placement publicitaire à l'aide de la programmation biniveau. Nous appellerons "M-Spot" leur modèle dans la suite de ce mémoire.

Voici un bref aperçu de l'architecture de ce problème biniveau :

$$\begin{array}{l} \text{Meneur (chaîne tv)} = \left\{ \begin{array}{l} \text{Maximiser revenu} \\ \text{s.à contraintes d'inventaire} \\ \text{éviter les conflits} \end{array} \right. \\ \text{Suiveur (client)} = \left\{ \begin{array}{l} \text{Minimiser coûts} \\ \text{s.à objectif d'audience} \end{array} \right. \end{array}$$

La chaîne leader est le meneur dans ce problème, et elle veut maximiser son profit, tout en faisant face à des contraintes d’inventaire. D’une part, elle ne peut pas programmer plus de spots publicitaires qu’il n’y a de temps disponible pour chaque pause. D’autre part, la chaîne leader doit éviter les conflits entre certains spots publicitaires : elle désire par exemple garantir un espacement prédéfini entre deux spots publicitaires de même type, pour éviter la concurrence entre les annonceurs.

Les clients sont les suiveurs dans ce modèle. Ils minimisent leurs coûts, tout en atteignant leur objectif d’audience.

Cette représentation biniveau pour la gestion du revenu est assez commune, et elle peut rappeler les modélisations dans le transport aérien (Côté *et al.* (2007)) ou dans la tarification des réseaux (Marcotte et Savard (2002)).

Cependant, le modèle M-Spot rencontre deux problèmes majeurs. Premièrement, il ne parvient pas à résoudre des instances de grande taille. La plupart des instances résolues par Chan (2008) comportent cinq clients et cinq pauses publicitaires, ce qui reste limité pour l’adaptation du problème à l’industrie des médias. Deuxièmement, le M-Spot peut proposer des solutions irréalistes : en effet l’ajout de contraintes relatives à la programmation des spots n’a pas été possible car celles-ci complexifient considérablement le problème. Par exemple, ces contraintes pourraient forcer les spots attribués à un client à être placés sur des créneaux différents, ou espacés d’une certaine durée.

Nous proposons alors une variante du M-Spot, en redéfinissant des variables qui correspondent à des ensembles prédéterminés de spots que l’on appellera “patrons”. Un patron pourra alors être assimilé à un assortiment de spots, dont nous définirons nous-même l’architecture, et nous supposerons que la chaîne de télévision vend au client des patrons. Par exemple, un patron pourra contenir trois spots publicitaires se trouvant sur trois pauses successives. Nous baptiserons notre modèle “M-Patron”.

Il suffit de regarder la télévision pour concevoir la notion de patron. Force est de constater que lorsqu’on visionne certains programmes, plusieurs spots sont récurrents

lors des pauses publicitaires. Tous les soirs à la même heure il arrive de voir la même annonce publicitaire. Certaines entreprises conçoivent même spécialement des spots pour des événements sportifs (Jeux Olympiques par exemple), qu'elles veulent diffuser à chaque épreuve sportive. Généralement lors de la diffusion de séries ou lors de matchs de hockey, le même spot publicitaire est retransmis à toutes les pauses. Il se peut également que la même publicité revienne plusieurs fois dans un même après-midi. Cette observation nous a amenés à considérer un modèle utilisant des patrons.

Nous nous baserons sur le modèle biniveau développé par Chan (2008), et l'adapterons au M-Patron. En effet, les problèmes de type biniveau offrent un paradigme encore peu étudié dans le placement média.

Ce modèle permettra de déterminer l'allocation optimale ainsi que la tarification qui s'y rattache, et personne de nos jours n'a développé un tel modèle mathématique. Grâce à l'allocation simultanée de plusieurs spots, nous espérons réaliser la programmation de publicités pour une soirée. De plus, de par leur régularité, les patrons offrent un espacement certain entre les spots publicitaires, donc les solutions proposées par le M-Patron sont plus réalistes que celles proposées par le M-Spot. Nous mènerons également une analyse de sensibilité poussée sur l'ensemble des paramètres du modèle, étude jamais effectuée sur le sujet.

Nous effectuerons dans le chapitre suivant une revue de littérature, pour situer notre projet dans le monde de la recherche. Ensuite nous décrirons dans le chapitre 3 le modèle mathématique associé au problème biniveau. Dans le chapitre 4 nous détaillerons les algorithmes de résolution exacte développés pour résoudre le problème. Par la suite nous nous pencherons dans le chapitre 5 sur la résolution numérique qui met en jeu les algorithmes développés, et nous mènerons une analyse de sensibilité basée sur les paramètres du modèle. Enfin, nous proposerons en conclusion des ouvertures pour la poursuite du projet.

Chapitre 2

REVUE DE LITTÉRATURE

Nous présentons dans ce chapitre une revue de littérature afin d'exposer les informations relatives au sujet de ce mémoire. Dans la première section, nous nous intéressons à la gestion du revenu (connue en anglais sous le nom de *revenue management*). À la section 2.2, nous présentons les applications de la gestion du revenu en nous attardant sur les travaux réalisés dans l'industrie des médias. Finalement, la section 2.3 porte sur la programmation mathématique à deux niveaux, le paradigme utilisé pour modéliser notre problème.

2.1 La gestion du revenu

La gestion du revenu est une stratégie mise en place par une entreprise, dans le but de maximiser ses revenus ou ses profits. Étant devenu l'un des domaines de la recherche opérationnelle le plus étudié, la gestion du revenu s'applique aux industries qui disposent d'une capacité limitée et qui proposent des produits périssables. Elle permet aux gestionnaires de répondre à des questions telles que : “quels produits dois-je vendre?”, “à qui?”, “à quel prix?”, “de quelle manière?”. Heo et Lee (2009) décrivent la gestion du revenu comme une “stratégie de tarification pour contrôler les niveaux d'inventaires optimaux et prévoir une demande en temps réel”.

Son réel essor remonte plus précisément aux années 1980, suite à l'Acte de Dérégulation Aérienne aux États-Unis, comme le précisent Talluri et Van Ryzin (2004). Cette loi, signée en 1978, autorise les compagnies aériennes à fixer elles-mêmes les prix de leurs services. C'est en 1985 que les premières compagnies aériennes se sont

équipées de systèmes de gestion du revenu afin de faire face à cette situation nouvelle.

La clé de la gestion du revenu est de comprendre, anticiper et influencer le comportement des clients, c'est-à-dire le secteur de la demande. Les gestionnaires utilisent cette pratique pour prendre des décisions qui vont leur permettre de maximiser leurs revenus. Talluri et Van Ryzin (2004) distinguent de ce fait trois catégories de décisions au niveau de la gestion de la demande :

- **Décisions structurelles** : Quel format de vente utiliser (négociations, ventes aux enchères) ; comment segmenter le marché ; quelle politique d'achat/vente adopter.
- **Décisions sur les prix** : Comment déterminer les prix affichés ; comment tarifier dynamiquement ; comment appliquer des réductions.
- **Décisions sur les quantités** : Doit-on accepter ou rejeter une offre d'achat ; comment allouer la capacité à différents segments ou produits.

La gestion du revenu remporta un franc succès auprès des compagnies aériennes. Cette pratique a pris une telle ampleur dans le transport aérien, que les compagnies n'ayant pas choisi de l'adopter ont fait faillite. En effet son utilité dans le secteur aérien permet aux compagnies de déterminer le prix de leurs billets d'avion ainsi que leur allocation, tout en maximisant leurs revenus. Les billets d'avion sont des produits périssables et la capacité des avions est limitée ; ainsi la détermination du prix des places, la segmentation du marché et l'allocation de la capacité de l'avion jouent un rôle prépondérant dans la génération des revenus par l'entreprise. Côté *et al.* (2007) ont notamment travaillé sur le problème de l'allocation des places ainsi que la tarification des billets d'avion.

Après avoir été appliquée au transport aérien, la gestion du revenu a été adaptée à d'autres secteurs tels que l'hôtellerie et la location de véhicules. Cet art est devenu une stratégie essentielle pour plusieurs industries à capacité limitée, comme le décrivent

Chiang *et al.* (2007). Dans leur article, ils expliquent les différentes techniques de résolution associées aux problèmes de gestion du revenu, et proposent l'étude future de problèmes sous-jacents à cette pratique.

On peut compter parmi ceux-ci l'hôtellerie, qui dispose d'une capacité limitée et de produits périssables : la gestion du revenu s'avère donc utile pour savoir comment déterminer le prix des chambres ou les segments à viser. Donaghy *et al.* (1995) étudient le problème et donnent des pistes sur le sujet. Heo et Lee (2009) s'intéressent aux parcs d'attraction, spécifiant l'apport de la gestion du revenu pour cette industrie. Ils soulignent également l'importance de la bonne perception du grand public de la politique de gestion du revenu. Effectivement, si cette politique est mal perçue par les clients (sentie comme injuste, abusive), l'augmentation des revenus ne peut être que temporaire.

Marcotte et Savard (2002), pour leur part, se sont intéressés à la tarification des routes et des réseaux de communication, et proposent différentes modélisations et techniques de résolution adaptées aux problèmes considérés. Anjos *et al.* (2005), eux, proposent une famille de fonctions continues de tarification pour lesquelles une stratégie de tarification optimale peut être implémentée.

2.2 L'industrie des médias

Des sommes colossales sont dépensées par toutes les entreprises pour faire de la publicité, plus particulièrement à la télévision. Avec l'avènement d'internet, ces deux moyens de communication sont les plus convoités pour la diffusion de publicités. En effet, des publicités apparaissent récurrentement sur chaque page internet que l'on parcourt. Talluri et Van Ryzin (2004) décrivent l'approche "gestion du revenu" pour l'industrie des médias, et soulèvent parfaitement l'ensemble des problèmes liés à cette industrie, dont une partie a été traitée dans la littérature actuelle.

L'industrie des médias est très étudiée, et on retrouve dans la littérature scien-

tifique deux approches invoquant deux sciences : l'économie et la programmation mathématique. Bien qu'utilisant une grande part de mathématiques, la première approche, plutôt économique, cherche à obtenir des résultats qualitatifs concernant la structure du marché économique de la publicité. La deuxième approche, plutôt mathématique, propose des modèles déterministes et probabilistes pour obtenir des résultats quantitatifs : maximiser le revenu d'une chaîne de télévision et/ou l'audimat, c'est-à-dire le nombre de personnes qui regardent les spots publicitaires.

L'approche économique cherche à modéliser le marché de la publicité à la télévision, et à déterminer les paramètres les plus importants qui régissent ce marché. Brown et Alexander (2005) ont structuré le marché de façon à faire intervenir la tarification des spots, le nombre et le type de téléspectateurs, et l'allocation des licences données par la FCC. Ils sont arrivés à la conclusion suivante : entre 18h et 20h, une augmentation de 20% des téléspectateurs provoque une augmentation de 9% des revenus de la chaîne de télévision regardée. Leur modèle a également suggéré qu'une concentration plus forte de la catégorie des 18-34 ans pourrait faire monter le prix des spots visant cette catégorie démographique. Mangani (2003) démontre qu'une chaîne télévisuelle peut maximiser ses revenus en différenciant au minimum ses programmes, ou maximiser son audimat en étayant la différenciation des programmes. Dans ce cas précis, la différenciation concerne la variété et la diversité des programmes proposés.

Par ailleurs, d'autres chercheurs ont modélisé le marché de la publicité à la télévision grâce à la théorie des jeux. Anderson et Gabszewics (2006) développent un marché avec deux groupes (les téléspectateurs et les diffuseurs), qui interagissent via une plateforme, la chaîne de télévision. Les téléspectateurs trouvent la publicité nuisible, et plus elle est présente sur la chaîne, plus ceux-ci sont mécontents, alors qu'ils ne paient pas un prix direct pour regarder cette chaîne de télévision. Les annonceurs exercent alors une "externalité négative" sur les téléspectateurs. Ce paradigme est difficile à modéliser, et l'équilibre déterminé par le modèle est loin de la réalité, montrant alors une non-efficience du marché de la publicité. Ce modèle a également été étudié par

Peitz et Valletti (2008) ; ces derniers comparent le contenu des programmes pour une chaîne de télévision gratuite et une chaîne de télévision payante.

La programmation mathématique offre deux types de modèles, dont la modélisation déterministe. Cette approche traite uniquement de la vente des spots publicitaires sur l'*Up-market*. NBC, première chaîne de télévision à mettre au point un système d'optimisation du revenu, a collaboré avec Bollapragada *et al.* (2002) pour développer un modèle consacré à l'allocation de l'espace publicitaire dans le but de maximiser ses revenus. Pour cela, ils considèrent la requête d'un seul client pour lequel ils cherchent à définir l'ensemble optimal de spots publicitaires à vendre. Cette requête est étudiée de façon détaillée ; les objectifs d'audience du client, son budget, ses préférences de programmes et sa longueur de spot préférentielle sont considérés. Le modèle proposé est un programme linéaire résolu à l'aide d'une méthode tabou. Ce système est un succès, car il permet à NBC de gagner 50 millions de dollars supplémentaires chaque année depuis sa mise en service (Bollapragada *et al.* (2002)). Les autres modèles proposés dans la littérature abordent le problème d'une autre manière. Les clients dressent une liste de spots qu'ils souhaitent diffuser, et le modèle aide les dirigeants de la chaîne de télévision à décider dans quelle mesure elle doit accepter la demande des clients. Kimms et Müller-Bungart (2007) ont attaqué ce problème, et ont développé pas moins de cinq heuristiques pour le résoudre. Zhang (2006) s'y est intéressé également, et a proposé une résolution en deux étapes pour ce même problème d'allocation. Dans une première étape il sélectionne les annonceurs pour leur assigner un *show*, grâce à un algorithme de génération de colonnes. Dans une deuxième étape, il programme de manière optimale les spots des annonceurs sélectionnés.

Le deuxième type de modèle étudié grâce à la programmation mathématique est stochastique. Cette modélisation prend alors en compte l'incertitude sur les spots publicitaires et sur les résultats d'audience. Araman et Popescu (2007) développent un modèle qui tient compte de l'incertitude sur l'audimat espéré, tout en minimi-

sant l'espérance de la perte financière de la chaîne de télévision. Bollapragada *et al.* (2008) ont essayé d'anticiper cette incertitude sur la prévision de l'audimat, et ont implémenté la prédiction de la demande pour l'*Up-market* de NBC. Ce nouveau dispositif prévoit la demande avec précision, en combinant la méthode Delphi ainsi que des méthodologies de prévision appelées "Grass Roots". Ce système inédit produit une erreur de 2,8% une semaine avant l'ouverture de l'*Up-market*, alors que l'erreur oscillait d'ordinaire entre 5 et 12%.

Certains chercheurs s'emploient à utiliser l'approche probabiliste pour maximiser le nombre de personnes touchées par la publicité à la télévision, et se placent ainsi du point de vue de l'annonceur publicitaire. Mihiotis et Tsakiris (2004) cherchent la meilleure combinaison de placements de spots publicitaires de façon à obtenir le maximum d'audimat. Pérez-Gladish *et al.* (2010) s'appuient sur des recherches psychologiques montrant que l'impact des répétitions de spots sur les téléspectateurs croît en fonction du logarithme des répétitions de ces spots. Il développe alors un modèle qui aide les compagnies à optimiser l'impact de leur campagne publicitaire sur les chaînes de télévision.

Chan (2008) a été le premier à proposer un modèle biniveau pour le problème du placement publicitaire. Le grand avantage de son modèle réside dans le fait que les problèmes de tarification et d'allocation sont traités simultanément grâce à la programmation biniveau. Ce mémoire s'ancre dans la continuité de leurs travaux.

2.3 La programmation biniveau

Utilisée pour résoudre certains problèmes de gestion du revenu, la programmation biniveau implique deux problèmes d'optimisation où le domaine réalisable du premier niveau est implicitement déterminé par un autre problème d'optimisation. Comme le définit Dempe (2002), ce sont des problèmes hiérarchiques, où l'agent concerné par le premier niveau est appelé le meneur, et l'agent au second niveau est le suiveur. On

peut modéliser ces problèmes de la manière générale suivante :

$$\begin{aligned} \max_{x \in X, y} \quad & F(x, y) \\ \text{s.à} \quad & G(x, y) \leq 0 \\ & y \in \arg \min_z \{f(x, z) : g(x, z) \leq 0\}. \end{aligned}$$

Nous considérons la forme “verticale” du problème, plus simple et représentative de la structure hiérarchique du problème :

$$\begin{aligned} \max_{x \in X, y} \quad & F(x, y) \\ \text{s.à} \quad & G(x, y) \leq 0 \\ & \min_y f(x, y) \\ & \text{s.à} \quad g(x, y) \leq 0. \end{aligned}$$

Nous pouvons classer les variables en deux catégories : les variables de premier niveau (ou de niveau supérieur), ici x , et les variables de second niveau (ou de niveau inférieur), ici y . De même, on distingue l’objectif de premier niveau, ici F , et l’objectif de second niveau, ici f . Les contraintes sont également différenciées : G et g constituent respectivement les contraintes de premier et second niveaux.

Colson *et al.* (2005) rappellent plusieurs notions concernant les problèmes biniveau. Nous allons reprendre ces notions, qui sont essentielles à la compréhension de l’architecture d’un programme biniveau.

L’ensemble réalisable du problème se note Ω et s’écrit :

$$\Omega = \{(x, y) : G(x, y) \leq 0, g(x, y) \leq 0\}$$

Pour un vecteur donné $\bar{x} \in X$, l'ensemble réalisable du second niveau est défini par :

$$\Omega(\bar{x}) = \{y : g(\bar{x}, y) \leq 0\}$$

et l'ensemble de réaction du second niveau correspond à :

$$R(\bar{x}) = \{y : y \in \arg \min\{f(\bar{x}, y') : y' \in \Omega(\bar{x})\}\}$$

Pour x donné, $R(x)$ peut être vide. Finalement, on introduit une expression de l'ensemble réalisable du programme biniveau (également appelé *région induite*), que l'on note \mathcal{IR} :

$$\mathcal{IR} = \{(x, y) : x \in X, G(x, y) \leq 0, y \in R(x)\}$$

Cet ensemble est généralement non convexe, et peut être non connexe, voire vide en présence de contraintes de premier niveau impliquant des contraintes de second niveau.

Une solution optimale du programme biniveau est un point (x^*, y^*) tel que :

$$\begin{aligned} (x^*, y^*) &\in \mathcal{IR} \\ \forall (x, y) \in \mathcal{IR} \quad F(x^*, y^*) &\geq F(x, y) \end{aligned}$$

Finalement, on parlera du *problème relaxé* d'un problème biniveau lorsque l'objectif de second niveau n'est pas considéré. La valeur optimale du problème relaxé constitue une borne supérieure pour la valeur du programme biniveau. Nous pouvons le formuler de la sorte :

$$\begin{aligned} \max_{x \in X, y} \quad & F(x, y) \\ \text{s.à} \quad & G(x, y) \leq 0 \\ & g(x, y) \leq 0 \end{aligned}$$

Au final, le programme biniveau peut être perçu comme un problème d'optimisation, dans lequel on modélise hiérarchiquement les décisions prises par deux agents. L'agent meneur, au premier niveau, exécute ses décisions, tout en voulant optimiser son objectif (choix de x dans la formulation précédente), et en tenant compte de la réaction de l'agent suiveur au second niveau. Ce dernier prend également ses décisions (choix de y), en optimisant son objectif (second niveau), mais en utilisant certaines variables de décisions de l'agent meneur.

Colson *et al.* (2005) décrivent également les principales propriétés des problèmes biniveau, et proposent des méthodes de résolution. Pour des problèmes linéaires dont l'ensemble des contraintes est un polyèdre, la solution optimale se trouve sur un des points extrêmes du polyèdre, et il existe des méthodes d'énumération de ces points extrêmes. Il est également possible de remplacer le second niveau par les conditions de KKT, lorsque le problème de second niveau est convexe. Le problème à un seul niveau ainsi obtenu se résout généralement grâce à une méthode de séparation et d'évaluation (*branch-and-bound*). Il existe également d'autres méthodes, telle la méthode de pénalités, ou la méthode de région de confiance.

Vicente *et al.* (1996) étudient la géométrie de l'ensemble réalisable et l'existence de solutions optimales pour les problèmes biniveau linéaires impliquant des variables discrètes. Dans un premier temps, ils considèrent le cas de figure où seules les variables de premier niveau sont discrètes, puis celui où les variables de premier et deuxième niveau sont discrètes, et enfin la situation où seules les variables de second niveau le sont. Brotcorne *et al.* (2009) étudient un type particulier de programme biniveau à une seule variable, dans lequel le meneur détermine la capacité d'un problème de sac-à-dos de façon à maximiser son profit, tandis que le suiveur résout le problème de sac-à-dos résultant. Les auteurs proposent un algorithme de programmation dynamique pour résoudre le problème.

Dempe (2003) présente une bibliographie complète de 460 références sur la programmation biniveau, et le lecteur pourra s'y référer pour plus d'informations sur le

sujet.

Hansen *et al.* (1992) ont montré que la programmation biniveau est NP difficile. De plus en plus de méthodes heuristiques sont alors développées pour résoudre ce type de problèmes. Gendreau *et al.* (1996) ont travaillé sur une méthode hybride combinant une recherche tabou et une méthode de descente pour approcher les solutions optimales d'un problème biniveau linéaire.

Calvete *et al.* (2008) ont développé des algorithmes génétiques pour résoudre des problèmes biniveau dont l'objectif et les contraintes sont linéaires. Leur algorithme combine des techniques classiques d'énumération de points extrêmes avec une méthode génétique de recherche qui associe les chromosomes aux points extrêmes du polyèdre. Dussault *et al.* (2006) ont quant à eux mis au point une méthode heuristique basée sur des points intérieurs et combinant plusieurs techniques de programmation implicite.

La programmation biniveau s'avère intéressante en tarification, où le meneur maximise son profit et le suiveur minimise ses coûts perçus. Marcotte et Savard (2002) ont utilisé ce concept pour modéliser plusieurs situations de ce type. Dans le problème de la tarification des autoroutes, le meneur veut maximiser son profit en taxant les segments d'autoroutes, et le suiveur veut minimiser ses coûts lors d'un déplacement. Le même type de schéma s'applique aux réseaux de télécommunication.

Chapitre 3

MODÉLISATION DU PROBLÈME DU PLACEMENT MÉDIA

Dans ce chapitre, nous exposons dans une première section le modèle proposé par Chan (2008). Puis nous développons dans la section 3.2 les hypothèses relatives à notre modèle, qui s'avère être une variante du modèle de Chan (2008) : en effet le M-Patron alloue des ensembles de spots alors que le M-Spot alloue des spots. Ensuite nous enchaînons dans la section 3.3 avec la formulation mathématique. Nous terminons enfin par la résolution numérique d'un exemple.

3.1 Le modèle par spots

Nous reprenons dans cette section le modèle de Chan (2008).

3.1.1 Généralités sur le modèle

Dans ce problème biniveau, le meneur (chaîne de télévision) fixe ses tarifs et obtient une allocation optimale qui maximise son revenu, et le suiveur (agence de publicité) utilise le tarif fixé par le meneur pour déterminer lui aussi l'allocation optimale qui minimise ses coûts. Notons que la programmation exacte des spots n'est pas déterminée lors de l'allocation des spots publicitaires. Notre modèle détermine quel spot ira à quelle pause publicitaire, mais l'ordre n'est pas fixé. Cet ordre est

laissé à l'appréciation de la chaîne leader si elle veut éviter d'éventuels conflits entre les annonceurs publicitaires. Bollapragada et Garbiras (2004) s'y sont intéressés si le lecteur désire approfondir le sujet.

La chaîne de télévision est leader sur le marché : elle estime alors les prix des spots vendus par les autres chaînes concurrentes et fixe sa politique de tarification en conséquence. De plus, si le client ne trouve pas de spot du meneur qui lui convienne, celui-ci peut choisir des spots proposés par la concurrence. Ne connaissant pas l'inventaire de la concurrence, nous le supposons infini, et pouvant ainsi toujours répondre à la demande des clients. On introduit également le concept de *désutilité*, qui est associée à chaque spot publicitaire. Cette désutilité, quantifiée financièrement, pénalise les spots publicitaires selon l'attrait qu'ils dégagent aux yeux des clients. Par exemple, un spot publicitaire qui intéresse beaucoup un client aura une désutilité faible, contrairement à un spot publicitaire qui intéresse moyennement le client.

Nous développons ici un modèle d'aide à la décision. Ce modèle fixe des prix des spots vendus par la chaîne leader, et ces prix guident les clients vers la combinaison optimale renvoyée par le programme.

Ces généralités, qui ont été énoncées en premier lieu pour le modèle M-Spot, s'appliquent également à notre modèle. Nous les utiliserons pour écrire notre programme biniveau.

3.1.2 Les notations du modèle M-Spot

Nous rappelons ici brièvement les notations concernant le modèle M-Spot. Dans ces notations, i désigne un agent du modèle ($i = 1$ pour la chaîne leader, $i = 2$ pour la concurrence). Une catégorie démographique concerne un ensemble particulier de téléspectateur, définie selon l'âge des spectateurs. Par exemple, la catégorie [18-25] concernent les personnes dont l'âge est compris entre 18 et 25 ans. Étant donné que les annonceurs produisent des spots publicitaires pour atteindre un certain type de

personnes (un vendeur de jouets voudra cibler la catégorie [7-12]), nous prenons ce choix en compte dans le modèle.

Les indices du modèle sont les suivants :

s : indice des pauses publicitaires ;

d : indice des catégories démographiques ;

c : indice des clients ;

l : indice des durées possibles des spots publicitaires.

Nous présentons ensuite les différents ensembles du modèle :

U^1 : ensemble des pauses de la chaîne leader sur l'*Up-market* ;

U^2 : ensemble des pauses des chaînes concurrentes sur l'*Up-market* ;

D : ensemble des catégories démographiques ;

C : ensemble des clients ;

L : ensemble des durées de spot publicitaire ;

$D(s)$: ensemble des catégories démographiques qui constituent l'audience de la pause s ;

$d(c)$: catégorie démographique qui intéresse le client c .

Les caractéristiques des pauses publicitaires sont les suivantes :

$t_{s,l}^1$: prix d'un spot de l secondes lors de la pause publicitaire s vendu par le leader ;

$t_{s,l}^2$: prix d'un spot de l secondes lors de la pause publicitaire s vendu par la concurrence ;

t^i : vecteur de prix pour l'agent i ;

$a_{s,l,d}$: niveau d'audimat dans la catégorie démographique d de la pause publicitaire s pour l secondes ;

$l(s)$: durée de la pause publicitaire s ;

$I_{s,l}^1$: nombre maximum de spots publicitaires de durée l qui peuvent être vendus lors de la pause s proposée par la chaîne leader ;

$w_{s,l,c}^i$: désutilité financière du client c pour un spot publicitaire de durée l lors de la pause publicitaire s vendu par i .

Les variables de décision au niveau inférieur sont :

$x_{s,l,c}^i$: élément de $\{0, 1\}$ qui indique si oui ou non un spot publicitaire de durée l est vendu au client c lors de la pause s par l'agent i .

Le dernier paramètre du modèle est :

\underline{A}_c : borne inférieure d'audimat pour le client c .

3.1.3 Le programme mathématique M-Spot

Voici le modèle M-Spot développé par Chan (2008) pour modéliser le problème du placement média :

$$\max_{t^1, x^1, x^2} \sum_{c \in C} \sum_{l \in L} \sum_{s \in U^1} t_{s,l}^1 x_{s,l,c}^1 \quad (3.1)$$

$$\text{s.à} \quad \sum_{c \in C} x_{s,l,c}^1 \leq I_{s,l}^1 \quad \forall s, l \in U^1 \times L \quad (3.2)$$

$$\sum_{l \in L} \sum_{c \in C} l x_{s,l,c}^1 \leq l(s) \quad \forall s \in U^1 \quad (3.3)$$

$$\min_{x^1, x^2} \sum_{c \in C} \sum_{l \in L} \left(\sum_{s \in U^1} (t_{s,l}^1 + w_{s,l,c}^1) x_{s,l,c}^1 + \sum_{s \in U^2} (t_{s,l}^2 + w_{s,l,c}^2) x_{s,l,c}^2 \right) \quad (3.4)$$

$$\text{s.à} \quad \sum_{\substack{s,l \in U^1, L \\ d(c) \in D(s)}} a_{s,l,d} x_{s,l,c}^1 + \sum_{\substack{s,l \in U^2, L \\ d(c) \in D(s)}} a_{s,l,d} x_{s,l,c}^2 \geq \underline{A}_c \quad \forall c \in C \quad (3.5)$$

$$x_{s,l,c}^i \in \{0, 1\} \quad \forall i, s, l \in \{1, 2\} \times U^i \times L \quad \forall c \in C \quad (3.6)$$

$$t_{s,l}^1 \geq 0 \quad \forall s, l \in U^1 \times L \quad (3.7)$$

Ce modèle est un modèle par spot, dans le sens où la chaîne leader vend aux clients des *spots publicitaires*. La variable de décision $x_{s,l,c}$ sera égale à 1 si le modèle alloue au client c un *spot* d'un prix $t_{s,l}$ de longueur l lors de la pause s . Également, $w_{s,l,c}$ est la désutilité d'un *spot* de longueur l lors de la pause s pour un client c .

Au premier niveau, la chaîne leader veut maximiser son revenu par la vente des spots publicitaires (3.1). Elle dispose de deux contraintes d'inventaires relatives aux durées des pauses publicitaires (3.3) et à la répartition des spots sur ces pauses (3.2). Au second niveau, les clients minimisent leurs coûts (3.4), tout en atteignant leur objectif d'audience (3.5). Pour plus de détails concernant les fonctions du modèle M-Spot, le lecteur consultera le mémoire de Chan (2008).

Dans la prochaine section, nous énonçons les hypothèses relatives à notre modèle,

et relevons les différences avec le modèle M-Spot.

3.2 Hypothèses du modèle M-Patron

En premier lieu, définissons clairement ce que nous entendons par *patron*. En recherche opérationnelle, les patrons sont très utilisés pour résoudre le problème de la découpe des rouleaux (*Cutting Stock Problem*). Dans ce problème, l'idée est de générer des patrons de découpe de façon à avoir le moins de pertes possibles. Pour en savoir plus sur la génération des patrons pour ce problème, le lecteur pourra se référer aux travaux de Suliman (2001). De même que les patrons sont composés de découpes de rouleaux de différentes longueurs dans le *Cutting Stock Problem*, l'idée ici est de découper l'espace publicitaire disponible, en proposant des patrons composés de spots de durées différentes.

Nous pouvons par exemple concevoir un patron comprenant un spot d'une longueur 30 secondes, sur quatre pauses consécutives. La chaîne leader vendra alors au client un ensemble de quatre spots de 30 secondes répartis sur quatre pauses consécutives. La chaîne peut également proposer des combinaisons de patrons comprenant des spots de 30 secondes en début de soirée et de patrons comprenant des spots de 15 secondes à la fin de soirée. Le patron peut être ainsi vu comme un contrat portant sur la vente de plusieurs spots publicitaires.

De plus, nous considérons que les patrons contiennent des spots sur une journée uniquement, car le modèle développé ici est un modèle séparable par jour. Nous définissons également l'architecture des patrons, et les spots n'ont pas d'ordre fixé dans les patrons. Par exemple, si la chaîne de télévision propose un patron de deux spots publicitaires sur deux pauses publicitaires, rien ne précise si les spots seront programmés au début, au milieu ou à la fin des pauses. Enfin, un client peut éventuellement acheter plusieurs patrons.

Également, l'utilisation des patrons permet à la chaîne de télévision d'espacer les spots publicitaires de manière régulière. Effectivement, le modèle M-Spot alloue les spots aux clients de manière à maximiser le revenu de la chaîne de télévision. Si un client désire quatre spots publicitaires pour atteindre son objectif d'audience, aucune contrainte ne permet d'espacer les spots alloués. Ainsi, il peut arriver que les quatre spots soient placés lors de la même pause. En définissant des patrons contenant des spots sur chaque pause publicitaire, cela empêche le précédent scénario de se produire. L'introduction des patrons permet alors d'obtenir des solutions plus réalistes.

Nous sommes prêts désormais à formuler notre problème avec des patrons, et nous allons ainsi détailler dans une deuxième partie le modèle mathématique associé au problème de gestion du revenu dans l'industrie des médias.

3.3 Modèle mathématique

3.3.1 Notations relatives au modèle

Nous commençons par décrire les différents indices et ensembles du modèle.

Les indices du modèle :

s : indice des pauses publicitaires ;

d : indice des catégories démographiques ;

c : indice des clients ;

l : indice des durées possibles des spots publicitaires ;

p : indice des patrons.

Les ensembles du modèle :

P^i : ensemble des patrons vendus par l'agent i sur une journée ;

C : ensemble des clients ;

- L : ensemble des durées de spot publicitaire ;
- $d(c)$: catégorie démographique qui intéresse le client c ;
- $D(s)$: ensemble des catégories démographiques qui constituent l'audience de la pause s ;
- U^1 : ensemble des pauses de la chaîne leader réparties sur une journée ;
- U^2 : ensemble des pauses de la concurrence réparties sur une journée ;
- D : ensemble des catégories démographiques.

L'indice i se réfère soit à la chaîne leader ($i = 1$), soit aux autres chaînes concurrentes ($i = 2$), que l'on considère comme une seule entité.

Définissons désormais les paramètres du modèle.

Les paramètres du modèle :

- t_p^2 : prix du patron p vendu par la concurrence ;
- α_l : durée d'un spot publicitaire de longueur l ;
- $l(s)$: durée de la pause s ;
- $\delta_{s,l,p}^i$: coefficient binaire de $\{0, 1\}$ qui indique si le patron p proposé par i contient un spot de longueur l lors la pause s ;
- $I_{s,l}^1$: nombre maximum de spots publicitaires de durée l qui peuvent être vendus lors de la pause s proposée par la chaîne leader ;
- $a_{s,l,d}^i$: niveau d'audimat dans la catégorie démographique d de la pause publicitaire s proposée par i , pour l secondes ;
- \underline{A}_c : borne inférieure d'audimat pour le client c ;
- $w_{c,p}^i$: désutilité pour un patron p vendu au client c par l'agent i ;

La longueur de spot l est un indice, ainsi a-t-on $l = 1, 2, \dots$. On introduit alors la durée d'un spot publicitaire α_l , qui sera égale à 15 secondes si $l = 1$ par exemple. La

désutilité permet de quantifier financièrement l'aversion qu'a un client pour certains patrons : ce facteur permet ainsi de pénaliser les patrons. Prenons deux patrons p_1 et p_2 . Le patron p_1 contient 5 spots en début de soirée. Le patron p_2 contient 5 spots répartis sur toute la soirée. Un client β préfère faire de la publicité concentrée, de façon à marquer les esprits. L'aversion pour le patron p_2 sera alors plus importante que celle pour le patron p_1 , car ce dernier couvre toute la soirée, ce qui ne rentre par forcément dans les plans du client. On pourra alors par exemple quantifier financièrement les désutilités de p_1 et de p_2 comme étant égales respectivement à un et dix dollars.

Nous introduisons enfin les variables du modèle :

Les variables :

$y_{c,p}^i$: variable binaire de $\{0, 1\}$ qui indique si un patron p est choisi par le client c ;

t_p^1 : prix du patron p vendu par la chaîne leader.

3.3.2 Les fonctions et les contraintes du modèle

Nous pouvons désormais écrire les fonctions du modèle régissant les objectifs et contraintes de premier et deuxième niveaux.

Les objectifs et contraintes :

La chaîne leader maximise l'ensemble de ses revenus, soit la somme des prix de ses patrons achetés par les clients. Cela se traduit par la fonction suivante :

$$\max_{t^1, y^1, y^2} \sum_{c \in C} \sum_{p \in P^1} t_p^1 \cdot y_{c,p}^1$$

La chaîne leader est contrainte par son inventaire. La première contrainte d'inventaire caractérise la longueur des pauses publicitaires ; la somme des longueurs α_l de spots vendus lors d'une pause s ne peut dépasser la longueur $l(s)$ de cette même pause :

$$\sum_{c \in C} \sum_{p \in P^1} \sum_{l \in L} \alpha_l \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 \leq l(s) \quad \forall s \in U^1$$

De plus, on modélise le fait que lors de certaines pauses s , on ne veut pas qu'il y ait plus de $I_{s,l}^1$ spots de même longueur l :

$$\sum_{p \in P^1} \sum_{c \in C} \delta_{s,l,p}^1 \cdot y_{c,p}^1 \leq I_{s,l}^1 \quad \forall s, l \in U^1 \times L$$

Au second niveau, les clients veulent minimiser les coûts occasionnés par l'achat des patrons soit à la chaîne leader, ou soit à la concurrence. Nous incluons les désutilités, ce qui donne comme fonction objectif :

$$\min_{y^1, y^2} \sum_{c \in C} \left(\sum_{p \in P^1} (t_p^1 + w_{c,p}^1) \cdot y_{c,p}^1 + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \right)$$

À cela vient s'ajouter la contrainte d'audimat pour chaque client c , qui souhaite rejoindre minimalement un nombre de personnes A_c :

$$\sum_{l \in L} \left(\sum_{\substack{s \in U^1 \\ d(c) \in D(s) \\ p \in P^1}} a_{s,l,d}^1 \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 + \sum_{\substack{s \in U^2 \\ d(c) \in D(s) \\ p \in P^2}} a_{s,l,d}^2 \cdot \delta_{s,l,p}^2 \cdot y_{c,p}^2 \right) \geq A_c \quad \forall c \in C$$

Les variables $y_{c,p}^i$ sont binaires dans l'ensemble du modèle. Le prix t_p^1 des patrons vendus par la chaîne leader est positif.

$$\begin{aligned} y_{c,p}^1 &\in \{0, 1\} & \forall c, p \in C \times P^1 \\ y_{c,p}^2 &\in \{0, 1\} & \forall c, p \in C \times P^2 \\ t_p^1 &\geq 0 & \forall p \in P^1 \end{aligned}$$

3.3.3 Le modèle mathématique

Nous obtenons alors le problème biniveau suivant :

$$\begin{aligned}
& \max_{t^1, y^1, y^2} \sum_{c \in C} \sum_{p \in P^1} t_p^1 \cdot y_{c,p}^1 \\
& \text{s.à} \sum_{p \in P^1} \sum_{c \in C} \delta_{s,l,p}^1 \cdot y_{c,p}^1 \leq I_{s,l}^1 \quad \forall s, l \in U^1 \times L \\
& \sum_{c \in C} \sum_{p \in P^1} \sum_{l \in L} \alpha_l \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 \leq l(s) \quad \forall s \in U^1 \\
\\
& \min_{y^1, y^2} \sum_{c \in C} \left(\sum_{p \in P^1} (t_p^1 + w_{c,p}^1) \cdot y_{c,p}^1 + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \right) \\
& \text{s.à} \sum_{l \in L} \left(\sum_{\substack{s \in U^1 \\ d(c) \in D(s) \\ p \in P^1}} a_{s,l,d}^1 \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 + \sum_{\substack{s \in U^2 \\ d(c) \in D(s) \\ p \in P^2}} a_{s,l,d}^2 \cdot \delta_{s,l,p}^2 \cdot y_{c,p}^2 \right) \geq \underline{A}_c \quad \forall c \in C \\
& y_{c,p}^i \in \{0, 1\} \quad \forall c, p \in C \times P^i \quad \forall i \in \{1, 2\} \\
& t_p^1 \geq 0 \quad \forall p \in P^1
\end{aligned}$$

Ce modèle constitue une généralisation du M-Spot.

3.4 Un exemple

Nous allons maintenant illustrer le modèle par un exemple. Avant cela, nous rappelons la notion d'*Ensemble Minimal*, qui sera essentielle à la compréhension de l'exemple. Introduite par Chan (2008), cette notion permet de limiter le nombre de patrons considérés.

3.4.1 Ensemble Minimal

L'ensemble minimal pour un client c qualifie un ensemble de spots qui satisfait l'audimat d'un client c . Dès que l'on retire un spot de cet *ensemble*, la contrainte d'audimat relative au client en question est violée, ce qui explique le qualificatif *minimal*.

Cette notion est pertinente pour notre modèle : en effet la contrainte d'audience est exactement la même que celle du M-Spot, à la différence que l'on vend au client un ensemble de spots. L'ensemble minimal pourra alors contenir des patrons à la place des spots.

Voici la définition mathématique de l'ensemble minimal :

Définition 1 Soit $c \in C$. Soit E un sous-ensemble de $P^1 \cup P^2$. E est un ensemble minimal pour c si et seulement si :

$$\begin{aligned} \forall F \subset E \quad \sum_{p \in F \cap P^1} \sum_{(s,l) \in (U^1 \times L)} \delta_{s,l,p}^1 \cdot a_{s,l,d}^1 + \sum_{p \in F \cap P^2} \sum_{(s,l) \in (U^2 \times L)} \delta_{s,l,p}^2 \cdot a_{s,l,d}^2 &< \underline{A}_c \\ \sum_{p \in E \cap P^1} \sum_{(s,l) \in (U^1 \times L)} \delta_{s,l,p}^1 \cdot a_{s,l,d}^1 + \sum_{p \in E \cap P^2} \sum_{(s,l) \in (U^2 \times L)} \delta_{s,l,p}^2 \cdot a_{s,l,d}^2 &\geq \underline{A}_c \end{aligned}$$

Définition 2 Soit $c \in C$. On note $Em(c)$ l'ensemble des ensembles minimaux pour c .

Nous pouvons alors retenir les mêmes conclusions que celles du modèle M-Spot :

Propriété 1 Soit (y_{opt}^1, y_{opt}^2) une solution optimale du problème de second niveau. Soit $c \in C$. Alors $\{p \in (P^1 \cup P^2) \mid y_{c,p}^i = 1\}$ est un élément de $Em(c)$.

Définition 3 Soient $t \in (\mathbb{R}^+)^n$, $c \in C$, $E \in Em(c)$. On dit que E est optimal pour c relativement à t si et seulement si il existe une solution optimale (y_{opt}^1, y_{opt}^2) du problème de second niveau, définie par t , telle que $E = \{p \in (P^1 \cup P^2) \mid y_{c,p}^i = 1\}$.

3.4.2 L'exemple

Considérons un problème avec :

C : 2 clients, voulant respectivement un audimat de 80 et 60 ;

D : 1 une catégorie démographique ;

L : 2 longueurs de spot publicitaire, $\alpha_1 = 10\text{sec}$ et $\alpha_2 = 20\text{sec}$;

U^1 : 2 pauses publicitaires pour le leader, respectivement d'une longueur de 30 et 20 secondes ;

U^2 : 2 pauses publicitaires pour la concurrence ;

P^1 : 2 patrons proposés par le leader ;

P^2 : 2 patrons proposés par la concurrence ;

$I_{s,l}^1$: $I_{1,1}^1 = 1$, $I_{1,2}^1 = 1$ et $I_{2,1}^1 = 2$.

La chaîne leader propose donc deux patrons : le patron 1 comprend deux spots de 10 secondes lors des pauses 1 et 2, et le patron 2 comprend un spot de 20 secondes lors de la pause 1 et un spot de 10 secondes lors de la pause 2. La figure 3.1 représente de manière schématique l'exemple énoncé. Le tableau 3.1 donne d'autres paramètres associés aux patrons, à savoir l'audience des spots publicitaires et la désutilité.

La concurrence propose quant à elle un patron contenant deux spots de 10 secondes lors des pauses 1 et 2, et un patron contenant deux spots de 20 secondes lors des pauses 1 et 2. Ces pauses ne sont pas les mêmes que celles proposées par la chaîne leader. Les autres informations relatives à ces patrons (audience des spots, désutilités, prix) sont données dans les tableaux 3.2.

TABLEAU 3.1 Audiences et désutilités pour le **leader**

	pause 1	pause 2
longueur 1	$a_{1,1,1}^1 = 30$	$a_{1,2,1}^1 = 20$
longueur 2	$a_{2,1,1}^1 = 40$	

	client 1	client 2
patron 1	$w_{1,1}^1 = 5$	$w_{1,2}^1 = 10$
patron 2	$w_{2,1}^1 = 0$	$w_{2,2}^1 = 5$

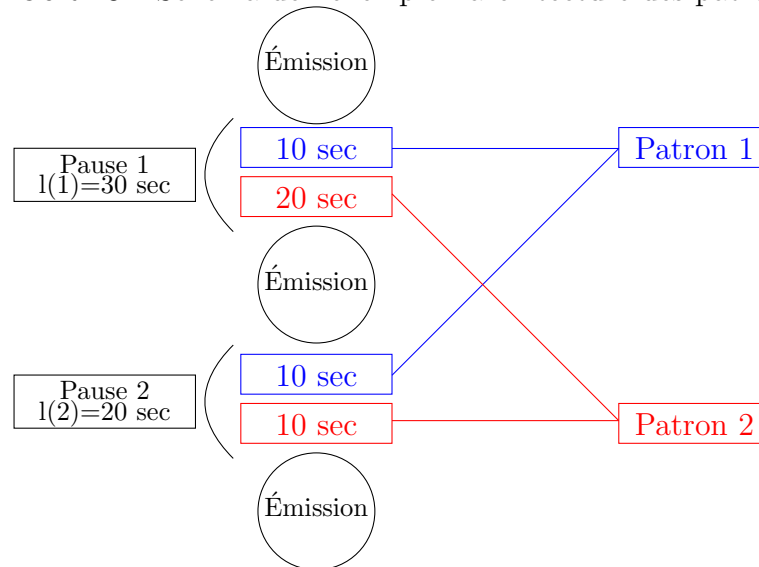
TABLEAU 3.2 Audiences, désutilités et prix des patrons de la **concurrence**

	pause 1	pause 2
longueur 1	$a_{1,1,1}^2 = 30$	$a_{1,2,1}^2 = 10$
longueur 2	$a_{2,1,1}^2 = 25$	$a_{2,2,1}^2 = 20$

	client 1	client 2
patron 1	$w_{1,1}^2 = 15$	$w_{1,2}^2 = 5$
patron 2	$w_{2,1}^2 = 10$	$w_{2,2}^2 = 5$

	patron 2	patron 1
prix	$t_1^2 = 20$	$t_2^2 = 25$

FIGURE 3.1 Schéma de l'exemple : architecture des patrons



Le modèle se réécrit ainsi :

$$\begin{aligned}
& \max_{t^1, y^1, y^2} && t_1^1 \cdot (y_{1,1}^1 + y_{2,1}^1) + t_2^1 \cdot (y_{1,2}^1 + y_{2,2}^1) \\
& \text{s.à} && 10 \cdot (y_{1,1}^1 + y_{2,1}^1) + 20 \cdot (y_{1,2}^1 + y_{2,2}^1) \leq 30 \\
& && 10 \cdot (y_{1,1}^1 + y_{2,1}^1) + 10 \cdot (y_{1,2}^1 + y_{2,2}^1) \leq 20 \\
& && y_{1,1}^1 + y_{2,1}^1 \leq 1 \\
& && y_{1,2}^1 + y_{2,2}^1 \leq 1 \\
& && y_{1,1}^1 + y_{2,1}^1 + y_{1,2}^1 + y_{2,2}^1 \leq 2 \\
& \min_{y^1, y^2} && (t_1^1 + 5) \cdot y_{1,1}^1 + (t_1^1 + 10) \cdot y_{1,2}^1 + t_2^1 \cdot y_{1,2}^1 + (t_2^1 + 5) \cdot y_{2,2}^1 + \\
& && 35y_{1,1}^2 + 25y_{2,1}^2 + 35y_{1,2}^2 + 30y_{2,2}^2 \\
& \text{s.à} && 50y_{1,1}^1 + 60y_{1,2}^1 + 40y_{1,1}^2 + 45y_{1,2}^2 \geq 80 \\
& && 50y_{2,1}^1 + 60y_{2,2}^1 + 40y_{2,1}^2 + 45y_{2,2}^2 \geq 60 \\
& && y_{1,1}^1, y_{1,2}^1, y_{2,1}^1, y_{2,2}^1, y_{1,1}^2, y_{1,2}^2, y_{2,1}^2, y_{2,2}^2 \in \{0, 1\} \\
& && t_1^1, t_2^1 \geq 0
\end{aligned}$$

Nous déterminons dans un premier temps les ensembles minimaux pour le client 1, $Em(1)$, et ceux pour le client 2, $Em(2)$ (nous les calculons à la main). Nous employons la notation (i, c, p) lorsqu'un patron p est attribué au client c par i . Nous obtenons

alors :

$$\text{Em}(1) = \{(1,1,1),(1,1,2)\};\{(1,1,1),(2,1,1)\};\{(1,1,1),(2,1,2)\} \\ \{(1,1,2),(2,1,1)\};\{(1,1,2),(2,1,2)\};\{(2,1,1),(2,1,2)\}$$

$$\text{Em}(2) = \{(1,2,1),(2,2,1)\};\{(1,2,1);(2,2,2)\};\{(1,2,2)\};\{(2,2,1),(2,2,2)\}$$

À chaque ensemble minimal on associe un prix, fonction de (t_1^1, t_2^1) . Le tableau 3.3 résume les prix affectés aux ensembles minimaux.

Pour chaque client, l'ensemble optimal de patrons attribués est un ensemble minimal. Un ensemble minimal est optimal pour chaque client c si son coût est le plus petit possible, soit le plus petit parmi les autres ensembles minimaux. Par exemple, considérons le client 1. Pour l'ensemble minimal suivant $\{(1,1,1);(2,1,1)\}$, les équations suivantes doivent être respectées :

$$t_1^1 + 40 \leq t_2^1 + 35 \quad \text{i.e.} \quad t_2^1 - t_1^1 \geq 5$$

$$t_1^1 + 40 \leq 70 \quad \text{i.e.} \quad t_1^1 \leq 30$$

L'ensemble de ces contraintes forme alors un système linéaire sur t_1^1 et sur t_2^1 , qui assurera l'optimalité pour l'ensemble minimal du client 1. Ce raisonnement doit être appliqué à tous les ensembles minimaux des clients 1 et 2. Nous obtenons ainsi

TABLEAU 3.3 Les ensembles minimaux et leurs prix

Client 1		Client 2	
$\{(1,1,1);(2,1,1)\}$	$t_1^1 + 40$	$\{(1,2,1);(2,2,1)\}$	$t_1^1 + 35$
$\{(1,1,1);(2,1,2)\}$	$t_1^1 + 40$	$\{(1,2,1);(2,2,2)\}$	$t_1^1 + 40$
$\{(1,1,2);(2,1,1)\}$	$t_2^1 + 35$	$\{(1,2,2)\}$	$t_2^1 + 40$
$\{(1,1,2);(1,2,2)\}$	$t_2^1 + 35$	$\{(2,2,1);(2,2,2)\}$	55
$\{(2,1,1);(2,1,2)\}$	70		

un ensemble de polyèdres illustrés à la figure 3.2. Les traits noirs représentent les contraintes relatives au client 1, et les traits rouges celles relatives au client 2.

Chaque polyèdre ainsi défini forme un programme linéaire dont les contraintes de premier niveau doivent être vérifiées. On résout alors ce sous-problème sur chacun de ces polyèdres, et nous obtenons une solution optimale pour chaque polyèdre. La solution optimale du problème est la plus grande de toutes les solutions optimales de chaque sous-problème, et elle est égale à 80 dans cet exemple, pour $t_1^1 = 30$ et $t_2^1 = 50$.

La figure 3.2 ressemble fortement à celle de l'exemple traité par Chan (2008) (voir figure A.1 en Annexe A) et les polyèdres ont la même physionomie. Ayant repris la même structure à deux niveaux que ceux-ci, le modèle M-Patron semble alors fortement ressembler au modèle M-Spot : nous avons alors adopté le même cadre algorithmique.

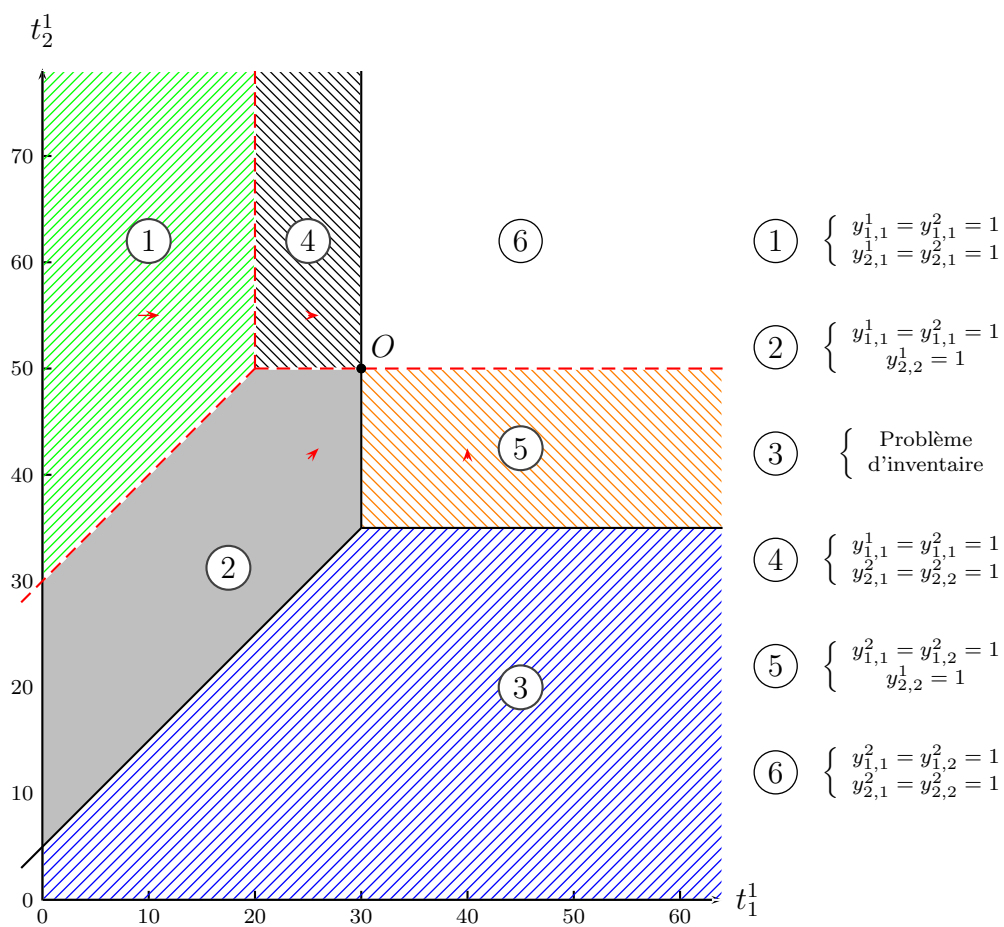


FIGURE 3.2 Ensemble des polyèdres d'optimalité pour le M-Patron

Chapitre 4

ALGORITHMES DE RÉSOLUTION

Ce chapitre est consacré à l'élaboration de plusieurs méthodes de résolution. Nous partons de l'algorithme développé par Chan (2008), et nous proposons deux nouveaux algorithmes.

4.1 Algorithme initial

Dans cette section, nous reprenons l'algorithme développé par Chan (2008) pour résoudre le modèle M-Spot. Nous adaptons cet algorithme au modèle M-Patron, et nous en décrivons les grandes lignes.

4.1.1 Cadre de l'algorithme

La première étape de l'algorithme consiste à relaxer le problème, en éliminant l'objectif du suiveur. La solution de ce problème nous retourne une allocation par client qui maximise le revenu de la chaîne leader. Dans une deuxième étape, nous utilisons les prix optimaux fixés par le problème relaxé pour résoudre le sous-problème du suiveur. Ce problème de sac-à-dos renvoie une allocation optimale par client qui minimise leurs coûts perçus. Ensuite nous comparons les coûts perçus renvoyés par les étapes une et deux et ce, pour chaque client. Si les coûts renvoyés par le problème de second niveau sont plus grands ou égaux que ceux renvoyés par le problème relaxé (pour tous les clients), alors on a trouvé des prix optimaux qui satisfont les deux

agents. Si ce n'est pas le cas, on ajoute des coupes progressivement à chaque itération.

Le vecteur de prix et le coût perçu pour un client c (renvoyés par le problème relaxé) sont définis respectivement par t^* et $R_{R,c}^*$. Le coût perçu pour un client c renvoyé par le problème de sac-à-dos est caractérisé par $R_{K,c}^*$. En résumé, voici la procédure de résolution :

Étape 1 : On résout le problème relaxé $(t^*, R_{R,c}^*)$.

Étape 2 : On résout le problème de second niveau avec t^* ($R_{K,c}^*$).

Étape 3 : On compare $R_{R,c}^*$ et $R_{K,c}^*$ pour chaque client c .

Des coupes sont ajoutées si $R_{K,c}^* < R_{R,c}^*$ et on retourne à l'étape 1.

Étape 4 : Lorsque $R_{R,c}^* \leq R_{K,c}^*$ pour tous les clients, Arrêt.

4.1.2 Le problème relaxé initial

Nous nous intéressons au problème relaxé dans cette sous-section. Nous enlevons la fonction de minimisation des coûts perçus des clients et ajoutons une étape d'initialisation.

Nous calculons en premier lieu R_c , qui est le coût perçu minimum pour un client c s'il ne choisissait que des patrons de la concurrence. Il suffit pour cela de résoudre un problème de sac-à-dos séparable par client, de la forme :

$$\begin{aligned}
& \min_{y^2} \sum_{c \in C} \sum_{p \in P^2} (t_{c,p}^2 + w_{c,p}^2) \cdot y_{c,p}^2 \\
& \text{s.à} \sum_{\substack{s,l \in U^2, L \\ p \in P^2}} \sum_{d(c) \in D(s)} a_{s,l,d}^2 \cdot \delta_{s,l,p}^2 \cdot y_{c,p}^2 \geq \underline{A}_c \quad \forall c \in C \\
& y_{c,p}^2 \in \{0, 1\} \quad \forall c, p \in C \times P^2
\end{aligned}$$

L'allocation optimale par client renvoyée par ce problème a donc pour valeur R_c . Nous ajoutons alors la contrainte suivante au problème relaxé :

$$\sum_{p \in P^1} (t_p^1 + w_{c,p}^1) y_{c,p}^1 + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) y_{c,p}^2 \leq R_c \quad \forall c \in C$$

Cette contrainte empêche que la meilleure combinaison formée des patrons du leader et de la concurrence soit plus élevée que la meilleure combinaison des patrons de la concurrence et ce, par client. En effet le client n'achètera jamais à la chaîne leader un assortiment de patrons ayant un prix plus élevé qu'un autre assortiment de patrons proposé par la concurrence.

Une deuxième difficulté subsiste pour résoudre ce problème relaxé. En effet le revenu du leader (objectif à maximiser) est bilinéaire. À l'aide de techniques utilisées par Marcotte et Savard (2002), nous pouvons linéariser l'objectif en posant :

$$T_{c,p}^1 = t_p^1 y_{c,p}^1 \quad \forall c, p \in C \times P^1$$

Ensuite nous posons les deux équations suivantes pour pouvoir récupérer le prix t_p^1 d'un patron p .

$$\begin{aligned}
- M_{c,p}(1 - y_{c,p}^1) &\leq T_{c,p}^1 - t_p^1 \leq M_{c,p}(1 - y_{c,p}^1) && \forall c, p \in C \times P^1 \\
- M_{c,p} \cdot y_{c,p}^1 &\leq T_{c,p}^1 \leq M_{c,p} \cdot y_{c,p}^1 && \forall c, p \in C \times P^1
\end{aligned}$$

Propriété 2 $M_{c,p} = \max_{c \in C} \{R_c\} - w_{c,p}$, et constitue une borne supérieure sur $T_{c,p}^1$.

Nous savons que le coût perçu pour un client c ne peut excéder $\max_{c \in C} \{R_c\}$: nous prenons le max ici car le prix des patrons est le même pour l'ensemble des clients. Nous obtenons alors l'équation suivante :

$$\sum_{p \in P^1} (T_{c,p}^1 + w_{c,p}^1 \cdot y_{c,p}^1) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) y_{c,p}^2 \leq \max_{c \in C} \{R_c\} \quad \forall c \in C$$

d'où,

$$T_{c,p}^1 \leq \sum_{p \in P^1} T_{c,p}^1 \leq \max_{c \in C} \{R_c\} - \sum_{p \in P^1} w_{c,p}^1 y_{c,p}^1 - \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) y_{c,p}^2 \quad \forall c, p \in C \times P^1$$

et donc on a :

$$T_{c,p}^1 \leq \max_{c \in C} \{R_c\} - w_{c,p}^1 = M_{c,p}$$

Nous considérons alors le problème relaxé initial suivant :

Problème relaxé initial

$$\begin{aligned}
& \max_{t^1, y^1, y^2, T^1} \sum_{c \in C} \sum_{p \in P^1} T_{c,p}^1 \\
& \text{s.à} \quad \sum_{c \in C} \sum_{p \in P^1} \sum_{l \in L} \alpha_l \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 \leq l(s) \quad \forall s \in U^1 \\
& \quad \sum_{c \in C} \sum_{p \in P^1} \delta_{s,l,p}^1 \cdot y_{c,p}^1 \leq I_{s,l}^1 \quad \forall s, l \in U^1 \times L \\
& \quad \sum_{p \in P^1} (T_{c,p}^1 + w_{c,p}^1 \cdot y_{c,p}^1) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \leq R_c \quad \forall c \in C \\
& \quad \sum_{\substack{s,l \in U^1, L \\ p \in P^1 \\ d(c) \in D(s)}} a_{s,l,d}^1 \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 + \sum_{\substack{s,l \in U^2, L \\ p \in P^2 \\ d(c) \in D(s)}} a_{s,l,d}^2 \cdot \delta_{s,l,p}^2 \cdot y_{c,p}^2 \geq \underline{A}_c \quad \forall c \in C \\
& \quad - M_{c,p} \cdot (1 - y_{c,p}^1) \leq T_{c,p}^1 - t_p^1 \leq M_{c,p} \cdot (1 - y_{c,p}^1) \quad \forall c, p \in C \times P^1 \\
& \quad - M_{c,p} \cdot y_{c,p}^1 \leq T_{c,p}^1 \leq M_{c,p} \cdot y_{c,p}^1 \quad \forall c, p \in C \times P^1 \\
& \quad y_{c,p}^i \in \{0, 1\} \quad \forall i, c, p \in \{1, 2\} \times C \times P^1 \\
& \quad T_{c,p}^1 \geq 0 \quad \forall c, p \in C \times P^1 \\
& \quad t_p^1 \geq 0 \quad \forall p \in P^1
\end{aligned}$$

Ce problème relaxé est un programme linéaire mixte en nombre entier (MIP). Une fois résolu, il nous retourne les prix t_p^* des patrons p , ainsi qu'une allocation optimale $(y^{1,*}, y^{2,*})$.

4.1.3 Le problème relaxé modifié et la génération des coupes

Nous avons décrit précédemment l'étape 1. Nous allons ici décrire les étapes 2 et 3 car l'étape 2 est très liée à l'étape 3. À l'étape 2 (itération i) nous utilisons les prix t_p^* des patrons p renvoyés par le problème relaxé pour résoudre le problème de sac-à-dos qui minimise les coûts perçus des clients. Ce problème est le suivant :

$$\begin{aligned}
& \min_{y^1, y^2} \sum_{c \in C} \left(\sum_{p \in P^1} (t_p^* + w_{c,p}^1) \cdot y_{c,p}^1 + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \right) \\
& \text{s.à} \sum_{\substack{s, l \in U^1, L \\ p \in P^1 \\ d(c) \in D(s)}} a_{s,l,d}^1 \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 + \sum_{\substack{s, l \in U^2, L \\ p \in P^2 \\ d(c) \in D(s)}} a_{s,l,d}^2 \cdot \delta_{s,l,p}^2 \cdot y_{c,p}^2 \geq \underline{A}_c \quad \forall c \in C \\
& y_{c,p}^i \in \{0, 1\} \quad \forall c, p \in C \times P^i \quad \forall i \in \{1, 2\}
\end{aligned}$$

Ce problème de sac-à-dos est très facile à résoudre, et donne une solution optimale $y^{1,inf}(i), y^{2,inf}(i)$. Pour générer les coupes que l'on va ajouter au problème relaxé initial, nous comparons les coûts retournés par le problème relaxé et le problème de second niveau, pour chaque client. Nous vérifions alors que, pour tout client c , $(t^* + w_c^1) \cdot y^{1,*} + (t^{2,*} + w_c^2) \cdot y^{2,*}$ est inférieur ou égal à la valeur optimale de l'objectif du problème de second niveau, soit $(t^* + w_c^1) \cdot y^{1,inf}(i) + (t^{2,*} + w_c^2) \cdot y^{2,inf}(i)$.

Dans le cas contraire, nous ajoutons des coupes pour chaque client $c \in C(i)$, où $C(i)$ caractérise les clients dont la valeur de l'objectif retournée par le problème de second niveau est strictement plus petite que la solution optimale retournée par le problème relaxé, et ce, à l'itération i . On ajoute la coupe suivante relative à chaque client $c \in C(i)$:

$$\begin{aligned}
& \sum_{p \in P^1} (T_{c,p}^1 + w_{c,p}^1 \cdot y_{c,p}^1) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \leq \\
& \sum_{p \in P^1} (t_{c,p}^1 + w_{c,p}^1) \cdot y_{c,p}^{1,inf}(i) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^{2,inf}(i) \quad \forall c \in C(i)
\end{aligned}$$

Nous appellerons désormais “problème relaxé modifié” le problème relaxé initial auquel on ajoute des coupes à chaque itération.

Résumons les différentes étapes de l’algorithme :

Algorithme de génération de coupes - A0

Phase 0 : Initialisation

- On résout pour chaque client le problème de sac-à-dos nous donnant le meilleur prix proposé par la concurrence. On obtient alors R_c et on ajoute pour chaque client la contrainte de prix au problème relaxé initial.
- $i = 0$

Phase 1 : On résout le problème relaxé (MIP). On obtient t^* , $y^{1,*}$ et $y^{2,*}$.

Phase 2 :

- On résout le problème de second niveau avec t^* .
- On obtient $y_{c,p}^{1,inf}(i)$ et $y_{c,p}^{2,inf}(i)$.
- Pour chaque client c , on vérifie que le prix payé est le même dans le problème relaxé et le problème de second niveau.
- Si c’est le cas pour chaque client, alors $(t^*, y^{1,*}, y^{2,*})$ est une solution optimale. L’algorithme se termine.
- Sinon on passe à la phase 3.

Phase 3 : Génération des coupes

- Pour chaque client c ne vérifiant pas l’égalité énoncée à la phase 2, on introduit la coupe :

$$\sum_{p \in P^1} (T_{c,p}^1 + w_{c,p}^1 \cdot y_{c,p}^1) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \leq$$

$$\sum_{p \in P^1} (t_{c,p}^1 + w_{c,p}^1) \cdot y_{c,p}^{1,inf}(i) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^{2,inf}(i) \quad \forall c \in C(i)$$

- $i = i + 1$
- On retourne ensuite à la phase 1.

Il se peut que le couple $(y^{1,inf}(i), y^{2,inf}(i))$ ne vérifie pas les contraintes de premier niveau. Cependant on utilise uniquement ces valeurs pour la génération des coupes relatives à $(t^*, y^{1,*}, y^{2,*})$. Ces vecteurs vérifient l'ensemble des contraintes de premier et deuxième niveaux, puis nous nous assurons qu'ils sont bien solutions optimales du problème de second niveau.

4.2 Génération initiale de coupes

Après avoir effectué quelques tests préliminaires sur l'algorithme précédent, nous avons réalisé qu'à la première itération, la plupart des patrons ont des prix nuls et seuls les patrons choisis par l'algorithme ont des prix strictement positifs (voir section 5.2.3). Ces prix, nuls au départ, augmentent au fur et à mesure de la résolution, car à chaque itération nous ajoutons des coupes sur les prix des patrons.

Nous avons généré des coupes initiales additionnelles afin de pallier à ce problème. Nous considérons en premier lieu le problème de second niveau, et le résolvons pour chaque patron et pour chaque client, tout en forçant chaque patron à faire partie de l'allocation optimale pour chaque client. Nous appelons p_c le patron p introduit de

force dans l'allocation optimale d'un client c .

$\forall (c, p_c) \in C \times P^1$ on résout :

$$\begin{aligned} & \min_{y^2} \sum_{p \in P^2} (t_{c,p}^2 + w_{c,p}^2) \cdot y_{c,p}^2 \\ \text{s.à} \quad & \sum_{\substack{s,l \in U^1,L \\ D(c) \in D(s)}} a_{s,l,d}^1 \cdot \delta_{s,l,p_c}^1 \cdot y_{c,p_c}^1 + \sum_{\substack{s,l \in U^2,L \\ p \in P^2 \\ D(c) \in D(s)}} a_{s,l,d}^2 \cdot \delta_{s,l,p}^2 \cdot y_{c,p}^2 \geq \underline{A}_c \\ & y_{c,p_c}^1 = 1 \end{aligned}$$

Ce problème nous retourne l'allocation optimale $(y_{c,p_c}^1, y_{c,p}^{2,*})$. On introduit une coupe relative à chaque patron et à chaque client à l'aide de l'allocation optimale :

$$\begin{aligned} \sum_{p \in P^1} (T_{c,p}^1 + w_{c,p}^1 \cdot y_{c,p}^1) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \leq \\ \sum_{p \in P^1} (t_{c,p}^1 + w_{c,p}^1) \cdot y_{c,p_c}^1 + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^{2,*} \quad \forall c, p_c \in C \times P^1 \end{aligned}$$

Soit, pour résumer :

Algorithme de génération de coupes initiales - A1

Phase 0 : Initialisation

- On résout pour chaque client le problème de sac-à-dos nous donnant le meilleur prix proposé par la concurrence. On obtient alors R_c et on ajoute pour chaque client la contrainte de prix au problème relaxé.
- $i = 0$

Phase 0bis : Génération additionnelle de coupes

$\forall (c, p_c) \in C \times P^1$:

- On résout le problème de second niveau en forçant chaque patron du leader, et ce pour chaque client, à faire partie de la solution optimale ($y_{c,p_c}^1 = 1$).
- Ce problème nous donne une solution optimale $y_{c,p_c}^1, y_{c,p}^{2,*}$.
- On introduit une coupe relative à ce client et au patron du leader :

$$\sum_{p \in P^1} (T_{c,p}^1 + w_{c,p}^1 \cdot y_{c,p}^1) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \leq \sum_{p \in P^1} (t_{c,p}^1 + w_{c,p}^1) \cdot y_{c,p_c}^1 + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^{2,*} \quad \forall c \in C$$

Phase 1 : On résout le problème relaxé (MIP). On obtient $t^*, y^{1,*}, y^{2,*}$.

Phase 2 :

- On résout le problème de second niveau avec t^* .
- On obtient $y_{c,p}^{1,inf}(i)$ et $y_{c,p}^{2,inf}(i)$.
- Pour chaque client c , on vérifie que le prix payé est le même dans le problème relaxé et le problème de second niveau.
- Si c'est le cas pour chaque client, alors $(t^*, y^{1,*}, y^{2,*})$ est une solution optimale. L'algorithme se termine.
- Sinon on passe à la phase 3.

Phase 3 : Génération des coupes

- Pour chaque client c ne vérifiant pas l'égalité énoncée à la phase 2, on introduit

la coupe :

$$\sum_{p \in P^1} T_{c,p}^1 + w_{c,p}^1 \cdot y_{c,p}^1 + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \leq \sum_{p \in P^1} (t_{c,p}^1 + w_{c,p}^1) \cdot y_{c,p}^{1,inf}(i) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^{2,inf}(i) \quad \forall c \in C(i)$$

- $i = i + 1$
- On retourne ensuite à la phase 1.

On ajoute alors $|C| \cdot |P^1|$ coupes initiales avant chaque résolution ($|X|$ étant la dimension de l'ensemble X), en espérant que cela diminuera le nombre d'itérations, accélérant ainsi la résolution.

4.3 Algorithme de relaxation sous-optimale

Le gap d'optimalité d'un MIP est la différence entre une borne supérieure sur la meilleure solution possible et la meilleure solution connue à un instant donné de la résolution. Au début de la résolution du MIP, ce gap (mesuré en % ici) est assez important, et diminue au fur et à mesure du *branch-and-bound*, jusqu'à devenir nul quand on arrive à l'optimalité.

Nous avons observé que les problèmes mixtes en nombres entiers (étape 1 des algorithmes *A0* et *A1*) prennent beaucoup de temps lors de chaque itération. Nous avons également remarqué que le gap se réduit à 10% en quelques secondes, mais que la phase de transition de 10% à 0% prend le plus de temps (environ 95% du temps de résolution).

Notre algorithme agit sur le gap d'optimalité, le fixant à des valeurs positives lors

des premières itérations (on “relaxe” ce gap), et le diminuant jusqu’à allant le fixer à 0 pour les dernière itérations. En effet, nous développons un algorithme de résolution exacte : la résolution doit être exacte lors de la dernière itération. Nous gardons ainsi la même structure algorithmique que l’algorithme $A0$, mais en relaxant le gap du problème relaxé pour les premières itérations. Nous le fixons ainsi en première instance à $INIT_{GAP}$, puis nous le diminuons en le multipliant par $DECREASE_{GAP}$, jusqu’à le fixer à 0 lorsque le nombre de “clients satisfaits”¹ dépasse $CLIENT_{MIN}\%$ du nombre de clients total.

Lors des résolutions avec l’algorithme $A0$, on constate en effet que le nombre de “clients satisfaits” augmente au fur et à mesure des itérations. Lors des premières itérations seul les prix des patrons alloués sont positifs. Voulant minimiser les coûts perçus par client, le problème de sac-à-dos alloue les patrons ayant des prix nuls. Les prix des patrons varient avec l’ajout progressif de coupes, ainsi le nombre de “clients satisfaits” augmente avec le nombre d’itérations. Nous utilisons alors cet indicatif pour savoir quand fixer le gap à 0 : lorsque le nombre de “clients satisfaits” est proche du nombre total de clients, les algorithmes $A0$ et $A1$ sont généralement proches de l’optimalité.

Voici le déroulement de l’algorithme :

Algorithme de relaxation Sous-optimale - A2

Phase 0 : Initialisation

- On résout pour chaque client le problème de sac-à-dos nous donnant le meilleur prix proposé par la concurrence. On obtient alors R_c et on ajoute pour chaque client la contrainte de prix au problème relaxé.
- On initialise $INIT_{GAP}$ et $DECREASE_{GAP}$.
- On fixe le GAP du MIP à $INIT_{GAP}$.

¹clients dont les coûts perçus renvoyés par le problème de sac-à-dos sont supérieurs ou égaux à ceux renvoyés par le problème relaxé.

- $i = 0$

Phase 1 : On résout le problème relaxé . On obtient t^* , $y^{1,*}$ et $y^{2,*}$.

Phase 2 :

- On résout le problème de second niveau avec t^* .
- On obtient $y_{c,p}^{1,inf}(i)$ et $y_{c,p}^{2,inf}(i)$.
- Pour chaque client c , on vérifie que son coût perçu renvoyé par le problème de second niveau est supérieur ou égal à celui renvoyé par le problème relaxé.
- Si c'est le cas pour tous les clients, alors $(t^*, y^{1,*}, y^{2,*})$ est une solution optimale. L'algorithme se termine.
- Sinon on passe à la phase 3.

Phase 3 : Génération des coupes

- Pour chaque client c ne vérifiant pas l'inégalité énoncée à la phase 2, on introduit la coupe :

$$\sum_{p \in P^1} (T_{c,p}^1 + w_{c,p}^1 \cdot y_{c,p}^1) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^2 \leq \sum_{p \in P^1} (t_{c,p}^1 + w_{c,p}^1) \cdot y_{c,p}^{1,inf}(i) + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) \cdot y_{c,p}^{2,inf}(i) \quad \forall c \in C(i)$$

- Si moins de $CLIENT_{MIN}\%$ des clients vérifient l'inégalité énoncée à la phase 2, $GAP(i) = GAP(i) * DECREASE_{GAP}$.
- Sinon $GAP(i) = 0$.
- $i = i + 1$
- On retourne ensuite à la phase 1.

Chapitre 5

RÉSOLUTION NUMÉRIQUE

Dans ce dernier chapitre, nous décrivons dans la première section la génération des données. Dans la section 5.2, nous analysons le comportement des algorithmes proposés sur un jeu de données initial. En partant de ce jeu de données, nous procédons dans la section 5.3 à une analyse de sensibilité sur les paramètres du modèle. Nous terminons ce chapitre par une note sur les pertes de revenu liées à certaines hypothèses, ainsi que la description de la résolution d'un exemple de type *prime time*¹.

5.1 Les paramètres de simulation

5.1.1 Choix de génération de données

Les catégories de patrons

Nous avons généré les patrons en nous inspirant de notre expérience de téléspectateur. Nous avons décidé de créer 12 types de patrons, que nous décrivons ci-dessous :

$P1$: 1 spot de longueur l sur z pauses s consécutives ;

$P2$: 2 spots de longueur l sur z pauses s consécutives ;

$P3$: 2 spots de longueurs différentes l_1 et l_2 sur z pauses s consécutives ;

$P4$: 1 spot de longueur l une pause s sur deux, sur z pauses ($s = 2i$ ou $s = 2i + 1$, $i \in N$) ;

$P5$: 2 spots de même longueur l une pause s sur deux, sur z pauses ($s = 2i$ ou $s = 2i + 1$, $i \in N$) ;

¹Le *prime time* est la tranche horaire correspondant au début de la soirée, 19h-23h généralement.

$P6$: 2 spots de longueurs différentes l_1 et l_2 une pause s sur deux, sur z pauses
 $(s = 2i \text{ ou } s = 2i + 1, \quad i \in N)$;

$P7$: 1 spot d'une longueur l sur une pause s aléatoire ;

$P8$: 2 spots de même longueur sur une pause s aléatoire ;

$P9$: 2 spots de longueurs différentes l_1 et l_2 sur une pause s aléatoire ;

$P10$: 1 spot de longueur l sur z pauses s aléatoires ;

$P11$: 2 spots de même longueur sur z pauses s aléatoires ;

$P12$: 2 spots de longueurs différentes sur z pauses s aléatoires.

L'avantage de cette répartition est de couvrir au maximum l'inventaire du leader, pour éviter au maximum les conflits entre les patrons lors de l'allocation. Nous décidons de regrouper les patrons ayant les mêmes caractéristiques, ainsi trois catégories de patrons se distinguent :

$T1$: La première catégorie regroupe les patrons $P1$ à $P6$;

$T2$: La deuxième catégorie regroupe les patrons $P7$ à $P9$;

$T3$: La troisième catégorie regroupe les patrons $P10$ à $P12$.

Nous allons désormais décrire les caractéristiques des patrons de chaque catégorie.

Patrons de Type $T1$

La catégorie $T1$ regroupe les patrons "réguliers". Ces patrons contiennent des spots apparaissant à toutes les pauses publicitaires, ou une pause sur deux. Ils intéressent généralement les clients faisant beaucoup de publicité.

Patrons de Type $T2$

La catégorie $T2$ regroupe les patrons "ponctuels". Ces patrons ne contiennent qu'un ou deux spots publicitaires lors de la même pause. Ils intéressent alors les clients ne faisant pas beaucoup de publicité.

Patrons de Type $T3$

La catégorie $T3$ regroupe les patrons “aléatoires”, car les spots qu’ils définissent peuvent être répartis sur une soirée, de manière aléatoire. Cette catégorie ressemble à la catégorie $T1$, dans le sens où les patrons proposés ont globalement le même audimat. La différence est que la programmation des spots dans ces patrons est réalisée sur des pauses publicitaires aléatoires, ce qui peut potentiellement intéresser d’autres clients.

Le regroupement des patrons en trois catégories nous pousse alors à segmenter les clients, car chaque client sera plus ou moins intéressé par certains types de patron.

Les différentes catégories de clients

Nous avons regroupé les clients en trois catégories :

$C1$: Clients intéressés par des patrons de type $T1$, et aussi par quelques patrons de type $T2$;

$C2$: Clients intéressés uniquement par des patrons de type $T2$;

$C3$: Clients intéressés par des patrons de type $T3$, et aussi par quelques patrons de type $T2$.

Clients de type $C1$

Les clients de type $C1$ sont intéressés majoritairement par des patrons de type $T1$, donc ces clients font beaucoup de publicité. Parmi ces clients on pourrait retrouver des concessionnaires automobiles, ou des opérateurs téléphoniques. Ces clients sont également intéressés par quelques patrons de type $T2$. On utilise les patrons “ponctuels” $T2$ pour combler l’audimat du client lorsqu’un patron de type $T1$ ne suffit pas à répondre à sa demande, alors que deux patrons de type $T1$ touchent trop de public par rapport à son objectif d’audience.

Par exemple, un client $C1$ pourrait se voir attribuer un patron $T1$ de cinq spots publicitaires, répartis sur cinq pauses consécutives, et un patron $T2$ d'un spot publicitaire programmé sur une autre pause de la soirée.

Clients de type $C2$

Les clients de type $C2$ sont uniquement intéressés par des patrons de type $T2$. Ces clients font peu de publicité, car un ou deux spots par *prime time* sont nécessaires pour satisfaire leur objectif d'audience. Il arrive souvent lors d'une soirée que l'on voit une publicité apparaître ponctuellement.

Clients de type $C3$

Les clients de type $C3$ ressemblent aux clients de type $C1$, dans le sens où ils font beaucoup de publicité. Contrairement aux clients $C1$ qui préfèrent faire une publicité ciblée lors d'une soirée de *prime time*, ceux-ci se contentent de pauses aléatoires, pourvu que leur objectif d'audience soit atteint.

Le processus d'achat/vente des patrons

Nous modélisons le processus d'achat/vente des patrons de la façon suivante. Nous considérons que la chaîne leader propose aux clients une grille complète de patrons. Cependant, les clients ne sont pas intéressés par tous les patrons proposés par le leader. En effet les clients peuvent avoir des préférences pour des patrons se situant en début de soirée par exemple, ou pour des patrons contenant deux spots par pause publicitaire. C'est pourquoi nous restreignons l'accès de certains patrons aux clients.

Nous appelons P_c^* l'ensemble des patrons qui intéressent un client c . Par exemple, la chaîne leader propose une grille de 20 patrons, et le client 2 est intéressé par trois patrons, disons les patrons p_1 , p_3 et p_7 . Nous aurons alors :

$$\{P_2^* = p_1, p_3, p_7\}.$$

Nous définissons ainsi une matrice d'incidence $A_{c,p}$ de taille $|A| \cdot |C|$, dont les coefficients sont égaux à 1 si le client c est intéressé par le patron p , et 0 sinon, soit :

$$a_{c,p} = 1 \quad \forall (c,p) \in C \times P_c^*$$

Nous utiliserons cette matrice pour générer dynamiquement les variables binaires $y_{c,p}^1$ telles que $a_{c,p} = 1$.

5.1.2 Les paramètres du modèle

Dans cette sous-section, nous décrivons les quatre types de paramètres du modèle, et expliquons la manière dont nous les générons.

Les paramètres régissant la capacité du leader

Un premier groupe de paramètres apparaît naturellement, formé des paramètres relatifs aux contraintes de capacité du leader.

Les contraintes concernées sont :

$$\begin{aligned} \sum_{c \in C} \sum_{p \in P^1} \sum_{l \in L} \alpha_l \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 &\leq l(s) & \forall s \in U^1 \\ \sum_{c \in C} \sum_{p \in P^1} \delta_{s,l,p}^1 \cdot y_{c,p}^1 &\leq I_{s,l}^1 & \forall s, l \in U^1 \times L \end{aligned}$$

On considère ici la longueur $l(s)$ des pauses publicitaires s , le nombre de longueurs de spots L , le nombre de pauses publicitaires U^1 proposées par la chaîne leader et l'indice $I_{s,l}^1$.

L'influence des paramètres $l(s)$, U^1 et $I_{s,l}^1$ peut être considérable : les augmenter permettra à la chaîne leader de vendre plus de patrons. Si ces paramètres diminuent, moins de patrons pourront alors être achetés et le choix de ces patrons sera restreint, car il y aura moins de capacité disponible pour les clients. Toutes ces possibilités

peuvent influencer sur les temps de résolution et sur le revenu, et il est donc important de les considérer.

Le nombre de longueurs de spots L est également un paramètre important, car plus les spots sont longs moins il est facile de les programmer, car il y a moins de flexibilité concernant les différentes combinaisons de spots lors d'une pause. Cela se retrouvera dans la contrainte de longueur des pauses publicitaires.

Nous fixons de manière déterministe certains de ces paramètres lors des tests effectués sur les algorithmes : U^1 et L . Nous aborderons dans la section 5.3 la manière dont ils sont déterminés lors de la campagne de tests.

Si l'on décide par exemple de prendre 4 longueurs de spots, celles-ci seront 15 secondes, 30 secondes, 45 secondes et 60 secondes.

Les deux paramètres $l(s)$ et $I_{s,l}^1$ sont générés aléatoirement :

- La longueur $l(s)$ d'une pause publicitaire s est générée aléatoirement entre deux bornes, que nous déterminerons dans la sous-section 5.3.
- L'indice $I_{s,l}^1$ est généré aléatoirement entre deux bornes fixées arbitrairement, mais de façon à ne pas rendre la contrainte inactive (voir sous-section 5.3).

Les paramètres associés aux clients

Le deuxième groupe de paramètres concerne les clients, soit :

- Le nombre de clients C ;
- L'audience A_c voulue par un client c ;
- La répartition des catégories de clients.

Le nombre de clients intervient à tous les niveaux, car il définit le nombre de variables binaires $y_{c,p}^i$ du modèle. Il caractérise aussi le nombre de contraintes de deuxième niveau :

$$\sum_{\substack{s,l \in U^1, L \\ p \in P^1 \\ d(c) \in D(s)}} a_{s,l,d}^1 \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 + \sum_{\substack{s,l \in U^2, L \\ p \in P^2 \\ d(c) \in D(s)}} a_{s,l,d}^2 \cdot \delta_{s,l,p}^2 \cdot y_{c,p}^2 \geq A_c \quad \forall c \in C$$

Cette contrainte est également paramétrée par l'audimat A_c qu'un client c veut atteindre. Ce paramètre est non négligeable, car plus l'audimat sera élevé, plus il y aura de combinaisons possibles de patrons comblant l'audimat d'un client, ce qui pourra éventuellement influencer les temps de résolution.

La répartition des clients concerne les catégories de clients : en effet on peut choisir de créer des instances avec une majorité de clients de type $C2$ par exemple (voir section 5.3). Le nombre de clients est également fixé arbitrairement, nous le ferons varier d'une certaine manière pour mener à bien nos tests numériques.

L'audimat A_c qu'un client c veut réaliser en faisant de la publicité a été établi à l'aide d'une loi uniforme sur $[A, B]$, en prenant une borne inférieure (A) et une borne supérieure (B) sur l'audimat. Pour chaque catégorie de patron, nous prenons le patron avec la plus petite audience et le patron avec la plus grande audience. Ensuite nous générons un audimat pour la catégorie de clients correspondante à sa catégorie de patron, comprise entre l'audimat minimal et deux fois l'audimat maximal. De cette façon, le client a généralement besoin d'entre deux et trois patrons pour atteindre son objectif d'audience.

Par exemple, si le plus petit audimat des patrons $T2$ est égal à 300 ($Aud_{\min}(T2) = 300$) et si le plus grand audimat des patrons $T2$ est égal à 600 ($Aud_{\max}(T2) = 600$), nous allons générer un audimat A_c pour les clients c appartenant à la catégorie $C2$ tel que : $300 \leq A_c \leq 1200$.

Les paramètres associés aux patrons

Le troisième et dernier groupe de paramètres ainsi défini caractérise les patrons, soit :

- Le nombre P^1 de patrons vendus par le leader ;
- Le nombre P^2 de patrons vendus par la concurrence ;
- Le nombre P_c^* de patrons qui intéressent les clients.

Au même titre que le nombre de clients C , le nombre de patrons P^i définit le nombre de variables binaires $y_{c,p}^i$, qui joue un rôle important dans la résolution du programme linéaire en nombres entiers.

Le nombre de patrons qui intéressent les clients est également important : nous générons en effet seulement les variables binaires $y_{c,p}^1$ pour lesquelles $p \in P_c^*$, donc augmenter ou diminuer ce paramètre augmente ou diminue le nombre de variables.

Le nombre de patrons p_c^* qui intéressent les clients est également généré aléatoirement à l'aide d'une loi uniforme. Lors de la génération des données, nous faisons varier le nombre de patrons par client np_c entre deux bornes. Puis pour chaque client, nous lui choisissons aléatoirement np_c patrons appartenant à sa catégorie.

Par exemple, si le client 1 de la catégorie $C1$ est intéressé par trois patrons de type $T1$ ($np_1 = 3$), ceux-ci sont choisis au hasard parmi l'ensemble des patrons de type $T1$.

Les paramètres invariants

Les autres paramètres du modèle ne seront pas considérés. Nous allons décrire la manière dont nous les générons, mais nous ne ferons pas d'analyse de sensibilité sur ces paramètres.

Intervenant dans l'équation :

$$\sum_{\substack{s,l \in U^1, L \\ p \in P^1 \\ d(c) \in D(s)}} a_{s,l,d}^1 \cdot \delta_{s,l,p}^1 \cdot y_{c,p}^1 + \sum_{\substack{s,l \in U^2, L \\ p \in P^2 \\ d(c) \in D(s)}} a_{s,l,d}^2 \cdot \delta_{s,l,p}^2 \cdot y_{c,p}^2 \geq A_c \quad \forall c \in C$$

l'audience $a_{s,l,d}^i$ des spots publicitaires de longueur l lors de la pause s pour la catégorie démographique d est générée aléatoirement. Nous engendrons cette audience en prenant une borne inférieure et une borne supérieure, ces deux bornes n'étant pas les mêmes pour les patrons de la chaîne leader et de la concurrence.

Les paramètres de l'objectif du second niveau sont également générés, et ils concernent ainsi la fonction :

$$\min_{y^1, y^2} \sum_{c \in C} \left(\sum_{p \in P^1} (t_p^1 + w_{c,p}^1) y_{c,p}^1 + \sum_{p \in P^2} (t_p^2 + w_{c,p}^2) y_{c,p}^2 \right)$$

Les prix t_p^2 des patrons p vendus par la concurrence varient en fonction des audiences des patrons de la concurrence (le patron est un ensemble de spots, et la somme des audiences de ces spots définit l'audience du patron).

Si on définit $A(p_3)$ comme étant l'audience du patron 3 de la concurrence, on aura $t_3^2 = 2 \cdot A(p_3)$.

La désutilité $w_{c,p}$ d'un client c à l'égard d'un patron p est également générée de manière aléatoire, de la même façon que l'audience des spots.

5.1.3 La publicité à la télévision

Nous avons regardé la télévision pour tenter d'obtenir quelques tendances sur les temps de diffusion, les longueurs des spots et les longueurs des pauses publicitaires. Le tableau 5.1 nous montre les résultats de cette petite étude. La longueur moyenne des spots tourne autour de 24 secondes, ce qui motive notre choix de prendre des spots de

FIGURE 5.1 Lors du *prime time* aux États-Unis

	1 ^{ère} Pause	2 ^{ème} Pause	3 ^{ème} Pause
Longueur des Pauses	210 sec 3 min 30 sec	240 sec 4 min 00 sec	260 sec 4 min 20 sec
Nombre Moyen de Spots	8,5	9,5	11
Longueur Moyenne des Spots	24,7 sec	25,3 sec	23,6 sec

longueurs multiples de 15 secondes. Nos patrons contiendront néanmoins une majorité de spots de 15 et 30 secondes. La longueur moyenne des pauses publicitaires tourne autour de 230 secondes.

Cette étude a également révélé que sur 60 minutes de diffusion à la télévision, 15 minutes sont consacrées à la publicité. Nous utiliserons ces données pour simuler l'allocation et la tarification de patrons de spots publicitaires pour les résolutions numériques.

Nous sommes désormais prêts à décrire les jeux de données qui ont servi à tester nos algorithmes de résolution.

5.2 Tests numériques initiaux

Nous revenons sur les résultats de Chan (2008), pour ensuite définir un jeu de données initial et comparer nos temps de résolution à ceux du modèle M-Spot. Nous étudions également le comportement de nos algorithmes.

5.2.1 Résultats numériques du modèle M-Spot

Nous avons testé l'algorithme initial sur un jeu de données similaire aux jeux de données testés par Chan (2008). Les temps de résolution sont très faibles, de l'ordre du dixième de seconde.

TABLEAU 5.1 $C = 5 - L = 3 - D = 1 - U^1 = 10 - 6$ à 8 spots pour combler l'audimat des clients

Instance	Génération initiale			Génération dynamique		
	Nombre d'itérations	Temps (s)	Gap (%)	Nombre d'itérations	Temps (s)	Gap (%)
1	14	5777,8	0,00	6	3900,5	0,00
2	6	39,4	0,00	6	8,8	0,00
3	19	2633,6	0,00	5	1133,9	0,00
4	*	*	*	*	*	*
5	39	674,2	0,00	20	387,9	0,00
6	28	6283,3	0,00	11	3554,2	0,00
7	5	51,5	0,00	3	10,6	0,00
8	13	60,4	0,00	3	7,3	0,00
9	8	50,4	0,00	4	14,8	0,00
10	13	162,9	0,00	8	123,0	0,00

Le tableau 5.1 présente les temps de résolution pour le jeu de données le plus difficile résolu par le M-Spot.

Ces résultats sont obtenus après avoir résolu leur modèle avec deux méthodes d'ajout de coupes plus élaborées (ajout initial de coupes et ajout dynamique de coupes, le lecteur pour se référer à leur travaux pour plus de détails sur la génération des coupes). Leur algorithme d'ajout dynamique de coupes semble plus efficace que l'algorithme d'ajout initial, mais les temps restent élevés, allant jusqu'à une heure pour deux instances de faible taille.

Nous avons repris le même types de données, et les avons testé avec nos algorithmes. Les temps de résolution étaient tous inférieurs à une seconde, nous avons alors voulu densifier quelque peu le problème. Ces données densifiées constitueront notre jeu de données initial.

5.2.2 Le jeu de données initial

Le jeu de données initial développé ici est un jeu de “taille moyenne”. En effet ce jeu (dont les données sont ci-dessous) représente environ deux heures de *prime time* si l’on se fie aux données citées dans le tableau 5.1.

Voici les données :

- 9 clients ;
- 3 longueurs de spots (15, 30 et 45 secondes) ;
- 10 pauses publicitaires chez le leader ;
- Longueur des pauses comprise entre 150 et 205 secondes ;
- 20 pauses chez la concurrence ;
- 20 patrons chez le leader ;
- 30 patrons chez la concurrence ;
- Entre 2 et 5 patrons intéressent les clients ;
- $I_{s,l}^1$ est compris entre 2 et 5 ;
- Les patrons $T1$ et $T3$ contiennent des spots sur 5 pauses publicitaires.

L’audimat $a_{s,l,d}^1$ des spots proposés par la chaîne leader est compris entre 20 et 50, et l’audimat $a_{s,l,d}^2$ des spots proposés par la concurrence est compris entre 15 et 55. La désutilité $w_{c,p}^1$ d’un client c pour un patron p proposé par la chaîne leader est comprise entre 0 et 50, et la désutilité $w_{c,p}^2$ d’un client c pour un patron p proposé par la concurrence est comprise entre 5 et 55. Ces paramètres seront générées de la même manière pour l’ensemble des tests numériques effectués dans cette section.

Tous nos tests numériques ont été réalisés avec Xpress-Mosel 64-bit v2.2.0 et Sungrid, système qui permet la distribution et la gestion des tâches sur un réseau de machines : Sungrid s’occupe de lancer les simulations sur les ordinateurs du GERAD. Les algorithmes sont testés sur dix jeux de données générés aléatoirement.

Le tableau 5.2 décrit le nombre d'itérations, le nombre de coupes, et les temps de résolution pour l'algorithme *A0*, testé sur dix instances générées aléatoirement à partir du jeu de données initial.

TABLEAU 5.2 Temps de résolution de l'algorithme A0 testé sur le jeu de données initial - **2 à 3 patrons pour combler l'audimat des clients**

Instances	Nombre d'itérations	Nombre de coupes	Temps (s)
1	9	33	8,53
2	9	26	6,67
3	8	36	13,85
4	7	34	32,6
5	6	31	7,46
6	6	24	6,38
7	7	29	8,06
8	9	47	28,59
9	8	32	11,38
10	7	24	3,61

Le tableau 5.3 présente :

- les meilleurs temps de résolution et le nombre d'itérations pour le M-Spot sur le jeu de données le plus difficile résolu dans le mémoire de Chan (2008) ;
- les temps de résolution et le nombre d'itérations pour le M-Patron sur le jeu de données initial, qui s'avère être plus dense que le jeu de données utilisé pour tester le M-Spot.

Le modèle M-Patron est meilleur : les temps de résolution ne dépassent guère la trentaine de secondes. La moyenne des temps de résolution du M-Patron (12,7 secondes) est beaucoup plus faible que celle du M-Spot (1015,7 secondes). Cette observation est très pertinente : cette moyenne est cent fois plus petite sur des données testées encore plus denses que celles utilisées pour le M-Spot. L'utilisation de la formulation du problème avec des patrons permet alors de résoudre des problèmes de taille bien plus grande !

TABLEAU 5.3 Temps de résolution pour le **M-Spot** et le **M-Patron** (pour des instances **différentes**).

M-Spot			M-Patron		
Instance	Nombre d'itérations	Temps (s)	Instance	Nombre d'itérations	Temps (s)
1	6	3900,5	1	9	8,5
2	6	8,8	2	9	6,7
3	5	1133,9	3	8	13,9
4	*	*	4	7	32,6
5	20	387,9	5	6	7,5
6	11	3554,2	6	6	6,4
7	3	10,6	7	7	8,1
8	3	7,3	8	9	28,6
9	4	14,8	9	8	11,4
10	8	123,0	10	7	3,6

5.2.3 Commentaires sur le comportement de l'algorithme $A0$

Lors de la résolution du problème relaxé, seuls les patrons alloués (proposés par le leader) ont un prix strictement positif dès la première itération : le vecteur de prix des patrons comporte beaucoup de prix nuls. Ainsi, le problème de second niveau alloue aux clients des patrons de prix nuls, puis des coupes sont générées sur les patrons choisis, et plusieurs itérations s'enchaînent avant d'obtenir des prix strictement positifs. Une fois tous les prix des patrons strictement positifs, il suffit généralement de deux itérations pour atteindre l'optimalité.

L'allocation ne change pas beaucoup lors de la résolution. Premièrement, les clients ne sont intéressés que par un nombre assez restreint de patrons (3-5 patrons). Deuxièmement, cet intérêt est généré aléatoirement (à l'aide d'une loi uniforme). Dans une série de données de petite taille par exemple, entre un et trois patrons intéressaient le client parmi les cinq patrons proposés par la chaîne leader. Il est arrivé dans certaines instances qu'un client ne soit intéressé que par un seul patron.

Il est arrivé également qu'un patron n'intéresse qu'un seul client.

À cause de ces hypothèses, l'aspect combinatoire du modèle est faible parce que les clients ne se "battent" pas pour les patrons, et il peut arriver qu'ils ne soient qu'un ou deux à être intéressés par le même patron. Il n'y a donc pas beaucoup d'interaction entre les clients dans ce modèle, et c'est pourquoi l'allocation donnée dès la première itération ne change guère.

Le tableau 5.4 montre les correspondances entre les clients et les patrons proposés par le leader, sur deux instances de ce même jeu de données (instance γ et instance β). Comme nous pouvons le voir, il y a "conflit" sur quatre patrons, et ce entre deux clients à chaque fois. Sur l'instance β , ce conflit n'arrive que trois fois.

Nous avons aussi testé l'algorithme sur des données dans lesquelles le client est intéressé par un certain nombre de patrons. L'algorithme a également été testé avec les mêmes données, mais dans une configuration dans laquelle le client peut avoir accès à tous les patrons. Dans le cas où le client est intéressé par un nombre restreint de patrons, l'allocation a changé une fois pour dix instances testées. Pour les neuf autres instances, l'allocation est restée la même bien que les prix aient changé. Dans le cas contraire (pas de préférences), l'allocation a changé sept fois sur dix instances testées. Dans les trois autres simulations l'allocation est restée la même mais les prix des patrons ont changé.

5.2.4 Algorithme A1

Nous avons utilisé le jeu de données initial pour tester l'algorithme avec génération initiale de coupes (algorithme A1). Comme le montre le tableau 5.5, l'algorithme A1 est beaucoup plus efficace que l'algorithme A0.

Comme nous pouvons le constater, le nombre d'itérations est moins élevé pour l'algorithme A1, ceci étant dû à l'ajout initial de coupes. Cet ajout est également visible, et il y a effectivement plus de coupes générées par l'algorithme A1, et ce pour

TABLEAU 5.4 Correspondances clients/patrons

	Instance γ						Instance β					
Clients Patrons	1	2	3	4	5	6	1	2	3	4	5	6
1							X					
2	X	X										
3							X					
4	X	X						X				
5							X	X				
6									X			
7										X		
8				X						X		
9				X								
10			X	X				X				
11			X					X	X			
12					X	X					X	
13										X	X	
14						X						

TABLEAU 5.5 Temps de résolution des algorithmes A0 et A1 pour le jeu de données initial - 2 à 3 patrons pour combler l'audience des clients

Instances	Algorithme A0			Algorithme A1		
	Nombre d'itérations	Nombre de coupes	Temps (s)	Nombre d'itérations	Nombre de coupes	Temps (s)
1	9	33	8,53	4	34	4,92
2	9	26	6,67	3	33	3,23
3	8	36	13,85	3	40	4,61
4	7	34	32,60	4	45	15,78
5	6	31	7,46	2	35	3,30
6	6	24	6,38	2	32	1,94
7	7	29	8,06	4	40	4,63
8	9	47	28,59	2	39	4,77
9	8	32	11,38	5	42	12,23
10	7	24	3,61	4	36	2,87

chaque instance.

La réduction du nombre d'itérations confirme notre intuition concernant la diminution des temps de résolution : ceux-ci sont divisés par deux sur les instances 1, 2, 4, 5, 7. Ils sont sensiblement équivalents sur les instances 9 et 10. Les temps sont divisés par trois sur les instances 3 et 6 et par cinq sur l'instance 8.

Sur l'ensemble des instances évaluées lors de cette campagne d'essais, ces statistiques se sont répétées : dans les pires cas l'algorithme $A1$ avait des temps équivalents à l'algorithme $A0$, environ deux à trois fois sur dix. Dans les autres cas les temps de résolution étaient largement inférieurs.

5.2.5 Algorithme de Relaxation Sous-optimale $A2$

Nous avons fixé à 75% la valeur de $CLIENT_{MIN}$ de manière empirique. En effet nous avons remarqué que dans la plupart des cas, lorsqu'environ 75% des clients sont satisfaits, il ne reste plus beaucoup d'itérations à réaliser avant d'atteindre l'optimum. Nous avons également choisi de fixer $GAP_{INIT} = 10\%$ et $DECREASE_{GAP} = 0.7$. Le tableau 5.6 compare les temps de résolution de l'algorithme $A2$ avec ceux de l'algorithme $A0$. Il ne sont pas très satisfaisants, et nous expliquons par la suite les limites de l'algorithme $A2$.

Lors des premiers tests effectués avec l'algorithme $A2$, nous nous sommes aperçus que les MIP qui prenaient le plus de temps étaient les MIP des dernières itérations, car apparemment le fait d'ajouter des coupes complexifie le problème. Les MIP contribuent à environ 90% du temps de résolution total. Étant donné que le gap est fixé à zéro lors des dernières itérations, cela rend notre algorithme peu efficace.

Également, la barre du taux de clients satisfaits de 75% n'est pas toujours bonne. Dans 10 % des cas, il arrive que l'on ait moins de 75% de clients satisfaits une itération avant l'itération finale. L'itération finale est alors inexacte, car le gap utilisé pour résoudre le MIP n'est pas nul, ce qui nous donne une solution non-optimale. En

effet le taux a été fixé empiriquement à 75%, mais on ne peut pas toujours prévoir comment la résolution évoluera, et notre algorithme devient alors heuristique, s'approchant généralement à moins de 1% de la solution optimale (instances 7 et 10 dans le tableau 5.6).

Cet algorithme de relaxation sous-optimale ne peut pas être adapté à l'algorithme *A1*. Par le fait d'ajouter un certain nombre de coupes avant de lancer la résolution, on a généralement 70% des clients satisfaits lors de la première itération avec l'algorithme *A1*, et moins d'itérations sont nécessaires pour obtenir une solution optimale. Si nous relaxons le gap alors qu'une grande partie des clients sont satisfaits, on risque de voir l'algorithme se terminer en une itération, et la solution obtenue est alors sous-optimale.

Suite aux mauvais résultats de l'algorithme *A2*, les expérimentations numériques futures ne concerneront que les algorithmes *A0* et *A1*.

5.3 Plan d'expérimentation

Nous allons dans cette partie considérer les paramètres du modèle, et analyser leur influence sur les temps de résolution. Nous partons du jeu de données de base défini dans la partie précédente, puis nous faisons varier les paramètres de scénarios **un à un** pour voir comment les algorithmes *A0* et *A1* réagissent aux différentes instances qu nous leur soumettons. Nous générons dix jeux de données pour chaque variation des paramètres, et considérons les moyennes des temps de résolution et du revenu.

5.3.1 Les paramètres déterminant la capacité du leader

Dans un premier temps, nous agissons sur les paramètres relatifs aux contraintes de capacité du leader. Nous considérons ici la longueur des pauses $l(s)$, le nombre U^1 de pauses offertes par le leader et l'indice $I_{s,l}^1$.

TABLEAU 5.6 **Temps de résolution** des algorithmes $A0$ et $A2$ pour le jeu de données initial - **2 à 3 patrons pour combler l'audience des clients**

Instances	Algorithme $A0$			Algorithme $A2$		
	Nombre d'itérations	GAP (%)	Temps (s)	Nombre d'itérations	GAP (%)	Temps (s)
1	9	0.00	8,53	9	0.00	7,58
2	9	0.00	6,67	13	0.00	13,38
3	8	0.00	13,85	9	0.00	17,37
4	7	0.00	32,60	11	0.00	30,59
5	6	0.00	7,46	9	0.00	10,81
6	6	0.00	6,38	7	0.00	3,57
7	7	0.00	8,06	5	0.87	2,24
8	9	0.00	28,59	9	0.00	31,78
9	8	0.00	11,38	11	0.00	20,62
10	7	0.00	3,61	6	0.05	2,34

Variation de la longueur $l(s)$ des pauses publicitaires s

Un premier résultat évident est obtenu en faisant varier la longueur des pauses publicitaires : le revenu du leader augmente de 9,69% quand on augmente la longueur des pauses et diminue de 17,88% quand on diminue cette même longueur. Ce résultat semble intuitif ; en effet si on augmente le temps de diffusion, on donne la possibilité au leader de placer plus de spots ; il pourra alors vendre plus de patrons. Le raisonnement inverse tient également lorsqu'on diminue cette longueur.

Concernant l'algorithme $A0$, le fait de diminuer $l(s)$ fait diminuer le temps de résolution d'une seconde (négligeable). On aurait pu penser le contraire, car si on diminue la capacité du leader, alors cela complexifie le problème.

Cependant, cette diminution fait augmenter le temps de résolution pour l'algorithme $A1$, ce qui semble confirmer l'intuition précédente.

Par contre, quand on augmente la longueur des pauses, les temps de résolution sont inférieurs dans les deux cas : la moyenne de ces temps est divisée par deux pour

l'algorithme $A0$, et diminue de 50% pour l'algorithme $A1$. Le tableau 5.7 contient la moyenne des temps, leur écart-type ainsi que le revenu moyen pour les dix instances testées.

Augmentation du nombre U^1 de pauses offertes par le leader

Nous avons dans un premier temps augmenté le nombre de pauses proposées par le leader. Le nombre de patrons disponibles est resté le même, et les patrons découpent l'espace publicitaire sur l'ensemble des pauses. Comme le montre le tableau 5.8, le revenu a augmenté (ce qui semble intuitif) de 17,19%. L'inventaire est plus grand donc plus de patrons sont disponibles. En effet un client peut acheter plusieurs patrons, et le même patron peut être acheté par plusieurs clients.

Le temps a également diminué, dans les mêmes proportions que lorsqu'on a augmenté la longueur des pauses, que ce soit pour l'algorithme $A0$ ou l'algorithme $A1$: en effet la capacité du leader augmente, ce qui facilite la résolution. Le tableau 5.8 récapitule la moyenne des temps, l'écart-type et le revenu moyen pour le changement de ce paramètre.

Variation de l'indice $I_{s,l}^1$

Toujours avec le même raisonnement que précédemment, nous avons constaté (tableau 5.9) qu'augmenter l'indice $I_{s,l}^1$ augmente le revenu de 14,18% et le diminue de 19,14% quand on diminue l'indice. Cependant les temps de résolution ne sont pas affectés, et ils diminuent dans les deux cas (tableau 5.9). La diminution de cet indice a eu le même impact que la diminution du nombre de pauses : les temps de résolution n'ont pas été affectés. Également, l'augmentation du revenu est sensiblement la même que celle occasionnée par l'augmentation du nombre de pauses.

TABLEAU 5.7 **Paramètres de capacité du leader** - Variation de la longueur $l(s)$ des pauses s

Algorithme A0	$105 \leq l(s) \leq 150$	$150 \leq l(s) \leq 205$	$180 \leq l(s) \leq 235$
Temps moyen (s)	10,48	12,71	7,35
Écart-type (s)	6,42	9,36	1,97
Variation du revenu (%)	-17,88	0	+9,69

Algorithme A1	$105 \leq l(s) \leq 150$	$150 \leq l(s) \leq 205$	$180 \leq l(s) \leq 235$
Temps moyen (s)	8,07	5,83	3,59
Écart-type (s)	8,43	4,26	1,98
Variation du Revenu (%)	-17,88	0	+9,69

TABLEAU 5.8 **Paramètres de capacité du leader** - Variation de U^1

Algorithme A0	$l = 10, c = 20$	$l = 15, c = 20$
Av. Time (s)	12,71	6,20
Std. Dev. (s)	9,36	4,07
Revenue Var. (%)	0	+17,19

Algorithme A1	$l = 10, c = 20$	$l = 15, c = 20$
Av. Time (s)	5,83	3,40
Std. Dev. (s)	4,26	2,94
Revenue Var. (%)	0	+17,19

TABLEAU 5.9 **Paramètres de capacité du leader** - Variation de l'indice $I_{s,l}^1$

Algorithme A0	$1 \leq I_{s,l}^1 \leq 3$	$2 \leq I_{s,l}^1 \leq 5$	$3 \leq I_{s,l}^1 \leq 6$
Average Time (s)	29,77	12,71	6,26
Standard Deviation (s)	71,53	9,36	3,52
Revenue Variation (%)	-19,14	0	+14,18

Algorithme A1	$1 \leq I_{s,l}^1 \leq 3$	$2 \leq I_{s,l}^1 \leq 5$	$3 \leq I_{s,l}^1 \leq 6$
Average Time (s)	4,23	5,83	5,11
Standard Deviation (s)	7,22	4,26	2,89
Revenue Variation (%)	-19,14	0	+14,18

TABLEAU 5.10 Paramètres de capacité du leader - Variation de l

Algorithme A0	$L = 2$	$L = 3$	$L = 4$
Temps Moyen (s)	11,97	12,71	41,87
Ecart-type (s)	19,66	9,36	63,43
Variation du revenu (%)	-2,00	0	-1,61

Algorithme A1	$L = 2$	$L = 3$	$L = 4$
Temps Moyen (s)	4,96	5,83	10,72
Ecart-type (s)	8,15	9,36	32,21
Variation du revenu (%)	-2,00	0	-1,61

Le nombre de longueurs de spots l

Nous avons fait varier le nombre de longueurs de spots que la chaîne leader propose dans ses patrons.

Comme le montre le tableau 5.10, le revenu ne change quasiment pas. Donc, dans notre modèle la chaîne leader ne gagnera pas plus d'argent en diversifiant ses longueurs de spots. Également, quand nous choisissons d'imposer deux longueurs de spots, les temps de résolution ne changent guère. Cependant, quand nous fixons L à 4, les temps sont multipliés par trois pour l'algorithme A0 et par deux pour l'algorithme A1. Premièrement, le fait d'augmenter L accroît le nombre d'équations relatives à $I_{s,l}^1$, ce qui complexifie le problème. Deuxièmement, si les spots sont plus longs, alors on peut définir moins facilement des combinaisons de spots satisfaisant la contrainte de longueur des pauses.

En conclusion, les variations des paramètres des contraintes de capacité affectent peu les temps de résolution. En effet elles ne rendent pas le problème trop contraint. Seule l'augmentation du nombre de longueurs de spots perturbe de manière non négligeable le temps de résolution. Les variations de revenu associées aux variations de ces paramètres semblent naturelles.

5.3.2 Les paramètres associés aux clients

Nous avons testé les paramètres relatifs aux clients dans le modèle.

Le nombre de clients

Nous avons repris notre jeu de données de base, puis réalisé des tests en prenant 6 clients, puis 12 clients. Le nombre de clients est impliqué dans les deux sous-problèmes, car il définit le nombre de variables binaires $y_{c,p}^i$, et il est engagé dans l'équation d'audience qui sera dans la prochaine sous-section, ainsi que dans la génération des coupes (lorsque l'on génère les coupes, nous comparons les revenus pour chaque client). Le tableau 5.11 nous montre que les revenus ont augmenté de 15,43% quand nous avons porté le nombre de clients à 12, puis diminué de 25,10% quand nous l'avons fixé à 6. Nous remarquons que le revenu a plus diminué quand nous avons retiré 3 clients, ce qui est normal car on augmente de 50% le nombre de clients quand on passe de 6 à 9, et de 33% quand on passe de 9 à 12 clients, donc la fluctuation du revenu est moindre.

Les temps de résolution ont également beaucoup changé. Quand nous n'avons considéré que 6 clients, les temps ont diminué, ce qui peut paraître logique car les 6 clients ont alors à disposition le même inventaire que pour 9 clients.

TABLEAU 5.11 **Paramètres des clients** - Variation du nombre de clients

Algorithme A0	6 clients	9 clients	12 clients
Temps moyen (s)	5,54	12,71	84,66
Écart-type (s)	3,58	9,36	121,81
Variation du revenu (%)	-25,10	0	+15,43

Algorithme A1	6 clients	9 clients	12 clients
Temps moyen (s)	4,71	5,83	27,88
Écart-type (s)	3,58	4,26	39,33
Variation du revenu (%)	-25,10	0	+15,43

Cependant, quand nous sommes passés à 12 clients, les temps de résolution ont tout de suite augmenté, étant multipliés par sept pour l'algorithme $A0$, et par cinq pour l'algorithme $A1$. Le problème est alors devenu plus contraint : les contraintes d'inventaire ont joué un rôle, car leur capacité est restée fixe alors que le nombre de clients a augmenté de 33%.

On note que le nombre de clients est important, car il est impliqué à tous les niveaux du problème relaxé, ainsi que lors de la génération de coupes, ce qui explique son influence sur les temps de résolution.

L'audimat A_c

Nous avons généré aléatoirement l'audimat A_c visé par chaque client c compris entre l'audimat du plus petit patron (patron ayant le plus petit audimat) et deux fois l'audimat du plus gros patron (patron ayant le plus gros audimat). Nous avons effectué des tests en changeant les bornes de variation de ces audimats : nous avons gardé la même borne inférieure d'audimat (audimat du plus petit patron) et utilisé :

- soit l'audimat du plus gros patron comme borne supérieure ;
- soit trois fois l'audimat du plus gros patron comme borne supérieure.

TABLEAU 5.12 **Paramètres des clients** - Variation de l'audimat par client A_c

Algorithme $A0$	AudMax	2*AudMax	3*AudMax
Temps moyen (s)	2,75	12,71	2717,09
Écart-type (s)	1,20	9,36	5084,74
Variation du revenu (%)	-10,19	0	+9,66

Algorithme $A1$	AudMax	2*Audmax	3*Audmax
Temps moyen (s)	1,62	5,83	665,12
Écart-type (s)	0,85	4,26	1511,8
Variation du revenu (%)	-10,19	0	+9,66

Le revenu diminue de 10,19% (cf. tableau 5.12, où AudMax représente l’audience du patron ayant le plus gros audimat) quand on diminue l’audience visée par les clients. En effet la perte de revenu n’est pas très grande, car la chaîne leader est en mesure de combler plus de clients. Le revenu augmente de 9,66% quand on augmente l’audience visée par les clients. Dans ce cas-là, la chaîne leader alloue un maximum de patrons, tout en respectant l’inventaire.

Les temps diminuent de manière franche quand on diminue l’audimat visé par client. En effet seulement un à deux patrons sont nécessaires pour combler cet audimat, l’effet combinatoire est faible donc la résolution est efficace : on divise par cinq les temps de résolution. Cependant lorsqu’on accentue l’audimat voulu par client, les temps de résolution augmentent de manière dramatique. Deux résolutions ont duré très longtemps (3 heures pour l’algorithme $A0$). Cependant, l’algorithme $A1$ met en moyenne 400 secondes sur deux instances, et 5000 secondes sur une autre.

Ce résultat peut s’expliquer par le fait qu’il faille désormais plusieurs patrons pour satisfaire l’audience des clients, ce qui induit alors une combinatoire plus forte.

Les catégories de clients $C1$, $C2$ et $C3$

Les clients ayant été segmentés en fonction de leurs préférences de patrons, il peut être intéressant d’observer le comportement des algorithmes si nous changeons la répartition dans ces catégories. Dans le jeu de données initial, nous considérons 9 clients répartis également dans chaque groupe, soit 3 clients de type $C1$, 3 clients de type $C2$ et 3 clients de type $C3$.

Nous avons considéré premièrement une majorité de clients $C1$, soit 7, un client $C2$ et un client $C3$. Le revenu a augmenté de 9,85%, ce qui peut paraître logique : nous avons plus de clients de type $C1$, qui achètent des “gros” patrons, étant par conséquent plus chers. Cependant l’augmentation du revenu n’est pas énorme, car la capacité de la chaîne leader limite la vente de ces patrons.

Les temps de résolution augmentent de manière dramatique : en effet comme observé précédemment, la capacité vient à manquer, ce qui fait exploser les temps de résolution, qui sont 25 fois plus élevés pour l'algorithme *A0* et 20 fois plus élevés pour l'algorithme *A1*.

Nous avons ensuite considéré une majorité de clients de type *C2*, soit 7, un client de type *C1* et un client de type *C3*. Le revenu a diminué de 33,19% ; effectivement nous vendons désormais une majorité de très petits patrons, qui peuvent être assimilés à des spots. Les temps de résolution, quant à eux, sont multipliés par cinq pour l'algorithme *A0*, et par deux pour l'algorithme *A1*. Les patrons sont très petits et la combinatoire est ainsi plus élevée, mais comme l'inventaire reste large, ces temps n'explorent pas.

Nous avons enfin considéré une majorité de clients de type *C3*, dans les mêmes proportions que les deux cas précédents. La catégorie *C3* ressemble fortement à la catégorie de clients *C1*, à la différence que les clients de type *C3* achètent des patrons ayant des spots répartis sur plusieurs pauses aléatoires, alors que les clients *C1* achètent des patrons ayant des spots répartis sur plusieurs pauses consécutives. Cependant la taille des patrons est la même (cinq pauses considérées par patron dans les deux cas), c'est pourquoi ces catégories sont sœurs. Le revenu augmente de 3,25%, ce qui est négligeable. En effet même si les patrons *T3* sont semblables aux patrons *T1*, les patrons *T1* doivent être plus simples à programmer, de par leur régularité. Cependant les temps de résolution augmentent également de manière effroyable : ils sont multipliés par 1459 pour l'algorithme *A0* et par 32 pour l'algorithme *A1*. Cette augmentation est causée par les raisons énoncées au premier cas.

Le tableau 5.13 regroupe la moyenne des temps de résolution, l'écart-type ainsi que le revenu moyen pour les tests effectués sur les trois catégories de clients, pour les deux algorithmes.

TABLEAU 5.13 Paramètres des clients - Variation des catégories de clients

Alg. A0	Équiréparation	maj. de C1	maj. de C2	maj. de C3
Temps moyen (s)	12,71	318,21	60,61	18554,49
Écart-type (s)	9,36	422,32	86,36	54717,71
Variation du revenu (%)	0	+9,85	-33,19	+3,25

Alg. A1	Équiréparation	maj. de C1	maj. de C2	maj. de C3
Temps moyen (s)	5,82	113,58	10,86	185,74
Écart-type (s)	4,26	178,82	16,13	211,17
Variation du revenu (%)	0	+9,85	-33,19	+3,25

En conclusion, nous avons constaté que les paramètres concernant les clients sont très importants. En effet si le nombre de clients augmente fortement, compte tenu de la capacité existante, le temps de résolution peut augmenter de manière conséquente. L'audimat requis par client demeure également très important, car plus il est élevé, plus il faut de combinaisons de patrons différentes pour le combler, ce qui affecte grandement les temps de résolution. Enfin les temps de résolution dépendent fortement des proportions de clients dans leur catégorie.

5.3.3 Les paramètres associés aux patrons

Nous considérons ici les paramètres associés aux patrons.

Le nombre de patrons P^1 et P^2

Nous avons fait varier le nombre de patrons que proposent la chaîne leader et la concurrence. Dans un premier temps nous avons augmenté le nombre de patrons que propose la chaîne leader ; nous sommes passés de 20 à 30 patrons. Le tableau 5.14 indique que le revenu moyen a augmenté de 10,76%, ce qui peut paraître logique : en effet on met à disposition plus de patrons pour les clients, qui découpent encore plus l'inventaire, donc on peut mieux utiliser l'inventaire. Les temps de résolution

TABLEAU 5.14 **Paramètres des patrons** - Variation du nombre de patrons P^1 et P^2

Algorithme A0	$P^1 = 20, P^2 = 30$	$P^1 = 30, P^2 = 30$	$P^1 = 20, P^2 = 45$
Temps moyen (s)	12,71	23,70	556,42
Écart-type (s)	9,36	24,54	1177,61
Variation du revenu (%)	0	+10,76	-4,11

Algorithme A1	$P^1 = 20, P^2 = 30$	$P^1 = 30, P^2 = 30$	$P^1 = 20, P^2 = 45$
Temps moyen (s)	5,83	9,50	103,32
Écart-type (s)	4,26	8,36	207,13
Variation du revenu (%)	0	+10,76	-4,11

sont affectés par cette augmentation (tableau 5.14) : ils sont presque doublés pour l'algorithme A0 et augmentent de plus de 50% pour l'algorithme A1.

Nous avons également augmenté le nombre de patrons proposés par la concurrence : nous sommes passés de 30 à 45 patrons. Le revenu moyen a diminué de 4,11% (tableau 5.14), ce qui est négligeable, mais qui peut néanmoins être expliqué : la concurrence propose plus de patrons, donc un client a plus de chance de trouver un patron qui lui convient. Les temps de résolution augmentent de manière considérable : la moyenne pour l'algorithme A0 est multipliée par 40, et par 17 pour l'algorithme A1 (voir tableau 5.14). En effet on augmente le nombre de variables binaires $y_{c,p}^2$: on passe de 270 à 405 variables. Le problème de second niveau influence alors beaucoup les temps de résolution, car ces variables n'interviennent que dans ce cas-là.

Variation du nombre de patrons qui intéressent les clients P_{ic}^*

Nous avons dans un premier temps fait varier le nombre de patrons entre 2 et 4. Nous pouvons voir dans le tableau 5.15 que le revenu a alors diminué de 12,26% : effectivement les patrons sont attribués aléatoirement. Donc, si on en attribue moins à chaque client, il y a plus de chances d'obtenir des patrons moins intéressants que les

TABLEAU 5.15 Paramètres des patrons - Variation de P_c^*

Algorithme A0	$2 \leq P_{ic}^* \leq 4$	$2 \leq P_{ic}^* \leq 5$	$3 \leq P_{ic}^* \leq 6$
Temps moyen (s)	6,71	12,71	26,57
Écart-type (s)	3,24	9,36	43,88
Variation du revenu (%)	-12,26	0	+6,36

Algorithme A1	$2 \leq P_{ic}^* \leq 4$	$2 \leq P_{ic}^* \leq 5$	$3 \leq P_{ic}^* \leq 6$
Temps moyen (s)	3,97	5,83	10,71
Écart-type (s)	2,42	4,66	13,04
Variation du revenu (%)	-12,26	0	+6,36

patrons proposés par la concurrence. Les temps de résolution diminuent, car moins de variables binaires $y_{c,p}^1$ ($T_{c,p}^1$ également) sont générées. Nous en avons entre 18 et 45 dans le jeu initial, et ici nous en avons entre 18 et 36. Même si la réduction est relativement faible, cela influe sur le temps de résolution car ces variables sont présentes dans toutes les équations, que ce soit au premier ou au second niveau (voir tableau 5.15).

Quand nous augmentons le nombre de patrons qui intéressent les clients, le revenu moyen n'augmente que de 6,36% (tableau 5.15). Les temps de résolution augmentent également, et sont quasiment doublés. En effet, si chaque client est intéressé par entre 3 et 6 patrons, nous obtenons alors entre 27 et 54 variables binaires, et le raisonnement précédent s'applique dans le sens inverse.

En conclusion, les paramètres mettant en jeu les patrons sont très importants. En effet, un changement quelconque de l'un de ces paramètres peut influencer grandement le temps de résolution.

5.3.4 Conclusion sur le plan d'expérimentation

Nous avons rassemblé l'ensemble des paramètres que nous avons fait varier lors de ce plan d'expérimentation, et les avons regroupées au tableau 5.16. Les différents groupes obtenus s'appuient sur les temps de résolution acquis lors de l'ensemble des tests numériques effectués précédemment. Ces groupes d'instances sont classés par difficulté croissante de résolution ; les instances de type A sont très faciles à résoudre et celles de type F sont très difficiles à résoudre. Remarquons que les instances de type C définissent le jeu de données initial.

Le tableau 5.16 présente ainsi les moyennes des temps pour dix instances de chaque classe d'instances (A à F). Les instances que l'on qualifie de "faciles" ont des temps inférieurs à une seconde. Les instances que l'on qualifie de "moyennes" ont des temps tournant autour de dix secondes. Cependant, les instances "difficiles" ont des temps avoisinant les 1000 secondes. Les temps des instances de type F sont très élevés à cause d'une instance "rebelle" dont la résolution a duré quelques jours.

5.4 Les pertes de revenu occasionnées par certaines hypothèses du modèle

Dans cette section, nous vérifions si le revenu de la chaîne leader diminue à cause de certaines hypothèses du modèle, notamment celle concernant le processus d'achat/vente (Rappel : seulement une petite proportion de patrons intéressent chaque client). Pour ces tests, nous avons équiréparti le nombre de clients dans leurs catégories. Les instances testées sont des petites instances : en effet le fait de générer toutes les variables $y_{c,p}^1$ influe sur les temps de résolution.

TABLEAU 5.16 Résultats du plan d'expérimentation

Instances		Instances Faciles		Instances Moyennes		Instances difficiles	
		A	B	C	D	E	F
Param. Leader	$l(s)$	207,5	177,5	177,5	127,5	127,5	127,5
	U^1	15	15	10	10	10	10
	$I_{s,l}$	3-6	3-6	2-5	2-5	1-4	1-4
	L	2	3	3	4	4	4
Param. Clients	C	6	6	9	9	9	12
	A_c	1*	1*	2*	2*	3*	3*
	C_i	Equi	Equi	Equi	Equi	Equi	Equi
Param. Patrons	P^1	20	20	20	20	30	30
	P^2	30	30	30	30	45	45
	P_c^*	1-4	1-4	2-5	2-5	3-6	3-6
Tps. Résol. (s)	A0	1,03	0,86	14,89	34,81	2132,98	29709,80
Tps. Résol. (s)	A1	0,60	0,52	7,99	6,69	1028,14	17049,78

Pour une instance avec :

- 3 longueurs de spot ;
- 6 pauses chez le leader, 10 pour la concurrence ;
- 6 clients ;
- 16 patrons vendus par le leader, 27 par la concurrence ;

nous obtenons les résultats du tableau 5.17, soit une perte moyenne de 22,69%.

Pour une autre instance définie par :

- 4 longueurs de spot ;
- 4 pauses chez le leader, 4 chez la concurrence ;
- 6 clients ;
- 22 patrons vendus par le leader, 33 par la concurrence ;

nous obtenons les résultats dans le tableau 5.18, soit une perte moyenne de 17,98%.

En conclusion, la perte de revenu n'est pas négligeable, car elle atteint en moyenne environ 20%. Cela semble naturel, car le client a un choix restreint de patrons.

TABLEAU 5.17 **Perte de revenu** sur des très petites instances

Instance	Revenu Sans Choix (\$)	Revenu Avec Choix (\$)	Perte/Gain (%)
1	670	599	-10,60
2	736	1017	-38,18
3	1738	1322	-23,94
4	505	337	-33,27
5	1184	740	-37,50
6	1485	878	-40,88
7	1311	844	-35,62
8	674	560	-16,91
9	1468	972	-33,33
10	636	599	-5,82

TABLEAU 5.18 **Perte de revenu** sur des petites instances

Instance	Revenu Sans Choix (\$)	Revenu Avec Choix (\$)	Perte/Gain (%)
1	1300	1083	-16,69
2	1060	888	-16,23
3	923	611	-33,80
4	1776	934	-47,41
5	2016	1562	-22,52
6	737	733	-0,54
7	1193	1071	-10,23
8	852	808	-5,16
9	1047	873	-16,62
10	1400	1621	-15,79

5.5 Simulation d'un *prime time*

Nous avons voulu conclure ce mémoire en modélisant un *prime time*, et voir les temps de résolution associés à ce moment de la journée qui consistue la majeure partie du chiffre d'affaire des chaînes de télévision. Voici les données pour ce *prime time* :

- 15 clients ;
- 3 longueurs de spots ;
- 20 pauses chez le leader ;
- Longueur des pauses comprise entre 180 et 235 secondes ;
- 30 pauses chez la concurrence ;
- 20 patrons chez le leader ;
- 30 patrons chez la concurrence ;
- Entre 2 et 5 patrons intéressent les clients ;
- $I_{s,l}^1$ est compris entre 2 et 5 ;
- Les patrons $T1$ et $T3$ contiennent des spots sur 5 pauses publicitaires.

Le tableau 5.19 nous donne les temps pour le jeu de données le plus dense de ces travaux. Les temps de résolution sont très satisfaisants, car ils dépassent rarement les 1000 secondes. Notons également que sur l'instance 2, l'algorithme $A1$ est moins performant que l'algorithme $A0$, ayant un temps 50% plus élevé. Cependant les temps obtenus sont plus qu'acceptables sur l'ensemble des autres instances.

TABLEAU 5.19 Comportement des deux algorithmes lors d'un *prime time*

Instances	Algorithme A0			Algorithme A1		
	Nombre d'itérations	Nombre de coupes	Temps (s)	Nombre d'itérations	Nombre de coupes	Temps (s)
1	6	27	27,20	2	54	4,14
2	6	41	1191,05	2	54	1609,11
3	6	39	112,43	6	67	74,23
4	7	38	126,77	4	59	22,90
5	7	46	333,12	3	56	141,23
6	4	22	6,21	3	47	8,09
7	11	42	304,34	6	61	81,37
8	8	45	1045,27	4	59	577,43
9	9	47	373,50	5	65	100,10
10	9	53	881,41	5	67	301,84

Chapitre 6

CONCLUSION

Synthèse des travaux

Ce mémoire reprend les travaux de Chan (2008), qui a été le premier à étudier le problème de la tarification optimale pour le placement média par le biais de la programmation biniveau. Le modèle qu'il a proposé ne restait efficace que pour des instances de taille modeste, et les solutions du problème étaient souvent irréalistes. Nous avons ainsi voulu aborder ce problème de gestion du revenu avec des patrons de spots publicitaires.

Nous avons repris le cadre algorithmique mis en place par ce dernier, et proposé une variante de cet algorithme pour tester notre modèle. Les résultats sont plutôt satisfaisants, car nos temps de résolution sont bien meilleurs que le modèle M-Spot, et permettent d'effectuer de façon réaliste et optimale la couverture publicitaire d'un *prime time*.

En effectuant une analyse de sensibilité sur certains paramètres du modèle, nous avons déterminé lesquels perturbaient le plus les temps de résolution, à savoir le nombre de clients, la cible d'audience des clients et le nombre de patrons.

Les résultats numériques sont d'autant plus encourageants que l'ensemble des tests ont été réalisés avec Xpress, et nous avons montré que Cplex était bien meilleur que Xpress dans l'annexe B. Si la programmation avait été réalisée avec Cplex, il y a de grandes chances que l'on aurait pu programmer des spots publicitaires sur une journée entière.

Limitations de la solution proposée

Bien qu'ayant obtenu des temps de résolution satisfaisants, nous nous devons d'ajouter un bémol à nos résultats.

Le modèle M-Patron que nous proposons ici s'appuie sur notre propre expérience de téléspectateur. Tout ce qui concerne les patrons, de leur physionomie à la segmentation de ceux-ci, pourrait s'avérer différent dans la réalité. Les patrons créés dans l'industrie des médias peuvent diverger de ceux que nous suggérons ici.

Dans le même registre, l'ensemble des données utilisées pour les résolutions numériques sont générées aléatoirement. Les temps de résolution pourraient être différents avec des données réelles.

Enfin, nous nous limitons à programmer une soirée de publicité seulement avec Xpress. Même si Cplex s'avère être meilleur que Xpress, les résolutions ne devraient pas dépasser une journée de programmation. Il aurait été intéressant de pouvoir opérer la programmation des spots publicitaires pour une semaine, et ainsi développer des patrons "transversaux", qui répètent des spots publicitaires à des moments de la journée, mais sur plusieurs jours de la semaine.

Améliorations futures

Il serait captivant de travailler avec des données réelles afin d'évaluer l'efficacité du modèle. Le *design* des patrons pourrait alors être très précis, et nous pourrions alors proposer d'autres algorithmes liés à l'architecture de ces patrons.

Les programmes mixtes en nombres entiers prennent le plus de temps dans les résolutions, soit environ 99% du temps de résolution. Il serait alors judicieux de considérer le *Branch-and-Bound* pour résoudre ces programmes, et ce à chaque itération : on pourrait alors s'intéresser aux différentes stratégies de branchement, et procéder à un raffinement de celles-ci.

Il serait également avantageux de développer des heuristiques pour pouvoir s'at-

taquer à des problèmes de grande taille, et ainsi résoudre le problème sur au moins une semaine. On pourrait éventuellement travailler sur les trois paramètres énoncés précédemment : le nombre de clients, leur audience associée et le nombre de patrons.

Références

- ANDERSON, S. et GABSZEWICS, J. (2006). *Handbooks of the Economics of Art and Culture*. Elsevier Science.
- ANJOS, M., CHENG, R. et CURRIE, C. (2005). Optimal Pricing Policies for Perishable Products. *European Journal of Operational Research*, vol. 166, pp. 246–254.
- ARAMAN, V. et POPESCU, I. (2007). Media Revenue Management Under Audience Uncertainty. Rapport technique.
- BOLLAPRAGADA, S., CHENG, H., PHILLIPS, M. et GARBIRAS, M. (2002). NBC's Optimization Systems Increase Revenues and Productivity. *Interfaces*, vol. 32, pp. 47–60.
- BOLLAPRAGADA, S. et GARBIRAS, S. (2004). Scheduling Commercial on Broadcast Television. *Operations Research*, vol. 53, pp. 337–345.
- BOLLAPRAGADA, S., GUPTA, S., HURWITZ, B., MILES, P. et TYAGI, R. (2008). NBC-Universal Uses a Novel Qualitative Forecasting Technique to Predict Advertising Demand. *Interfaces*, vol. 38, pp. 103–111.
- BROTCORNE, L., HANAFAI, S. et MANSI, R. (2009). A Dynamic Programming Algorithm for the Bilevel Knapsack Problem. *Operations Research Letters*, vol. 37, pp. 215–218.
- BROWN, K. et ALEXANDER, P. (2005). Market Structure, Viewer Welfare, and Advertising Rates in Local Broadcast Tv Markets. *Economics Letters*, vol. 86, pp. 331–337.
- CALVETE, H., GALÉ, C. et MATEO, P. (2008). A New Approach for Solving Linear Bilevel Problems Using Genetic Algorithms. *European Journal of Operational Research*, vol. 188, pp. 14–28.

- CHAN, N. (2008). *Tarifcation Optimale pour le Placement Média*. Mémoire de maîtrise, École Polytechnique de Montréal.
- CHAN, N., SAVARD, G. et MARCOTTE, P. (2010). A Stackelberg Model for Optimal Media Placement in the Broadcasting Industry. *Operations Research*.
- CHIANG, C., CHEN, J. et XU, X. (2007). An Overview of Research on Revenue Management : Current Issues and Future Research. *International Journal of Revenue Management*, vol. 1.
- COLSON, B., MARCOTTE, P. et SAVARD, G. (2005). Bilevel Programming : A Survey. *4OR*, vol. 3, pp. 87–107.
- CÔTÉ, J., SAVARD, G. et MARCOTTE, P. (2007). A bilevel modeling approach to pricing and fare optimization in the airline industry.
- DEMPE, S. (2002). *Foundations of Bilevel Programming*. Springer US.
- DEMPE, S. (2003). Annotated Bibliography on Bilevel Programming and Mathematical Programs with Equilibrium Constraints. *Optimization : A journal of Mathematical Programming and Operations Research*, vol. 52, pp. 333–359.
- DONAGHY, K., MCMAHON, U. et MCDOWELL, D. (1995). Yield Management : an overview. *International Journal of Hospitality Management*, vol. 14, pp. 139–150.
- DUSSAULT, J.-P., MARCOTTE, P., ROCH, S. et SAVARD, G. (2006). A Smoothing Heuristic for a Bilevel Pricing Problem. *European Journal of Operational Research*, vol. 174, pp. 1396–1413.
- GENDREAU, M., MARCOTTE, P. et SAVARD, G. (1996). A Hybrid Tabu-Ascent Algorithm for the Linear Bilevel Programming Problem. *Journal of Global Optimization*, vol. 8, pp. 217–233.
- HANSEN, P., JAUMARD, B. et SAVARD, G. (1992). New Branch and Bound Rules for Linear Bilevel Programming. *SIAM Journal on Scientific and Statistical Computing*, vol. 13, pp. 1194–1217.

- HEO, C. Y. et LEE, S. (2009). Application of Revenue Management Practice to the Theme Park Industry. *International Journal of Hospitality Management*, vol. 29, pp. 446–453.
- KIMMS, A. et MÜLLER-BUNGART, M. (2007). Revenue Management for Broadcasting Commercials : The Channel's problem of Selecting and Scheduling the Advertisements to Be Aired. *International Journal of Revenue Management*, vol. 1.
- MACMILLAN, D. et LEHMAN, P. (2008). The Super Bowl Economy. *Business Week*.
- MANGÀNI, A. (2003). Profit and Audience Maximization in Broadcasting Markets. *Information Economics and Politics*, vol. 15, pp. 305–315.
- MARCOTTE, P. et SAVARD, G. (2002). A Bilevel Programming Approach to Optimal Price Setting. *Decision and Control in Management Science - Essays in Honor of Alain Haurie*, pp. 97–117.
- MIHIOTIS, A. et TSAKIRIS, I. (2004). A Mathematical Programming Study of Advertising Allocation Problem. *Applied Mathematics and Computation*, vol. 148, pp. 373–379.
- PEITZ, M. et VALLETTI, T. (2008). Content and Advertising in the Media : Pay-Tv Versus Free-to-Air. *International Journal of Industrial Organization*, vol. 26, pp. 949–965.
- PÉREZ-GLADISH, B., GONZALEZ, I., BILBAO-TEROL, A. et ARENAS-PARRA, M. (2010). Planning a TV Advertising Campaign : A Crisp Multiobjective Programming Model From Fuzzy Basic Datas. *Omega - The International Journal of Management Science*, vol. 38, pp. 84–94.
- SULIMAN, S. (2001). Pattern Generating Procedure for the Cutting Stock Problem. *International Journal of Production Economics*, vol. 74, pp. 293–301.
- TALLURI, K. T. et VAN RYZIN, G. J. (2004). *The Theory and Practice of Revenue Management*. Springer Science + Business Media, Inc.

VICENTE, L., SAVARD, G. et JUDICE, J. (1996). Discrete Linear Bilevel Programming Problem. *Journal of Optimization Theory and Applications*, vol. 89, pp. 597–614.

ZHANG, X. (2006). Mathematical Models for the Television Prime Time Advertising Allocation Problem. *International Journal of Operational Research*, no. 1, pp. 302–322.

Annexe A

Exemple numérique pour le M-Spot

Dans leur exemple numérique, Chan (2008) a considéré :

C : 2 clients, voulant respectivement un audimat de 50 et 55 ;

D : 1 une catégorie démographique ;

L : 2 longueurs de spot publicitaire, 15 et 30 secondes ;

U^1 : 1 pause publicitaire de longueur 75 secondes pour le leader ;

U^2 : 1 pause publicitaire pour la concurrence ;

P^1 : 4 spots proposés par le leader ;

P^2 : 4 spots proposés par la concurrence.

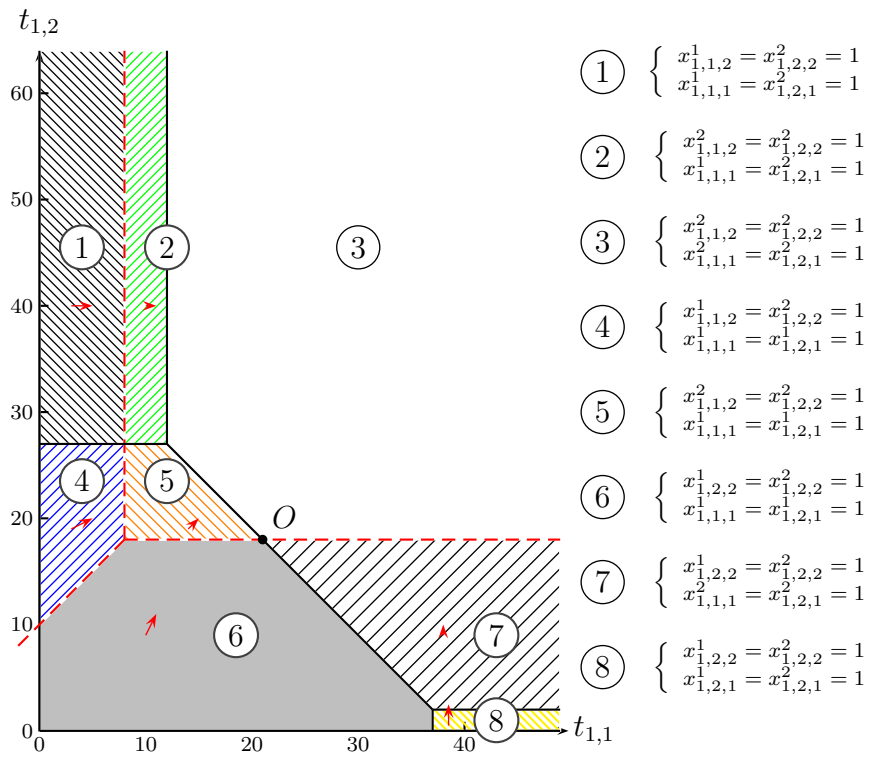


FIGURE A.1 Ensemble des polyèdres d'optimalité pour le M-Spot

Annexe B

Temps de résolution avec Cplex

Les problèmes ont été résolus sous Xpress-Mosel v2.2.0. Nous avons voulu les résoudre sous Cplex pour vérifier si les temps de résolution demeuraient les mêmes. La version de Cplex utilisée pour mener cette étude comparative est Cplex 10.1.1.

Dans notre algorithme, les problèmes relaxés (étape 1 de l'algorithme 1 et étape 2 de l'algorithme 2) occupent 95% du temps de résolution total. Les autres étapes (problème de second niveau, ajout de coupes) prennent quelques secondes. Nous avons ainsi comparé les temps de résolution des problèmes relaxés obtenus au fur et à mesure des itérations.

Nous avons testé les deux solveurs sur des jeux de données très difficiles à résoudre :

- 15 clients ;
- 4 longueurs de spots ;
- 20 pauses chez le leader ;
- Longueur des pauses comprise entre 180 et 235 secondes ;
- 30 pauses chez la concurrence ;
- 20 patrons chez le leader ;
- 30 patrons chez la concurrence ;
- Entre 2 et 5 patrons intéressent les clients ;
- $I_{s,l}^1$ est compris entre 2 et 5 ;
- Les patrons $T1$ et $T3$ contiennent des spots sur 5 pauses publicitaires.

TABLEAU B.1 4 Instances difficiles résolues avec **Cplex** et **Xpress**

Instance	Itération	Temps Xpress (s)	Temps Cplex (s)
3	1	12418,50	31,15
	2	349,60	15,84
	3	205,94	33,16
4	1	201,54	61,19
	2	8706,46	279,05
	3	10909,60	47,48
	4	85336,90	77,37
	5	4639,46	90,65
8	1	62,74	269,57
	2	4764,59	736,24
	3	48002,70	235,28
	4	11757,00	549,12
	5	5456,98	539,75
9	1	107,69	19,76
	2	554,52	19,97
	3	11093,60	18,52
	4	648,82	28,10
	5	40817,20	30,77
	6	8798,85	27,32

Le tableau B.1 regroupe les temps de résolution pour les quatre instances les plus difficiles à résoudre (dix instances ont été générées à partir du jeu de données précédent). Les temps de résolution renvoyés par Cplex sont nettement meilleurs, étant jusqu'à 430 fois plus rapide !

Dans le tableau B.2, les temps de chaque instance sont données. Les temps de résolution de Cplex sont meilleurs que ceux retournés par Xpress sur chaque instance. Les temps sont deux fois plus rapides dans le pire des cas, et 430 fois plus rapides dans le meilleur des cas ! En moyenne, Cplex est 50 fois plus rapide sur le jeu de données testé dans cette annexe (tableau B.3).

TABLEAU B.2 Temps de résolution des 10 instances avec **Cplex** et **Xpress**

Instance	Temps Xpress (s)	Temps Cplex (s)
1	5097,86	1519,77
2	577,74	12,16
3	12974,04	80,15
4	109793,96	555,74
5	5304,72	217,15
6	4826,23	183,43
7	129,59	54,17
8	70044,01	2329,96
9	62020,69	144,44
10	2125,29	154,50

TABLEAU B.3 Moyenne des temps et écarts-type avec **Cplex** et **Xpress**

Solveur	Moyenne des Temps (s)	Écart-type (s)
XPRESS	27289,40	36892,99
CPLEX	525,15	736,51