Titre: Title:	A Mathematical Programming Framework for Network Capacity Control in Customer Choice-Based Revenue Management
Auteur: Author:	Morad Hosseinalifam
Date:	2014
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Hosseinalifam, M. (2014). A Mathematical Programming Framework for Network Capacity Control in Customer Choice-Based Revenue Management [Ph.D. thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/1525/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/1525/
Directeurs de recherche: Advisors:	Gilles Savard, & Patrice Marcotte
Programme: Program:	Génie industriel

UNIVERSITÉ DE MONTRÉAL

A MATHEMATICAL PROGRAMMING FRAMEWORK FOR NETWORK CAPACITY CONTROL IN CUSTOMER CHOICE-BASED REVENUE MANAGEMENT

MORAD HOSSEINALIFAM
DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE INDUSTRIEL)
AOÛT 2014

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

A MATHEMATICAL PROGRAMMING FRAMEWORK FOR NETWORK CAPACITY CONTROL IN CUSTOMER CHOICE-BASED REVENUE MANAGEMENT

présentée par : HOSSEINALIFAM Morad

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

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

M. ANJOS Miguel F., Ph.D., président

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

M. MARCOTTE Patrice, Ph.D., membre et codirecteur de recherche

M. BASTIN Fabian, Ph.D., membre

M. KAZEMI ZANJANI Masoumeh, Ph.D., membre

DEDICATION

To my dear family

ACKNOWLEDGMENT

Enough words have been exchanged ;
Now at last let me see some deeds !
While you turn compliments,
Something useful should transpire.
What use is it to speak of inspiration ?
To the hesitant it never appears.
If you would be a poet,
Then take command of poetry.
You know what we require,
We want to down strong brew ;
So get on with it !
What does not happen today, will not be done tomorrow,
And you should not let a day slip by,
Let resolution grasp what's possible
and seize it boldly by the hair ;
it will not get away
and it labors on, because it must.

– Johann Wolfgang von Goethe

First of all, I would like to express my deepest and sincerest gratitude to my supervisors, professor Gilles Savard and professor Patrice Marcotte for accepting me in their research group, their great knowledge, their confidence in my abilities and their encouragements which made difficulties much easier during my research.

I would like to thank my industrial advisors, Dr. Louis-Philippe Bigras and Dr. Fabien Cirinei for their support and guidance during my research and I wish to express my appreciation to all my dear colleagues at ExPretio Technologies.

I am very thankful to my dear friends and colleagues at GERAD : Shadi, Yousef, Ahad, Mohsen, Hamed, Atousa and Thibault who made this journey as enjoyable as possible.

I wish to extend my deepest appreciation and gratitude to my friends in Savalan, my second family in Montreal : Neda, Shahrouz, Leila, Ali and all my friends in Savalan, which made unforgettable memories and joys during the last years.

Finally, I would like to thank my parents, my dear brother and sister for their encouragements and supports. This work was not possible without their presence. They are the driving force behind all of my successes. To them I tribute a fervent thanks.

RÉSUMÉ

Cette thèse est basée sur l'étude de différentes approches pour répondre à la problématique du contrôle de capacité pour les réseaux en gestion du revenu. Elle est composée de cinq chapitres. Le premier donne une vue d'ensemble de la thèse ainsi que la méthodologie suivie pour analyser chaque approche. Les trois chapitres suivants sont à mettre en lien avec des articles que nous avons soumis dans des revues internationales. Ils proposent de nouveaux modèles et algorithmes pour le contrôle de capacité en gestion du revenu. Les cinquième et sixième chapitres contiennent la conclusion et l'ouverture de la thèse. Nous décrivons, dans la suite, chaque chapitre plus précisément.

Dans le chapitre deux, nous proposons une approche de programmation mathématique avec choix de clients afin d'estimer les bid prices variant dans le temps. Notre méthode permet de prendre facilement en compte les contraintes techniques et pratiques d'un système de réservation central contrairement aux solutions actuelles proposées dans la littérature. En plus d'avoir développé un filtre vérifiant la disponibilité de combinaisons de produits sous un contrôle par bid price, nous avons mis au point un algorithme de génération de colonnes où une puissante heuristique est utilisée pour résoudre le sous-Problème fractionnel qui est NP-difficile. Encore une fois nos résultats numériques sur des données simulées montrent que notre solution est meilleure que les approches actuelles.

Dans le chapitre trois, nous développons une nouvelle méthode de programmation mathématique pour obtenir une allocation optimale des ressources avec un modèle de demande à choix non paramétriques. Notre méthode est alors complétement flexible et ne souffre pas des inefficacités des modèles paramétriques actuels comme ceux de type multinomial logit. Pour cela, nous avons modifié un algorithme de génération de colonnes afin de traiter efficacement des problèmes réels de grande taille. Nos résultats numériques montrent que notre méthode est meilleure que les méthodes de la littérature actuelle à la fois en qualité de la solution qu'en temps de résolution.

Dans le chapitre quatre, nous analysons un nouveau programme mathématique avec choix de clients pour estimer des booking limits qui doivent respecter une hiérarchie (nesting) ainsi que des règles commerciales imposées par le système de réservation central. De la même manière qu'au chapitre précédent, nous identifions les combinaisons de produits respectant ou non la hiérarchie (nesting) fixée par la politique de contrôle et nous développons une heuristique basée sur la décomposition. En simulant le processus stochastique d'arrivée, nous montrons encore une fois l'efficacité de notre méthode pour résoudre des problèmes complexes.

ABSTRACT

This dissertation, composed of five chapters, studies several policies concerned with the issue of capacity control in network revenue management. The first chapter provides an overview of the thesis, together with the general methodology used to analyze the control policies. In the next three chapters, each of which corresponding to a paper submitted to an international journal, we propose new models and algorithms for addressing network revenue management. The fifth and final chapter concludes the dissertation, opening avenues for further investigation. We now describe the content of each article in more detail.

In Chapter 2, we propose a customer choice-based mathematical programming approach to estimate time-dependent bid prices. In contrast with most approaches in the literature, ours can easily accommodate technical and practical constraints imposed by central reservation systems. Besides developing a filter that checks the compatibility of feasible product combinations under bid price control, we develop a column generation algorithm where a powerful heuristic is used to solve the NP-hard fractional subproblem. Again, our computational results show, based on simulated data, that the new approach outperforms alternative approaches.

In Chapter 3, we develop a new mathematical programming framework to derive optimal an optimal allocation of resources under a non-parametric choice model of demand. The implemented model is completely flexible and removes the inefficiencies of current parametric models, such as those of the ubiquitous multinomial logit. We develop for its solution a modified column generation algorithm that can efficiently address large scale, real world problems. Our computational results show that the new approach outperforms alternative approaches from the current literature, both in the terms of the quality of the solution and the required processing time.

In Chapter 4, we analyze a novel customer choice-based mathematical program to estimate

booking limits that are required to be nested, while simultaneously satisfying the business rules imposed by most central reservation system. Similar to what was accomplished in the previous chapter, we identify product combinations that are compatible (or not) with some nested control policy, and develop a decomposition-based heuristic algorithm. By simulating the stochastic arrival process, we again illustrate the efficiency of the method to tackle complex problems.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The origins of revenue management go back to the the late 1960's, when airlines started to differentiate their products and offer lower fares. At that time, the aim of the firms was to maximize revenue by discounting otherwise unsold seats while guaranteeing that high-paying customers would still purchase first-class tickets.

To do so, they imposed carefully designed fences (booking limits) between fares in order to protect high-fare seats. Currently, this revenue management practice has spread and been successfully applied to industries as diverse as rail transportation, cargo, telecoms, car rental, tourism, entertainment, hospitality, to name only a few. In this thesis, while we use the terminology of airlines, our analysis can be easily extended to the other revenue management aspects (Talluri and Van Ryzin (2005)).

In the literature, researchers decompose the revenue management problem into four sub-problems : demand forecasting, overbooking policy determination, capacity allocation (sometimes called seat inventory control) and pricing. These four related topics are the links, put together, make the revenue optimization policy of the airline (Talluri and Van Ryzin (2005)).

Although demand forecasting is a basically statistical task, the determination of overbooking and capacity allocation policies are mainly confronted by optimization techniques. Indeed, these issues have been investigated by many operations researchers over the last forty years. Major airlines have now developed and utilize computerized applications to address the overbooking and seat control problems.

Seat inventory or capacity control in revenue management is a mechanism that only accepts requests with the highest returns. In other words, it is a decision-making system that accepts or reject arrivals in order to maximize expected total revenue in a context where resource availability is limited. The main difficulty associated with capacity control in the airline industry arises when flights involve several legs.

Nowadays, firms use a variety of techniques to dynamically control perishable inventories, with the aim to maximize revenue. We divide existing approaches to control the availability of resources over a booking horizon into two main categories : bid price and nested booking. In this thesis, we study these booking control policies and we develop efficient approaches for their optimization.

Bid price policies set a threshold price for each leg (resource) and a request is accepted only if its revenue exceeds the sum of bid prices of its constituent legs. Even though this policy does not guarantee optimality, it is easy to implement and has an excellent revenue performance (Chaneton and Vulcano (2011b)).

Alternatively, a booking limit for a specific control class on a particular resource is the maximum number of units which can to be sold from that control class using that resource. The main idea in assigning booking limits is to restrict using capacity of the resources by the lower fare control classes and avoid rejecting future high willing-to-pay customers (van Ryzin and Vulcano (2008a)).

This thesis is concerned with the computation of optimal revenue management policies based on bid prices or booking limits. It is organized as follows. Chapter 2 presents a mathematical programming framework for computing improved bid prices. Our main contributions in this article are as follows :

– We develop a joint seat allocation and bid pricing model to obtain directly the value of bid prices and the corresponding allocation of resources.
– We develop two filtering approaches to decrease the size of the problem and solve the model more efficiently. These techniques can be embedded within any other capacity control policy and allow to significantly reduce the size of the problem
– Bid prices are naturally time-dependent and can be computed by dynamic programming techniques. In the spirit of (Chaneton et al. (2010))), we consider a column generation framework where the NP-Hard subproblem is solved by a new and efficient heuristic.

Chapter 3 presents a new mathematical programming formulation for network revenue management, under a non-parametric choice model. The main contributions of this paper are :

– The formulation of revenue management program where customer demand is based on a nonparametric a choice model based on ordered preference lists.
– The design and implementation of a column generation algorithms where the subproblem is solved by an algorithm that exploits the problem's structure.
– An aggregation of the ordered preference list that allow a significant reduction of the size of the problem.

Chapter 4 presents a new mathematical programming approach for computing optimal booking limits in the network revenue management problem. The main contributions of this paper are :

– We develop a mathematical programming formulation to compute optimal booking limits under a non-parametric choice model of demand that comply with the "nestedness" property implemented by most airlines.
– An important feature of the model is its compliance with rules implemented by most computer reservation systems.
– Besides the estimation of nested booking limits, the proposed approach provides the corresponding offer sets. These data provide vital information to the analysts who manage the revenue management system.
– We develop a fast decomposition-based heuristic approach to solve the large-scale problem, whose performance is assessed on realistic instances.

Finally, in Chapter 5, we present conclusions and discuss possible future work.

CHAPTER 2

ARTICLE 1 : A NEW AND EFFICIENT BID PRICE APPROACH FOR DYNAMIC RESOURCE ALLOCATION IN THE NETWORK REVENUE MANAGEMENT PROBLEM

Chapter Information : An article based on this chapter is submitted for publication M. Hosseinalifam, P. Marcotte, and G. Savard.

In this paper, we develop a joint seat allocation and bid pricing model that derives the value of time-dependent bid prices and the corresponding resource allocation in the customer choice-based revenue management framework.

ABSTRACT

Nowadays, firms that sell perishable products use a variety of techniques to maximize revenue, including the dynamic control of their inventories. One of the most powerful and simple approaches to address this issue consists in assigning threshold values ("bid prices") to each resource, and to accept requests whenever their revenue exceeds the sum of the bid prices associated with its constituent resources. In this paper, we propose a new customer choice-based mathematical program to estimate time-dependent bid prices. In contrast with most approaches from the current literature, ours is characterized by its flexibility. Indeed, it can easily embed technical and practical constraints that are observed in most central reservation systems (CRS). To solve the model, we develop a column generation algorithm where the NP-hard subproblem is addressed via an efficient heuristic procedure. Our computational results show that the new approach outperforms alternative proposals.

Key words : bid price, customer choice behavior, network capacity control, revenue management

2.1 Introduction

Capacity control is one of the key issues in network Revenue Management (RM). It involves the design of rules that specify whether specific requests for products should be accepted or not, taking into account the fact that products use resources, and that resources are in limited supply. The ultimate aim is to maximize revenue by controlling resource availability over a booking horizon. This can be achieved by a bid price control that sets a threshold price for each resource, and where an arriving request for a specific product is accepted only if the product is made available and its revenue exceeds the sum of the bid prices of its constituent resources (Talluri and Van Ryzin (2005)).

Over the years, many approaches to the optimal allocation of resources over a finite horizon have been proposed. Recently, some have taken into account the choice behavior of rational customers, and led to the deterministic linear programming formulation CDLP (Bront et al. (2009)). However, due in part to the computational complexity of solving CDLPs of practical sizes, most Central Reservation Systems (CRS in short) implement bid prices or set booking limits[1] on the number of products that can be accessed (Meissner and Strauss (2012)).

In the airline industry, bid price controls have become the method of choice for seat inventory control problem for other reasons as well. First, even in a real-world network setting involving a large number of products and resources, a single value (bid price) is assigned to each resource at each booking period. Since the number of resources is generally much less than the number of products, the number of decision parameters is relatively small. Next, the decision-making process can be implemented quickly and very simply. Indeed, whenever a request arrives, one only needs to compare the revenue to the sum of the corresponding bid prices. Finally, the concept of bid price control is intuitive and easy to understand. Even if the approach cannot theoretically guarantee the optimal revenue, good bid prices can yet lead to a significant revenue increase. In some cases, asymptotic optimality can even be proved

1. In the airline industry, this refers to policies that set bounds on the various fare products. Bid price policies can be interpreted as 'dual' methods that achieve a similar goal.

under weak assumptions (Talluri and Van Ryzin (2005)).

Historically, bid prices were introduced by Simpson (1989) and Williamson (1992), who considered several approximating models for their computation. By interpreting the bid price of a resource as the opportunity cost of one additional capacity unit, they proposed to set bid prices to the dual variables of a suitable linear program's capacity constraints.

Talluri and van Ryzin (1998) provided the theoretical foundations for the bid price approach. In particular, they extended the concept by specifying bid prices for each resource, each time period, each capacity, and provided a two-period counter example that showed that bid prices do not necessarily yield an optimal control. More recently, Topaloglu (2009) showed how to compute bid prices that depend on residual resource capacities, through the Lagrangian relaxation of certain capacity constraints.

In the context of choice based network revenue management, Chaneton and Vulcano (2011b) proposed a bid price control policy for addressing a continuous capacity/demand model. The model allows a simple calculation of the revenue function's sample path gradient, which is then embedded within a stochastic steepest ascent algorithm that converges towards a stationary point of the revenue function. In the static case, Chaneton et al. (2010) developed a framework for solving the CDLP linear program, by focusing on offer sets that are compatible with some bid price control policy. As we will see later, this feature is shared by our model, where the compatibility condition explicitly enters the column generation framework. In a recent work, Meissner and Strauss (2012) proposes a heuristic that iteratively improves an initial guess of bid prices, that could be provided by a dynamic estimate of the capacities' marginal values.

This paper's main contribution is concerned with the development of a joint seat allocation and bid pricing model that derives the value of time-dependent bid prices and the corresponding resource allocation. Our approach is based on the customer choice-based deterministic linear programming paradigm. Its structure enables to take into account not only the hub-and-spoke structure of the network, but also the behavior of customers, who base their purchase decisions upon the attributes of the products offered by the firms, as well as their own willingness to pay. In order to solve practical real-world problems, in the spirit of Chaneton et al. (2010), we develop a column generation algorithm that is based on an efficient heuristic procedure for solving the NP-hard subproblem. Moreover, we introduce two filtering approaches, that are compatible with arbitrary control policies, and allow the exact solution of small instances by off-the-shelf solvers.

The rest of the paper is organized as follows. Section 2 is devoted to the formulation of the inventory management model, i.e., a bid price model for choice based network revenue management. We present algorithmic approaches to the solution of the model in Section 3. In Section 4, we provide computational results, as well as comparisons with alternative approaches from the recent literature. Finally, concluding comments and avenues for further research are outlined in Section 5.

2.2 Problem formulation

In this section, we introduce the general definitions and notation, and provide mathematical formulations of the bid price model.

2.2.1 General definitions and notation

Let us consider a set of products indexed by $j \in J = \{1, 2, \dots, |J|\}$,

together with a revenue (fare) vector r and a capacity vector c, also of dimension $|J|$. The use of resources $i \in I = \{1, 2, \dots, |I|\}$ by the products is specified by an incidence matrix A of dimension $|I| \times |J|$, whose binary elements are defined as

$$a_{ij} = \begin{cases} 1, & \text{if resource } i \text{ is used by product } j, \\ 0, & \text{otherwise.} \end{cases}$$

In the model, customers are divided into segments indexed by $l \in L = \{1, 2, \dots, |L|\}$, and characterized by attributes such as time, price, or path preferences. We associate to each segment l a consideration set Γ_l, that specifies the subset of products considered by a customer belonging to segment l. We denote by a positive number v_{lj} the value of product j to a customer belonging to segment l.

Time $t \in T = \{1, 2, \ldots, |T|\}$ runs forward in discrete increments, and the arrival process of customers is governed by independent Poisson processes of respective rates $\lambda_l = \lambda p_l$, where p_l denotes the probability that an arriving customer belongs to segment l, and $\lambda = \sum_{l=1}^{\mathcal{L}} \lambda_l$ is the total arrival rate.

Within each time period t, the firm must decide which subset $S \subseteq N$ of products is made available to customers. Accordingly, we denote by $\xi_j(S)$ a binary variable that specifies whether or not $j \in S$, by $P_j(S)$ the probability that product j be selected by an arriving customer, and by $P_0(S) = 1 - \sum_{j \in S} P_j(S)$ the residual probability associated with the no-purchase option. Assuming that customers' choice probabilities are based on a discrete choice model, the probability that customer l select an available product $j \in S$ is given by

$$\begin{aligned} P_{lj}(S) &= \frac{v_{lj}}{v_{l0} + \sum_{h \in \Gamma_l \cap S} v_{lh}} && (2.1) \\ &= \frac{\xi_j(S) v_{lj}}{v_{l0} + \sum_{h \in J} \xi_h(S) v_{lh}} && (2.2) \end{aligned}$$

This definition is compatible with the classical multinomial logit choice model, where the utilities are the sum of a linear combination of product j's attributes and a Gumbel-distributed random variable. Since the firm does not have prior knowledge of the segment associated with an arriving customer, the probability that a product j be selected is given by

$$P_j(S) = \sum_{l=1}^{L} p_l P_{lj}(S). \tag{2.3}$$

Given an offer set S, the expected revenue is expressed as

$$R(S) = \sum_{j \in S} r_j P_j(S). \tag{2.4}$$

Now, let $Q_i(S)$ denote the probability of using a unit of resource $i \in I$, regrouped into the vector $Q(S)$. Letting $P(S) = (P_1(S), \ldots, P_n(S))^\top$, we can write

$$Q(S) = AP(S). \tag{2.5}$$

Finally, a *bid price policy* associates to each resource i a *bid price* π_i. Based on the vector π, a product $j \in S$ is then *offered* at time t if and only if

$$r_j \geq \sum_{i \in I} a_{ij} \pi_{it}. \tag{2.6}$$

2.2.2 Model formulation I

For each time index $t \in T$, let us denote by ξ_{jt} the binary variable that specifies whether product j belongs to the set S at time t, and by ξ the vector obtained by concatenation of these variables. The choice of an optimal bid price policy can then be formulated as the mathematical program :

$$\text{BID-Ia :} \qquad \max_{\xi, \pi} \sum_{t \in T} \sum_{j \in J} \sum_{l \in L} \lambda p_l r_j \frac{\xi_{jt} v_{lj}}{v_{l0} + \sum_{h \in J} \xi_h v_{lh}}, \tag{2.7}$$

subject to

$$\sum_{t \in T} \sum_{j \in J} \lambda p_l a_{ij} \frac{\xi_{jt} v_{lj}}{v_{l0} + \sum_{h \in J} \xi_h v_{lh}} \leq c_i \qquad \forall i \in I, \tag{2.8}$$

$$r_j < \sum_{i \in I} a_{ij} \pi_{it} \quad \Rightarrow \quad \xi_{jt} = 0 \qquad \forall t \in T, \forall j \in J, \tag{2.9}$$

$$\xi_{jt} \in \{0, 1\} \qquad \forall t \in T, \forall j \in J. \tag{2.10}$$

Now, upon the introduction of variables

$$y_{jt} = \frac{\xi_{jt} v_{lj}}{v_{l0} + \sum_{h \in J} \xi_h v_{lh}}$$

and a big-M constant M_1, BID-Ia can be recast as the mixed integer nonlinear program

$$\text{BID-Ib}: \quad \max_{y,\pi} \sum_{t \in T} \sum_{j \in J} \sum_{l \in L} \lambda p_l r_j y_{jt}, \tag{2.11}$$

subject to

$$\sum_{t \in T} \sum_{j \in J} \lambda p_l a_{ij} y_{jt} \leq c_i \qquad \forall i \in I, \tag{2.12}$$

$$\frac{r_j - \sum_{j \in J} a_{ij} \pi_{it}}{M_1} \leq \xi_{jt} \leq 1 + \frac{r_j - \sum_{j \in J} a_{ij} \pi_{it}}{M_1} - \epsilon \tag{2.13}$$

$$\forall t \in T, \forall j \in J,$$

$$y_{jt}(v_{l0} + \sum_{h \in J} \xi_{ht} v_{lh}) = \xi_{jt} v_{lj} \tag{2.14}$$

$$\forall j \in J, \forall h \in J, \forall l \in L, \forall t \in T,$$

$$\xi_{jt} \in \{0, 1\} \qquad \forall t \in T, \forall j \in J. \tag{2.15}$$

In the above, the constant ϵ is a threshold value that settles the degeneracy issue that allows a product to be inserted into S only if revenue strictly exceeds its bid price, i.e., the sum of the bid prices associated with its constituent resources. Next, to linearize BID-Ib, it suffices to introduce the variables

$$z_{hjt} = \xi_{ht} y_{jt}$$

and, for some suitably large constant M_2, the set of constraints :

$$\begin{aligned} z_{hjt} &\leq y_{jt} \\ z_{hjt} &\geq y_{jt} + M_2(\xi_{ht} - 1) \end{aligned}$$

$$z_{hjt} \leq \xi_{ht}$$

for every triple $(j, h, t) \in J \times J \times T$.

This mixed integer linear program involves a number of binary variables that is very large, albeit polynomial in terms of products, resources and number of periods. For this reason, we opted for an alternative formulation whose structure is amenable to efficient algorithmic approaches.

2.2.3 Model formulation II

The second formulation is based on the choice based models introduced by Gallego et al. (2004) and further developed by Liu and van Ryzin (2008) and Bront et al. (2009), where the customers' choice sets may overlap. We let the binary variable $X_t(S)$ denote whether or not the set S is offered at time t, and consider the mixed integer linear program

$$\text{MIP-I}: \quad \max_X \sum_{t \in T} \sum_{S \subseteq N} \lambda R(S) X_t(S) \tag{2.16}$$

$$\text{subject to} \tag{2.17}$$

$$\sum_{t \in T} \sum_{S \subseteq N} \lambda Q_i(S) X_t(S) \leq c_i \qquad \forall i \in I, \tag{2.18}$$

$$\sum_{S \subseteq N} X_t(S) \leq 1, \qquad \forall t \in T, \tag{2.19}$$

$$X_t(S) \in \{0, 1\} \qquad \forall t \in T, \forall S \subseteq N, \tag{2.20}$$

which is closely related to the model of Liu and Van Ryzin (2008) or Bront et al. (2009), where the decision variable $X_t(S)$ represents the fraction of the total booking horizon over which the set S is offered. Note that our variant is more flexible, allowing the introduction of time-related constraints and controls. Otherwise, both formulations can be easily converted to each other (Kunnumkal and Topaloglu (2008)).

A drawback of MIP-I is that it may involve sets that are inducible by no bid price policy (Talluri and Van Ryzin (2005)). To sidestep this difficulty, we enforce compatibility between choice sets and some bid price vector π. More precisely, we restrict our attention to the offer sets that satisfy the two inequalities

$$r_j \geq \sum_i a_{ij}\pi_{it} \qquad \forall j \in S, \tag{2.21}$$

$$r_j < \sum_i a_{ij}\pi_{it} \qquad \forall j \notin S. \tag{2.22}$$

According to these constraints, a request for a product is accepted if and only if its revenue exceeds its bid price and it will be rejected if its revenue is strictly less than its bid price. For practical reasons, one replaces the second strict inequality by

$$r_j \leq \sum_i a_{ij}\pi_{it} + \epsilon \qquad \forall j \notin S, \tag{2.23}$$

where ϵ is a threshold value that removes the degeneracy issue that would arise when revenue and bid price coincide, and agrees with industry practice. We denote by Ω the set of offer sets that are bid-price compatible. This yields

$$\text{MIP-II}: \qquad \max_{X\in\Omega} \sum_{t\in T}\sum_{S\subseteq N} \lambda R(S) X_t(S) \tag{2.24}$$

$$\text{subject to} \tag{2.25}$$

$$\sum_{t\in T}\sum_{S\subseteq N} \lambda Q_i(S) X_t(S) \leq c_i \qquad \forall i \in I, \tag{2.26}$$

$$\sum_{S\subseteq N} X_t(S) \leq 1, \qquad \forall t \in T, \tag{2.27}$$

$$X_t(S) \in \{0,1\} \qquad \forall t \in T, \forall S \subseteq N, \tag{2.28}$$

In contrast with the BID-I formulation, MIP-II involves an exponential number of variables. However, as we shall see in the next section, its structure makes it amenable to efficient algorithms for its solution.

2.3 Solution approaches

The MIP-II model involves a number of variables that is exponential with respect to the number $|J|$ of products. Depending on its size, we consider two solution approaches.

2.3.1 Small and medium-sized instances

When the number of products is small, one can enumerate the sets that belong to Ω and solve MIP-II via an off-the-shelf solver. This could simply be achieved by enumerating all possible combinations of products, and dropping those for which there does not exist a consistent vector π, i.e., a vector that satisfies the system of equations Ω.

It is also possible to identify, offline, sets that cannot belong to an optimal solution, through an "efficiency" certificate that takes the form of a probability vector $\alpha(S)$.

Definition 2.3.1 *Let $\alpha(S)$ be a set of probabilities over the set 2^J. We say that $\bar{S} \in 2^J$ is inefficient if the following two inequalities hold :*

$$\begin{aligned} Q(\theta) &\geq \sum_{S \subseteq N} \alpha(S) Q(S) \\ R(\theta) &< \sum_{S \subseteq N} \alpha(S) R(S). \end{aligned}$$

According to this definition, a set $\bar{S}$ is inefficient if a randomization of sets achieves an expected revenue that is strictly greater than the revenue of $\bar{S}$, with no increase in the conditional usage of capacity $Q(\bar{S})$. It has been proved by Liu and van Ryzin (2008) that any optimal solution of MIP-I must be efficient, i.e., not inefficient.

Finding efficient sets is computationally NP-Hard, which explains why various heuristic approaches have been developed for this problem, most of them requiring the (implicit) enumeration of feasible subsets, which is computationally challenging for large scale instances. Whenever customer segments are disjoints, a simple "nesting by fare" order yields all efficient sets. However, in the case of overlapping segments, this approach is not valid anymore (Liu and van Ryzin (2008)), and one might have to check efficiency through the solution of the linear program

$$\text{ES}: \qquad \max_{\alpha} \sum_{S \subset N} R(S)\alpha(S) \tag{2.29}$$

subject to

$$\sum_{S} Q_i(S)\alpha(S) \leq Q_i(\theta) \qquad \forall i, \tag{2.30}$$

$$\sum_{S \subset N} \alpha(S) \leq 1, \tag{2.31}$$

$$\alpha(S) \geq 0 \qquad \forall S$$

that involves an exponential number of variables. Whenever the objective of ES exceeds $R(\theta)$, the corresponding set θ is inefficient and can be removed. To solve ES, one may resort to column generation or to heuristics.

Note that the operations mentioned in this section need only be performed offline, and thus might be envisioned if the system needs to be reoptimized on a frequent basis.

2.3.2 Large instances

For problems that are too large to be addressed directly, and taking into account that the ultimate goal is to derive bid prices (rather than offer sets), we propose a column generation approach for solving the linear programming relaxation of MIP-II, which involves a small number $|I|+|J|$ of constraints. At a generic iteration, we consider the reduced master problem

$$\text{RMP}: \quad \max_X \sum_{t\in T}\sum_{S\in\bar{\Omega}} \lambda R(S) X_t(S) \tag{2.32}$$

$$\text{subject to} \tag{2.33}$$

$$\sum_{t\in T}\sum_{S\in\bar{\Omega}} \lambda Q_i(S) X_t(S) \le c_i \qquad \forall i \in I, \tag{2.34}$$

$$0 \le X_t(S) \le 1 \qquad \forall t \in T, \forall S \in \bar{\Omega}. \tag{2.35}$$

Let κ_i be the dual variable (multiplier) associated with the ith capacity constraint, and σ_t the multiplier associated with the tth time constraint. A variable with largest positive reduced cost, to be inserted in the set $\bar{\Omega}$, is then obtained by solving the column generation subproblem

$$\text{SUB}: \quad \max_{S\in\Omega} \{\lambda R(S) - \lambda \kappa^\top Q(S)\} - \sum_{t\in T} \sigma_t, \tag{2.36}$$

where the last term can be removed from the maximization, since it is independent of S and therefore has no influence on the optimal solution.

Now, let y_j be a binary variable that assumes value 1 if $j \in S$, and 0 otherwise. These are in one-to-one correspondence with subsets of J and, upon introduction of a suitably large "big-M" constant M, the bid-price compatibility constraints can be written as

$$r_j + (1 - y_j)M \ge \sum_i a_{ij}\pi_i + \epsilon \qquad \forall j \in J, \tag{2.37}$$

$$\sum_i a_{ij}\pi_i \ge r_j(1 - y_j) \qquad \forall j \in J, \tag{2.38}$$

and we denote by $\bar{Y}$ the set of bid-price compatible binary vectors y.

Now, replacing $R(S)$ and $Q(S)$ by their respective expressions (2.4) and (2.5), and letting $w_j = (r_j - A_j^\top \kappa)$, the subproblem takes the form

$$\text{SUB}: \qquad \max_{y\in\bar{Y}} \sum_{l=1}^{L} \lambda_l \frac{\sum\limits_{j\in\Gamma_l} w_j v_{lj} y_j}{\sum\limits_{h\in\Gamma_l} v_{lh} y_h + v_{l0}}\,. \tag{2.39}$$

If the optimum of (2.39) exceeds $\sum_{t\in T}\sigma_t$, otherwise, the offer set corresponding to the binary vector y enters $\bar{\Omega}$, which is reoptimized on the enriched subset. Otherwise, an optimal solution to the relaxed MILP has been found and the algorithm halts. The crux of the method consists in solving the fractional problem (SUB), which has been proved to be NP-hard, even in the absence of the compatibility constraints (see Bront et al. (2009)). In the next section, we address this issue.

2.3.3 Solving the column generation subproblem

A number of algorithms have been proposed for addressing the fractional problem problem (2.39) in the absence of compatibility constraints. A review of these approaches can be found in Bront et al. (2009) and Hosseinalifam (2009). However, these cannot easily be adapted to our subproblem, especially when the number of products becomes large. In this section, our aim is to present a solution procedure that is efficient, both in terms of computational complexity and solution quality.

First, let us consider the problem of maximizing the ratio of affine functions

$$\max_x \frac{P(x)}{D(x)} = \frac{\sum_{j=1}^{m} p_j x_j}{\sum_{j=1}^{m} d_j x_j}, \tag{2.40}$$

where the denominator $D(x)$ is strictly positive. A classical algorithm, due to Dinkelbach, is based on the solution of the parametric linear program

$$\max_x P(x) - \rho D(x). \tag{2.41}$$

Let $x(\rho)$ be an optimal solution of the above program. It has been proved that $x(\rho)$ is an optimal solution of the original fractional program if and only if $P(x(\rho)) - \rho D(x(\rho)) = 0$. The quest for an optimal ρ can then be performed by binary search or other suitable updating techniques. For instance, it is natural, at iteration $k+1$, to set

$$\rho_k = \frac{P(x^k)}{D(x^k)}.$$

Almogy and Levin (1971) have proposed an extension of this idea to the sum of ratios problem

$$\max \sum_{i=1}^{n} \frac{P_i(x)}{D_i(x)}, \tag{2.42}$$

which is NP-hard (see Falk and Palocsay (1992)) specifically, their algorithm iteratively solves the parametric linear program

$$\max_x \sum_{i=1}^{n} P_i(x) - \rho_i^k D_i(x), \tag{2.43}$$

where the parameters are updated according to the formula :

$$\rho_i^k = \frac{P_i(x^k)}{D_i(x^k)}.$$

The algorithm halts when the objective is (close to) zero. Although a numerical example provided by Falk and Palocsay (1992) shows that this approach might fail to yield an optimal solution, it has been observed that the performance of this polynomial algorithm is frequently very good. For this reason, we decided to adapt it to our column generation subproblem :

Algorithm CGSUB

Step 1 (Initialization)

Set $k = 0$ and let $y^0 \in \bar{Y}$.

Let ϵ be a small positive number.

Step 2 (parameter update)

$$\rho_l^k = \frac{\sum_{j\in\Gamma_l} w_j v_{lj} y_j^k}{\sum_{i\in\Gamma_l} v_{lj} y_i^k + v_{l0}}, \quad \forall l \in L.$$

Step 3 (solution update)

$$y^{k+1} \in \arg\max_{y\in\bar{Y}} G^{k+1}(\rho) = \sum_{l\in L} \frac{\sum_{j\in\Gamma_l} w_j v_{lj} y_j - \rho_l^k(\sum_{j\in C_l} v_{lj} y_j + v_{l0})}{\sum_{j\in\Gamma_l} v_{lj} y_j^k + v_{l0}}.$$

Step 4 (stopping criterion)

If $|G^{k+1}(\rho)| \leq \epsilon$ then stop.

Step 5 (loop)

Return to Step 2.

This heuristic procedure, which iteratively solves linear programs, is very fast, and we observed that it delivered an optimal solution in most of our experiments.

2.4 Numerical results

In this section, we compare the performance of our algorithm against alternative approaches. To this aim, we consider three benchmark examples that have been widely used in the literature (e.g., see Chaneton and Vulcano (2011b)). The computational results have been carried out on a 2.13 GHz twin-core computer with 4 GB of RAM. The environment FICO Xpress-Mosel 7.2.1 has been used both to formulate and solve the master problems that arise in our column generation algorithm.

In order to validate the quality of the bid prices, we simulated 2000 streams of customer demand over the planning horizon, according to a Poisson process with rate λ. The conditional probability that an arriving customer belong to the segment l is p_l. Throughout, we assumed at most one arrival within any given time period. The arriving customer will choose from available products based on the bid price control policy, and according to the utilities of products in his consideration set. These utilities are consistent with a multinomial logit discrete choice model, and therefore involve a Gumbel-distributed random term. In the tables presenting the numerical results, revenues are averaged over the 2000 streams.

2.4.1 Parallel flight example

In this example, we consider a network with three parallel legs (morning, afternoon, evening) with initial capacities of 30, 50 and 40, respectively, together with two fare classes, high (H) and low (L), This induces a total of six products. Data relevant to products and resources' usage are displayed in the Figure (2.1) and Table (2.1) . As shown in Table (2.2), customers are divided into four overlapping segments based on their origin, end destination, and price sensitivity (time over price ratio).

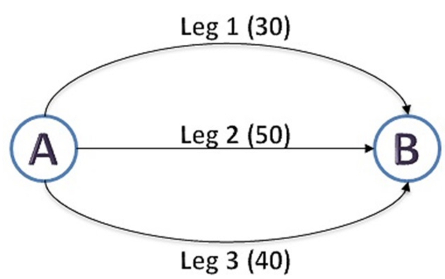

Figure 2.1 Parallel flight example : network.

Table 2.1 Parallel flight example : products

product	leg	class	fare
1	1	L	400
2	1	H	800
3	2	L	500
4	2	H	1000
5	3	L	300
6	3	H	600

The booking period is divided into $\tau = 300$ periods and an arrival rate of $\lambda = 0.5$ leads to an average of 150 arrivals. For each ot the four demand segments, three "no purchase" utilities

v_0 have been considered. Meanwhile, leg capacities have been weighted by a scale factor α in order to assess the quality of the solution with respect to various congestion levels. The total expected revenue obtained by different policies based on our bid price control policy are shown in Table 2.3.

Table 2.2 Segment definition for the parallel flight example

#	λ_l	consideration Set	preference vector	description
1	0.1	{2,4,6}	(5,10,1)	price insensitive, afternoon preference.
2	0.15	{1,3,5}	(5,1,10)	price sensitive, evening preference.
3	0.2	{1,2,3,4,5,6}	(10,8,6,4,3,1)	price sensitive, early preference.
4	0.05	{1,2,3,4,5,6}	(8,10,4,6,1,3)	price insensitive, early preference.

In this Table, column "UB" represents the upper bound obtained from the model CDLP. The next five columns represent total expected revenues obtained by different bid price control policies taken from Chaneton and Vulcano (2011b). Column "BP-SG" denotes the revenue obtained by the Stochastic Gradient method proposed by Chaneton and Vulcano (2011b). The shorthand "reopt" refers to the number of reoptimization performed over the course of the simulation, taking into account the "true" pseudo-random arrival process to adjust the amount of resources available. In the case of Deterministic Linear Programming (DLP) formulation and Choice-based Deterministic Linear programming formulation (CDLP), it has been observed that, unless a large number of reoptimizations is performed, the quality of the bid prices suffered. Note however that a large number of reoptimizations may actually induce a reduction in total revenue, as can be observed in the table. In the last column, the header "BP-CG" refers to our column generation method.

In most cases, BP-CG outperformed the alternative approaches. Actually, we could observe a pattern where BP-CG was only dominated by BP-SG, in the low congestion instances. BP-CG performed well over the range of α-values, contrary to DLP and CDLP, whose performance diminished with an increase of that parameter. Note that BP-SG, which also performed well, is handicapped by CPU times that do not scale well, making it impractical for real-life applications.

Table 2.3 Expected revenues obtained by different bid price capacity control policies on parallel flight example

#	α	v_0	UB	BP-SG 4 reopt	DLP bid-prices 10 reopt	DLP bid-prices 20 reopt	CDLP bid-prices 10 reopt	CDLP bid-prices 20 reopt	BP-CG 4 reopt
1		(1,5,5,1)	56884	**55964**	50362	50462	55178	55144	54677
2	0.6	(1,10,5,1)	56848	**55683**	50533	50596	55221	55191	54512
3		(5,20,10,5)	53819	**52029**	48788	48642	51688	50960	51451
4		(1,5,5,1)	71936	69024	63480	63265	69034	67515	**69768**
5	0.8	(1,10,5,1)	71794	69337	64142	63710	68068	66918	**69371**
6		(5,20,10,5)	61868	59132	57329	55972	58016	55826	**61724**
7		(1,5,5,1)	79155	75073	72726	71424	71861	70034	**76687**
8	1	(1,10,5,1)	76866	74064	71693	70027	70164	67515	**74916**
9		(5,20,10,5)	63255	62105	59098	58800	58160	56068	**62733**
10		(1,5,5,1)	80371	78964	74622	72604	74148	71797	**79197**
11	1.2	(1,10,5,1)	78045	74582	73208	70527	71867	69630	**76542**
12		(5,20,10,5)	63296	62118	59511	57691	59965	58110	**63108**

2.4.2 Hub and Spoke example I

The second example involves a seven-leg network inducing a total number of 22 products. The network and the product list are shown in Figure (2.2).

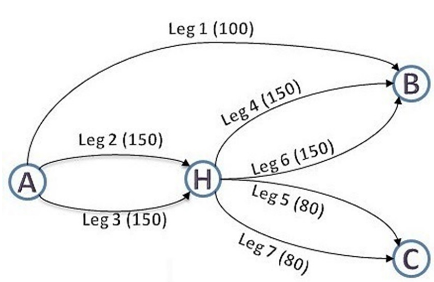

Figure 2.2 Hub and Spoke example I : network.

Due to the hub-and-spoke topology, products may use more than one resource (leg). Resources are endowed with initial capacities $(100, 150, 150, 150, 150, 80, 80)$, respectively. The booking horizon consists of $\tau = 1000$ periods. The arrival rate is set to $\lambda = 0.91$, which results in an average arrival rate of 910 customers per stream. Based on their their respective price sensitivity and origin-destination, customers belong to one of ten overlapping segments. This information is gathered in Table 2.5.

Table 2.4 Hub and Spoke example I : products

product	leg	class	fare	product	leg	class	fare
1	1	H	1000	12	1	L	500
2	2	H	400	13	2	L	200
3	3	H	400	14	3	L	200
4	4	H	300	15	4	L	150
5	5	H	300	16	5	L	150
6	6	H	500	27	6	L	250
7	7	H	500	28	7	L	250
8	{2,4}	H	600	19	{2,4}	L	300
9	{3,5}	H	600	20	{3,5}	L	300
10	{2,6}	H	700	21	{2,6}	L	350
11	{3,7}	H	700	22	{3,7}	L	350

The revenues shown in Table (2.6) illustrate the performance of BP-CG, with minor revenue losses (with respect to BP-SG) in low congestion offset by improved results as congestion increases.

2.4.3 Hub and Spoke example II

This network example is illustrated in the Figure (2.3). In this example we have five flight legs and three cities inducing the total number of sixteen products. The flight legs' initial capacities are $c := (12, 8, 8, 8, 8)$ illustrated in Table (2.7).

Customers are divided into nine overlapping segments. For all segments, three different scenarios of low, medium and high level of overlapping have been considered. These information are shown in the Table (2.8). The booking horizon is divided into $\tau = 80$ booking

Table 2.5 Segment definition for the network example I

#	λ_l	consideration set	preference vector	description
1	0.08	{1,8,9,12,19,20}	(10,8,8,6,4,4)	less price sensitive, early preference.
2	0.2	{1,8,9,12,19,20}	(1,2,2,8,10,10)	price sensitive.
3	0.05	{2,3,13,14}	(10,10,5,5)	less price senitive.
4	0.2	{2,3,13,14}	(2,2,10,10)	price sensitive.
5	0.1	{4,5,15,16}	(10,10,5,5)	less price sensitive.
6	0.15	{4,5,15,16}	(2,2,10,8)	price sensitive, slight early preference.
7	0.02	{6,7,17,18}	(10,8,5,5)	less price sensitive, slight early preference.
8	0.05	{6,7,17,18}	(2,2,10,8)	price sensitive.
9	0.02	{10,11,21,22}	(10,8,5,5)	less price sensitive, slight early preference.
10	0.04	{10,11,21,22}	(2,2,10,10)	price sensitive.

Table 2.6 Expected revenues obtained by different bid price capacity control policies on network example I

#	α	v_0	UB	BP-SG	DLP bid-prices		CDLP bid-prices		BP-CG
				4 reopt	10 reopt	20 reopt	10 reopt	20 reopt	4 reopt
1		(1,5)	215793	**212459**	172091	174913	187683	190384	210412
2	0.6	(5,10)	200515	**195037**	164570	166894	174372	175601	194033
3		(10,20)	170137	165638	154041	154547	158897	159685	**165651**
4		(1,5)	266934	238041	208185	210170	220013	224069	**262445**
5	0.8	(5,10)	223173	214847	197396	197947	201465	203810	**218998**
6		(10,20)	188574	185150	178190	178161	175395	175720	**185254**
7		(1,5)	281967	272569	237675	238923	240243	242195	**278980**
8	1	(5,10)	235284	231094	219023	219424	217070	216860	**233101**
9		(10,20)	192038	189349	186970	186800	188100	186438	**191181**

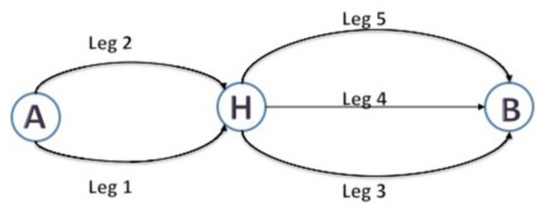

Figure 2.3 Hub and Spoke example II :network

periods and three demand scenarios have been considered, corresponding to the arrival rate λ multiplied by the scale factors $\alpha = 0.9$, 1, and 1.1, respectively.

Table 2.7 Hub and Spoke example II : products

product	leg	class	fare	product	leg	class	fare
1	1	L	400	9	5	L	400
2	1	H	800	10	5	H	800
3	2	L	300	11	1,4	L	500
4	2	H	600	12	1,4	H	1000
5	3	L	400	13	1,5	L	450
6	3	H	800	14	1,5	H	900
7	4	L	300	15	2,5	L	400
8	4	H	600	16	2,5	H	800

For this example, we compared the bid-price policies BP-CG and BP-SG to four capacity control policies that attempt to solve the exact dynamic programming formulation of the problem, without the bid-price restrictions. These policies are :

- D-CDLP : a network decomposition scheme based on the dual outcome of CDLP, and that matches the policy DCOMP analyzed in Bront et al. (2009)) ;
- TSA (time sensitive approximation) : an affine approximation of the dynamic programming value function proposed by Zhang and Adelman (2009))'
- TISA (time and inventory sensitive approximation) : a nonlinear approximation of the dynamic programming value function proposed by Meissner and Strauss (2008)) ;
- D-TSA : a network decomposition scheme based on the outcome of TSA.

Table 2.8 Segment definition for the Hub and Spoke example II

#	λ_l	consideration set	preference vector	description
1	0.07	{{{2, 4}, 1}, 3}	[[[10, 5], 10], 7]	price insensitive, early preference.
2	0.05	{{{2, 4}, 3}, 1}	[[[5, 10], 10], 7]	price insensitive, late preference.
3	0.15	{{{1, 3}}}	[[[10, 8]]]	price sensitive.
4	0.07	{{{6, 8, 10} 5} 7, 9}	[[[10, 5, 1], 10], 7, 3]	price insensitive, early preference.
5	0.05	{{{6, 8, 10} 9} 7, 5}	[[[1, 5, 10] 10], 7, 5]	price insensitive, late preference.
6	0.15	{{{5, 7, 9}}}	[[[5, 10, 5]]]	price sensitive .
7	0.07	{{{12, 14}, 11}, 13}	[[[10, 5] 10], 7]	price insensitive, early preference.
8	0.05	{{{16}, 15}, 11, 12}	[[[10], 10], 7, 5]	price insensitive, late preference.
9	0.15	{{{11, 13, 15}}}	[[[5, 8, 10]]]	price sensitive.

As expected, the revenues generated by the bid-price methods are less than those generated by the “exact” approaches, the trade-off being the very long processing times of the latter (Meissner and Strauss (2008)), in comparison with less than 5 minutes for algorithm BP-CG.

Table 2.9 Expected revenues obtained by different bid price capacity control policies on Hub and Spoke example II

#	α	overlap	D-CDLP	TSA	TISA	D-TSA	BP-SG	BP-CG
1	0.9	low	21073	21425	*21867*	21438	20792	**20895**
2	0.9	medium	20156	19190	*20793*	20324	19393	**19966**
3	0.9	high	19419	19365	*20326*	19628	19683	**19721**
4	1	low	22385	21802	*22904*	22537	21814	**21965**
5	1	medium	21544	21257	*22013*	21710	20945	**21086**
6	1	high	20495	21338	*21801*	21207	20745	**21059**
7	1.1	low	22825	23021	*23789*	23351	**22652**	22285
8	1.1	medium	22367	22490	*23116*	22633	21725	**21981**
9	1.1	high	22325	22043	*22986*	22501	**22188**	22081

2.4.4 Railroad

Our last example is based on a portion of the Thalys railroad system, which was already considered in Hosseinalifam et al. (2014). It involves five cities and four legs, and two fare classes, respectively low ‘L’ and high ‘H’ on each leg. Figure (2.4) illustrates this network and its associated 10 markets. Ten trains with a common capacity of 100 passengers satisfy the markets, which makes up a total of 200 products (see Table (2.10)).

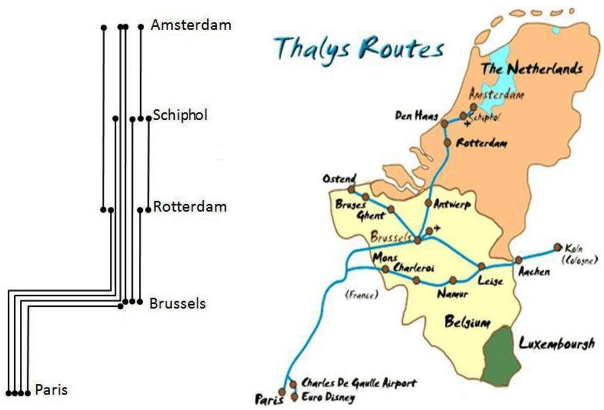

Figure 2.4 Thalys railroad system.

Table 2.10 Market fares

market	low fare	high fare
PAR → BRU	200	400
PAR → RTA	300	500
PAR → SCH	350	525
PAR → AMA	350	525
BRU → RTA	150	250
BRU → SCH	175	275
BRU → AMA	200	300
RTA → SCH	50	100
RTA → AMA	175	300
SCH → AMA	50	100

Based on their price sensitivity and origin-destination preferences, customers are divided into 20 segments detailed in Table (2.11). A comparison of different demand and capacity

Table 2.11 Segment definition for the railroad network

#	consideration set	preference vector	description
1	{1,...,20}	{10,55,25,15,6,4,3,4,5,6, 15,15,20,4,3,2,1,2,2,3,8}	PAR-BRU (L)
2	{11,...,20}	{8,70,60,10,7,4,4,4,5,40,60}	PAR-BRU (H)
3	{21,...,40}	{15,30,20,10,3,5,20,25, 10,4,4,4,8,2,1,2,2,3,3,2,2}	PAR-RTA (L)
4	{31,...,40}	{7,40,25,10,4,4,5,15,20,25,45}	PAR-RTA (H)
5	{41,...,60}	{25,25,20,4,5,5,5,6,6,10, 30,5,2,2,2,3,3,3,4,4,10}	PAR-SCH (L)
6	{51,...,60}	{7,32,21,3,3,4,5,15,15,20,30,}	PAR-SCH (H)
7	{61,...,80}	{20,20,2,5,5,6,6,7,7, 6, 8,15,3,3,4,3,3,4,4,5,4,4}	PAR-AMA (L)
8	{71,...,80}	{50,25,20,3,3,4,4,8,20,28,35}	PAR-AMA (H)
9	{81,...,100}	{10,60,50,6,4,4,5,20,22,7, 32,10,4,3,2,2,2,3,4,4,15}	BRU-RTA (L)
10	{91,...,100}	{20,90,45,5,6,2,3,4,30,60,70}	BRU-RTA (H)
11	{101,...,120}	{5,25,10,5,5,6,6,20,20, 10,8,5,4,3,3,3,4,4,5,5,5}	BRU-SCH (L)
12	{111,...,120}	{10,35,7,6,4,4,5,6,7,35,40}	BRU-SCH (H)
13	{121,...,140}	{30,24,4,4,3,3,5,6,6,10, 10,3,2,2,2,2,3,4,5,5,6}	BRU-AMA (L)
14	{131,...,140}	{15,8,6,5,4,5,6,7,10,12,10}	BRU-AMA (H)
15	{141,...,160}	{10,25,20,4,4,3,3,4,5, 6,10,4,4,3,2,2,3,3,4,4,4}	RTA-SCH (L)
16	{151,...,160}	{4,34,36,3,2,2,4,4,5,25,30}	RTA-SCH (H)
17	{161,...,180}	{20,40,10,5,4,3,4,5,5,6, 25,4,2,1,2,2,2,3,4,4,5}	PAR-AMA (L)
18	{171,...,180}	{5,50,25,25,3,4,5,6,6,35,40}	RTA-AMA (H)
19	{181,...,200}	{30,32,20,5,4,4,4,5,6,7, 20,4,4,3,2,3,3,4,4,5,5}	SCA-AMA (L)
20	{191,...,200}	{15,40,20,4,4,4,5,6,6,35,60}	SCA-AMA (H)

scenarios is illustrated in Table (2.12), where the first column relates to the length of the booking horizon. The Number of column generation iterations required by the algorithm appears in the third column, and number of different offer sets in the fourth. The last column displays the optimal revenue that is achieved by a control policy that ignores the bid price compatibility constraints.

Clearly, increasing the length of the booking horizon increases the number of variables and consequently the complexity of the problem, with a corresponding increase in processing time. As expected, decreasing the ratio of available capacity over demand (*i.e.* number of booking periods) also makes the problem more difficult. Nevertheless, computational times for the different scenarios testify to the capability of the algorithm to address realistic problems in a reasonable time frame, which would qualify for quasi-real time implementation.

Table 2.12 Comparision of results of railroad example

# periods	γ	# iterations	# offered sets	CPU time (seconds)	restricted revenue	upper bound
	0.5	3	1	18	31585	33365
T=100	1	3	1	18	31585	33365
	1.5	3	1	16	31585	33365
	0.5	7	4	592	146933	159797
T=500	1	4	3	218	157927	166163
	1.5	3	3	68	157927	166827
	0.5	13	6	4282	263314	286224
T=1000	1	5	3	611	295403	319595
	1.5	4	3	398	307849	330053

2.5 Conclusion

For various technical reasons, mathematical programming approaches like CDLP, that can provide high-quality solutions for the optimal allocation of resources in the context of revenue management, have not been well received in practice. Most firms actually based their policy on alternative approaches, when making sales decisions with respect to upcoming streams of demands from different type of customers.

In order to control the availability of products over a booking horizon, a common industry practice is that of bid prices, which are easy to understand and implement. Moreover, depute being suboptimal, they can readily be used to control the availability of group products.

In this paper, we have proposed a mathematical programming approach to compute improved bid prices in the customer choice-based network revenue management problem with the highest possible return. The novelty of our approach is that it is highly flexible and, unlike existing heuristic approaches from the literature, additional technical constraints can easily be incorporated within the mathematical framework. The efficiency of the method for addressing practical problems lies in a column generation algorithm that iteratively generates sets of products that enhance revenue while being compatible with a bid price control policy. Our numerical results show that the proposed algorithm outperforms alternative bid price control policies in most cases. Moreover, in the terms of processing times, it requires no more than a few minutes of processing time, and thus could be used in quasi-real time, which is more than sufficient in most applications.

While the present paper has focused on the multinomial logit model, it would be interesting to consider more sophisticated discrete choice models issued from the generalized extreme value family. From the mathematical programming point of view, another avenue for research is to improve the algorithm for solving, heuristically or not, the NP-hard sub problem. Finally, it is worth noting that a comprehensive study of the advantages and drawbacks of the various capacity control policies proposed in the literature has yet to be performed.

CHAPTER 3

ARTICLE 2 : NETWORK CAPACITY CONTROL UNDER A NONPARAMETRIC DEMAND CHOICE MODEL

Chapter Information : An article based on this chapter is submitted for publication M. Hosseinalifam, P. Marcotte, and G. Savard.

In this paper, we propose a nonparametric model for choice-based revenue maximization with corresponding algorithmic framework to solve practical large-scale problems.

ABSTRACT

This paper addresses a dynamic resource allocation problem which has its roots in airline revenue management, and where customers select the available product that ranks highest on a preset list of preferences. The problem is formulated as a flexible mathematical program that can easily embed technical and practical constraints, as well as accommodate hybrid (parametric-nonparametric) choice models. We propose for its solution a column generation algorithm whose performance, both in terms of solution quality and processing time, is assessed against that of alternative approaches.

Key words : Revenue management, customer choice behavior, mixed integer programming.

3.1 Introduction

Dynamic resource allocation is the main element of revenue management (RM), the discipline whose aim is to improve a firm's profitability through efficient pricing and asset management. These issues involve the design of decision rules that, over the booking horizon, allow or deny access to products that use common resources, based on assumptions concerning the behavior of customers facing distinct options (Talluri and Van Ryzin (2005)).

While traditional RM models assume cross-product independence, as well as independence from both capacity control strategies and from the state of the market (Talluri and Van Ryzin (2005)), more sophisticated discrete choice models have become increasingly popular (Talluri and Van Ryzin (2005)). These posit that customer behavior is dictated not only by product availability, but also by products' attributes (price, quality, restrictions, willingness to pay, etc.) (Talluri and Van Ryzin (2004)). In turn, alternative parametric models that obviate some of these models' limitations have been proposed Farias et al. (2013)).

In this paper, departing from the parametric approach, we consider a framework where the customer population is partitioned into segments, each segment being associated with an ordered list of product preferences that includes the 'no purchase' option. This demand model is then embedded within a capacity control system. More precisely, given a stochastic arrival process that governs each class, acceptance rules that determine the optimal set of products offered in each time period are obtained through the solution of a deterministic mathematical program. An important feature of the model is its flexibility with respect to additional constraints. In particular, it can accommodate arbitrary topologies that go beyond the traditional hub-and-spoke architecture, as well as user-specific constraints.

Our contribution is twofold. First, we propose a nonparametric model for choice-based revenue maximization, through the specification of the optimal sets of products that are made available at each booking period. Next, in the view that the number of variables grows exponentially with the number of products, we develop an efficient column generation algorithm that exploits the specific structure of the choice model, and has the capability of addressing real-life instances.

The structure of the paper is as follows. In Section 2, we review the main concepts of choice-based demand models, contrasting the parametric and nonparametric approaches. In Section 3, we introduce our mathematical programming framework. In Section 4, we develop a column generation scheme for its solution ; in particular, we provide an efficient algorithm for tackling the nonconvex subproblems. In Section 5, we illustrate through computational experiments that our approach can address realistic instances, and provide a comparison with alternative approaches from the RM literature. Finally, in the concluding section, we outline the challenges that remain to be addressed.

3.2 Choice modelling

In this section, we briefly survey choice modelling issues. Indeed, a key issue in network revenue management is that of estimating the probability $P_j(S)$ that a product j be selected by an arriving customer, given that a set S of products is on offer. Two main classes of models have been proposed for its solution. Parametric choice models are built upon the Random Utility Maximization paradigm (McFadden (2000)), whereby products are assigned attributes, and customers select the product that maximizes their own utility, expressed as a weighted sum of the attributes' values. Depending on the statistical model underlying the selection process, one derives a variety of models : multinomial logit (MNL), nested logit, mixed logit, probit, generalized extreme value, etc. For instance, in the MNL model, which is widely used in marketing and economics (Train (1986)), customers belong to predefined segments characterized by a weight vector v associated with products' attributes. The probability of selecting product j is then set to the ratio of that product's preference for the customer over the sum of all products' preferences.

Although random utility models are easy to understand, embed detailed information about products' features, and allow the accurate estimation of utilities, they yet suffer serious flaws. First, the choice of an appropriate parametric structure may not be obvious and, once a structure is adopted, the model is not flexible with respect to perturbations in the available information (Farias et al. (2013)). Next, specific random utility models have specific drawbacks. For example, MNL's independence of irrelevant alternatives property yields unrealistic substitution patterns, while the more sophisticated nested or mixed logit models are computationally challenging, both from an estimation and assignment viewpoint.

In contrast, nonparametric choice models are driven by historical data and do not assume specific probability distributions. They are highly flexible, dynamic, and provide more precise estimates of customer's choice behavior (Farias et al. (2013)). Due to the availability and increasing accuracy of large amounts of historical data, these choice models have been gaining in popularity and interest.

In the nonparametric choice model adopted in this paper, we assume that each demand segment is characterized by an Ordered Preference List (OPL), whereby customers select the available product that ranks highest on their OPL, possibly leaving the market if no available product belongs to the list (Chaneton and Vulcano (2011a), Chen and Homem-de Mello (2010)). The concept of OPL was first introduced by Mahajan and van Ryzin (2001), while Farias et al. (2013) and van Ryzin and Vulcano (2011) proposed different procedures to estimate a non-parametric choice model. Within this framework, van Ryzin and Vulcano (2008a) and Chaneton and Vulcano (2011a) used the concept of OPL to formulate a choice based capacity control model, and proposed for its numerical solution a stochastic gradient algorithm. Farias et al. (2011) developed an algorithm to compute optimal assortments under a nonparametric choice model of demand. Mahajan and van Ryzin (2001) proposed a model to compute optimal retail assortments, in an environment where customers adapt dynamically to available stocks. Chen and Homem-de Mello (2010) have proposed OPL-based linear stochastic formulations of the revenue maximization problem which, unfortunately, become intractable as the number of scenarios grows. Our approach shares several features with this work, while lifting its computational limitations.

3.3 Problem formulation

Let us consider a system where an arrival stream of customers follows a Poisson process with rate λ. Whenever a customer shows up, she selects the available product that ranks highest on her preference list. The aim of the model is to determine, over a finite planning horizon, the set of products to be offered at any given 'booking' period, in order to maximize total revenue. Of course, a product can only enter the offer set if the amount of resources required does not exceed the residual amount available.

The main parameters underlying the dynamic RM model are the following :

T : ordered set of time (booking) periods indexing forward

J : set of products

r_j : revenue associated with product $j \in J$

I : set of resources

c_i : initial amount of resource $i \in I$

L : set of customer segments

p_l : proportion of customers belonging to segment l

$\lambda_l = \lambda p_l$: arrival rate of segment $l \in L$

$P_l = \exp(-\lambda p_l)$: probability of an arrival issued from segment $l \in L$ within an arbitrary time period

$O^l = \{j_1^l, j_2^l, \ldots, j_{K_l}^l\}$: ordered preference list (OPL) of products associated with customer segment $l \in L$

a_{ij} : Boolean constant indicating whether resource i is used by product j ($a_{ij} = 1$) or not ($a_{ij} = 0$). The matrix A whose elements are the a_{ij}'s is referred to as the product-resource incidence matrix or, simply, the incidence matrix.

$S \in 2^J$: set of products, possibly including the 'null' product

$O^l(S) = \{j_1^l(S), j_2^l(S), \ldots, j_{K_l(S)}^l(S)\} \subseteq O^l$: ordered preference list (OPL) of cardinality $K_l(S)$ associated with offer set S and customer segment $l \in L$.

Assuming that there is at most one arrival within any given time period [1], the probability of choosing product j when set S is offered is equal to

$$P_j(S) = \sum_{l: j_1^l(S) = j} P_l, \tag{3.1}$$

where $j_1^l(S)$ is the first available preferred product of segment l among those belonging to the offer set S. This yields the expected revenue

$$R(S) = \sum_{j \in S} P_j(S) r_j \tag{3.2}$$

and the expected capacity usage of set S

$$Q_i(S) = \sum_{j \in S} P_j(S) a_{ij}. \tag{3.3}$$

The variables of the model are the indicators $X_t(S)$, which specify whether the subset of products S is offered or not in period t. For the sake of computational tractability, we allow these binary variables to assume fractional values. Based upon these definitions, letting $X = (X_t(S))_{t,S}$, and denoting by $|E|$ the cardinality of a generic set E, the model can be expressed as the linear program

$$\text{LP}: \qquad \max_X \lambda \sum_{t \in T} \sum_{S \in 2^J} R(S) X_t(S)$$

subject to

$$\sum_{t \in T} \sum_{S \in 2^J} \lambda Q_i(S) X_t(S) \le c_i \qquad \forall i \in I, \tag{3.4}$$

1. This assumption is reasonable if the duration of a time period is sufficiently small.

$$\sum_{S \in 2^J} X_t(S) \leq 1, \qquad \forall t \in T, \tag{3.5}$$

$$0 \leq X_t(S) \leq 1 \qquad \forall t \in T, S \in 2^J, \tag{3.6}$$

where the number of decision variables $2^{|J|} - 1$ is exponential (the empty set is excluded).

Note that the above program is similar to the customer MNL-based deterministic linear programming model (CDLP) considered by Liu and van Ryzin (2008) and Bront et al. (2009), the main differences being the way we model customer's choice behavior, and the way we compute the related probabilities that lead to the values of $R(S)$ and $Q_i(S)$. Note also that, in contrast with Liu and van Ryzin (2008) and Bront et al. (2009), our decision variables are related to individual time periods, thus allowing a finer control over the individual booking periods. Finally, the use of ordered preference lists allows to overcome the drawbacks of the MNL model, e.g., unrealistic substitution patterns resulting from the IIA property.

From the computational viewpoint, the complexity of LP is directly related to the number of OPLs and products, which increases sharply with the number of products and with a finer representation of the demand, i.e., the number of segments. We now show how to reduce the number of OPLs with no impact on the solution.

Definition 3.3.1 *Let $O^1 = \{j_1, j_2, \ldots, j_k\}$ and $O^2 = \{j_1, j_2, \ldots, j_k, j_{k+1}, \ldots, j_m\}$. We then say that O^1 is nested in OPL O^2.*

If two OPLs are nested, then they can be aggregated into a single OPL. This is achieved by introducing transition probabilities from high rank to low rank products. Standard OPLs are actually aggregated OPLs where those probabilities are equal to one. In practice, since many OPLs are nested, their aggregation allows to significantly decrease the size of the problem.

The transition probabilities are defined as follows. Let $P^+_{k\bar{l}}$ denote the transition probability from the product in rank k to the next one in rank $k+1$ in the aggregated OPL $\bar{l}$, and let P^z_l denote the probability of having an arrival from segment l in which z is the least preferred product in that segment. We have that $P^+_{1\bar{l}} = 1$. Since P^z_l must equal the probability of no-purchase after having considered product z in the aggregated OPL $\bar{l}$, times the product of former transition probabilities, we obtain

$$P^{+}_{k\bar{l}} = 1 - \frac{P_l^{z=j_k^l}}{\prod_{q=1}^{k-1} P^{+}_{q\bar{l}}}. \tag{3.7}$$

An example of an aggregated OPL and its parameters is illustrated in Table 3.1. If the preferred product A of customer is unavailable, she substitutes for product B with probability 0.9, or leaves the market with complementary probability 0.1. These transition probabilities can actually be interpreted as buy-up or buy-down coefficients of the market (Chen and Homem-de Mello (2010)).

Table 3.1 Standard VS aggregated OPL

P_l	standard OPL
$\frac{1}{10}$	$A \rightarrow X$
$\frac{2}{10}$	$A \rightarrow B \rightarrow X$
$\frac{3}{10}$	$A \rightarrow B \rightarrow C \rightarrow X$
$\frac{4}{10}$	$A \rightarrow B \rightarrow C \rightarrow D \rightarrow X$

$P_{\bar{l}}$	Aggregated OPL
1	$\begin{array}{lllllllll} A & \xrightarrow{\frac{9}{10}} & B & \xrightarrow{\frac{7}{9}} & C & \xrightarrow{\frac{4}{7}} & D & \xrightarrow{1} & X \\ \frac{1}{10}\searrow & & \frac{2}{9}\searrow & & \frac{3}{7}\searrow & & & & \\ & X & & X & & X & & & \end{array}$

Now, the probability P_l^C of having an arrival from a segment with least preferred product C is given by the product of all transition probabilities yielding C by the probability of leaving system after having visited C, (1 - $P^{+}_{3\bar{l}}$). Thus we obtain the equation $3/10 = (9/10 \times 7/9) \times (1 - P^{+}_{3\bar{l}})$, whose solution yields $P^{+}_{3\bar{l}} = 4/7$.

Of course, when an aggregated OPL is created, $P_j(S)$, the probability of choosing product j when set S must be adjusted accordingly.

Let $P^{++}_{kl} = \prod_{q=1}^{k-1} P^{+}_{ql}$ denotes the total transition probability till product in rank k in

OPL l. Then we have

$$P_j(S) = \sum_{l:j_1^l(S)=j} P_{kl}^{++} P_l, \tag{3.8}$$

where k is the rank of the product j in the OPL l.

3.4 A column generation framework

While the number of variables in LP is exponential ($2^J - 1$ potential offer sets), the number of constraints (with the exception of bound constraints on the variables), is limited to $|M|+|T|$. This suggests the use of column generation techniques for addressing LP (Bront et al. (2009)). More precisely, let us consider a feasible basic solution of LP, together with the multiplier vectors π and σ associated with the capacity and time constraints, respectively. A maximal reduced cost can be obtained by solving the subproblem

$$\max_{S \in 2^J} R(S) - \sum_{i \in I} \pi_i Q_i(S) - \sum_{t \in T} \sigma_t, \tag{3.9}$$

whose solution, whenever its objective is positive, yields an improving offer set S.

To this aim, we introduce the binary variables $y_{j_k^l}$ that are set to 1 if the product ranked k in OPL l belongs to S, and to 0 otherwise. Only one product will be chosen from any OPL, and after the product in rank k from OPL l has been selected, the products with ranks higher than k should be inactive. Note that j_k^l represents a unique index j and that, once a product $j = j_k^l$ is active, it will be considered active in all OPLs. To enforce this property, we introduce binary variables h_k^l that denote the rank of customer l's preferred available product. The column generation subproblem can then be formulated as the mixed integer program

$$\text{SUB}: \qquad \max_{y,h} \sum_{l \in L} \sum_{k=1}^{|O^l|} P_{kl}^{++} P_l (r_{j_k^l} - \sum_{i \in I} a_{i,j_k^l} \pi_i) y_{j_k^l} (1 - h_k^l)$$

subject to

$$y_{j_k^l} \leq h_{k+1}^l, \qquad \forall l \in L, k \in \{1, \ldots, |O^l| - 1\}, \tag{3.10}$$

$$h_k^l \leq h_{k+1}^l, \qquad \forall l \in L, k \in \{1, \ldots, |O^l| - 1\}, \tag{3.11}$$

$$y_{j_k^l}, h_k^l \in \{0, 1\} \qquad \forall l \in L, k \in \{1, \ldots, |O^l|\}.$$

In the above program, constraints (3.11) ensure that, once the product in rank k of OPL l is selected, h_{k+1}^l is set to 1 for products further down the list. In the objective function, the product $y_{j_k^l}(1 - h_k^l)$ controls the activation state of any product in the OPL, i.e., the highest ranked available product is selected, while the remaining ones are ignored. The definition of the variables h_k^l, together with constraints (3.10) and (3.11), ensures that any arriving customer purchases his highest ranked product within S.

The bilinear terms $y_{j_k^l} \times h_k^l$ can be linearized through the introduction of variables $z_k^l = y_{j_k^l} \times h_k^l$ and of the inequalities $z_k^l \leq y_{j_k^l}$, $z_k^l \leq h_k^l$, $z_k^l \geq y_{j_k^l} + h_k^l - 1$. This yields the equivalent mixed integer formulation

$$\text{SUB' :} \qquad \max_{y,h} \sum_{l \in L} \sum_{k=1}^{|O^l|} P_{kl}^{++} P_l (r_{j_k^l} - \sum_{i \in I} a_{i,j_k^l} \pi_i)(y_{j_k^l} - z_k^l)$$

subject to

$$y_{j_k^l} \leq h_{k+1}^l, \qquad \forall l \in L, k \in \{1, \ldots, |O^l| - 1\},$$

$$h_k^l \leq h_{k+1}^l, \qquad \forall l \in L, k \in \{1, \ldots, |O^l| - 1\},$$

$$z_k^l \leq h_k^l \qquad \forall l \in L, k \in \{1, \ldots, |O^l|\},$$

$$z_k^l \leq y_{j_k^l} \qquad \forall l \in L, k \in \{1, \ldots, |O^l|\},$$

$$z_k^l \geq y_{j_k^l} + h_k^l - 1 \qquad \forall l \in L, k \in \{1, \ldots, |O^l|\},$$

$$y_{j_k^l}, h_k^l, z_k^l \in \{0, 1\} \qquad \forall l \in L, k \in \{1, \ldots, |O^l|\},$$

which is amenable to an off-the-shelf solver, even for large scale instances, due to its compact formulation in terms of decision variables. Note also that the formulation is quite flexible, allowing the construction of arbitrary sets S compatible with the OPLs of the population's segments.

3.5 Numerical results

The performance of the numerical framework has been tested on two classical airline data, involving either a 'parallel' or 'hub-and-spoke' topology, as well as a railroad problem involving real size data. In the first two cases, our results are compared to those obtained by (Chen and Homem-de Mello (2010)), who also proposed an OPL-based model.

To assess the quality of our resource allocation model, average revenues over 2000 random streams of arrivals, each based on the arrival rates λ_l, have been computed, yielding accuracy levels of 0.5% with 95% confidence. For the record, computational results have been carried out on a 4-core, 2.4 GHz computer, with 8 GB of memory. Both the master problem LP and the mixed integer subproblems SUB' have been solved by FICO Xpress-Mosel 7.2.1.

3.5.1 Parallel flights

In this example taken from Chen and Homem-de Mello (2010), we consider a network with three parallel flights distinguished by their departure time : morning, afternoon, or evening. Initial capacities are set to 30, 50 and 40, respectively. A total number of six products are induced by these resources and the two fare classes : High (H) and Low (L) . Products' information and resource usage are shown in Figure 3.1 and Table 3.2.

The information concerning the four aggregated OPLs is displayed in Table 3.3. This includes the preference lists, the transition probabilities, and the demand corresponding to each segment. The booking period is divided into $\tau = 300$ periods, and the average number of arrivals is set to 150. To assess the robustness of our approach, we applied our control policy to demand scenarios where the base demand is multiplied by the scale factor α. In the 'low' scenario, only 75% of the forecasted demand arrives. In the 'high' scenario, 125% of the

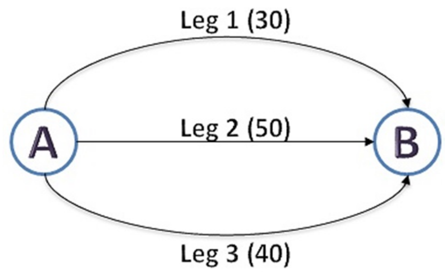

Figure 3.1 Parallel flight example : network.

Table 3.2 Parallel flight example : products

product	leg	class	fare
1	1	L	400
2	1	H	800
3	2	L	500
4	2	H	1000
5	3	L	300
6	3	H	600

forecasted demand is simulated.

Table 3.3 OPL setting for the parallel flights

OPL	ordered itineraries	transition probability	demand level
1	4, 2, 6	1, 0.5, 0.2	30
2	5, 1, 3	1, 0.5, 0.2	45
3	1, 2, 3, 4, 5, 6	1, 0.8, 0.75, 0.66, 0.75, 0.33	60
4	2, 1, 4, 3, 6, 5	1, 0.8, 0.75, 0.66, 0.75, 0.33	15

The total expected revenues are displayed in Table 3.4, where the first column denotes the scale factor, and the next four columns represent results obtained by different approaches developed in Chen and Homem-de Mello (2010).

The column NB^{LP} represents the optimal solution of the linear estimation of the stochastic model without implementing their proposed backup heuristic. The column NB^{SP}

represents the optimal solution of the stochastic model without implementing their proposed backup heuristic. The column DB^{LP} represents the optimal solution of the linear estimation of the stochastic model with implementing their backup heuristic and the column DB^{SP} represents the optimal solution of the stochastic model with implementing their backup heuristic.

Table 3.4 Parallel flights : 95% confidence intervals for the revenue.

Demand	NB^{LP}	NB^{SP}	DB^{LP}	DB^{SP}	CDLP	OPL	PI
α=0.75	65622	65651	65988	**66123**	65994	65766	66180
	±122	±123	±131	**±131**	±139	±171	±119
α=1	78706	79252	80711	81287	79917	**81446**	82334
	±83	±96	±86	±91	±108	**±130**	±94
α=1.25	83076	84684	83909	84866	83241	**85236**	87652
	±34	±44	±13	±15	±52	**±194**	±21

Column CDLP represents the results obtained by CDLP model under buy up/down structure. Column OPL presents results obtained by our column generation algorithm and the last column PI represents an upper bound to the problem. Throughout, one observes that algorithm OPL outperforms alternative ones on two out of three scenarios.

3.5.2 Hub and Spoke

This example is also borrowed from Chen and Homem-de Mello (2010). Its network (see Figure 3.2 and Table 3.5) consists of seven flight legs that induce a total number of twenty two itineraries (products), some of which use more than one resource. Initial resource capacities are set to $c = (100, 150, 150, 150, 150, 80, 80\ )$.

The booking horizon consists of $\tau = 1000$ periods, and the average number of arrivals 910 corresponds to the arrival rate $\lambda = 0.91$.

Displayed in Table (3.6) are ten aggregated OPLs with corresponding transition probabilities provided in the third column. It is important to note that, in Chen and Homem-de

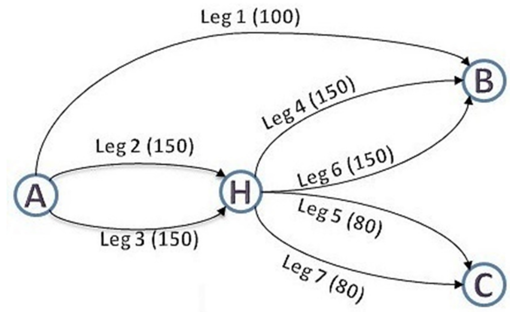

Figure 3.2 Hub and Spoke example : network.

Table 3.5 Hub and Spoke example : products

product	leg	class	fare	product	leg	class	fare
1	1	H	1000	12	1	L	500
2	2	H	400	13	2	L	200
3	3	H	400	14	3	L	200
4	4	H	300	15	4	L	150
5	5	H	300	16	5	L	150
6	6	H	500	27	6	L	250
7	7	H	500	28	7	L	250
8	{2,4}	H	600	19	{2,4}	L	300
9	{3,5}	H	600	20	{3,5}	L	300
10	{2,6}	H	700	21	{2,6}	L	350
11	{3,7}	H	700	22	{3,7}	L	350

Table 3.6 Hub and Spoke example : OPLs.

OPL	ordered itineraries	transition probability	demand level
1	1, 8, 9, 12, 19, 20	1, 0.8, 1, 0.75, 0.66, 1	80
2	20, 19, 12, 9, 8, 1	1, 1, 0.8, 0.25, 1, 0.5	200
3	2, 3, 13, 14	1, 1, 0.5, 1	50
4	14, 13, 3, 2	1, 1, 0.2, 1	200
5	4, 5, 15, 16	1, 1, 0.5, 1	100
6	16, 15, 5, 4	1, 1, 0.2, 0.8	150
7	6, 7, 17, 18	1, 0.8, 0.625, 1	20
8	18, 17, 7, 6	1, 1, 0.2, 0.8	50
9	10, 11, 21, 22	1, 0.8, 0.625, 1	20
10	22, 21, 11, 10	1, 1, 0.2, 1	40

Mello (2010), products belong to a specific control class, and the control policy is class based. For the sake of consistency, we generated product data consistent with their respective class data. The total expected revenues corresponding to different algorithms are presented in Table 3.7. Again, and yet more clearly than in the 'parallel flights' example, our algorithm outperforms alternative approaches.

Table 3.7 Hub and Spoke : 95% confidence intervals for the revenue.

Demand	NB^{LP}	NB^{SP}	DB^{LP}	DB^{SP}	CDLP	OPL	PI
α=0.75	210730	195630	211730	211760	211340	**222861**	227013
	±165	±213	±177	±168	±213	**±191**	±145
α=1	246780	255670	248910	256950	246970	**263526**	268277
	±79	±163	±73	±159	±215	**±189**	±147
α=1.25	254090	274050	254160	274320	255810	**276292**	304223
	±19	±61	±20	±67	±98	**±179**	±130

3.5.3 Railroad

Our third example involves a subset of the Thalys system, which operates high speed trains through four countries : France, Belgium, Netherlands, and Germany. It involves five cities and four legs, with two fare classes (low 'L' and high 'H') on each leg. The network and its associated market are illustrated in Figure 3.3.

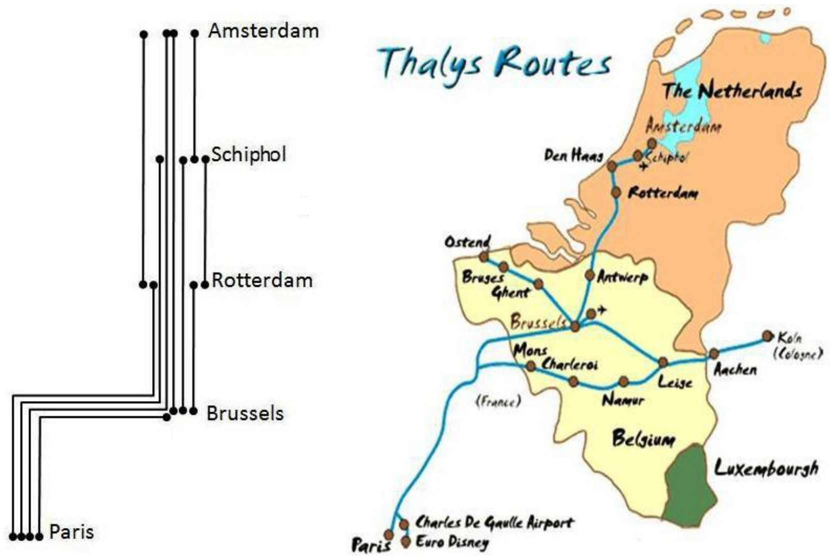

Figure 3.3 Thalys railroad system.

In this environment, 10 trains with a capacity of 100 passengers travel between Paris and Amsterdam. Each train stops in Brussels, Rotterdam, Schiphol, and Amsterdam, inducing the 10 markets shown in Figure 3.3, as well as 200 products, i.e., train-fare combinations. Price information associated with each market is displayed in Table ??.

Based on their price sensitivity and origin-destination assignments, customers are divided into 20 segments. Price sensitive (leisure) customers prefer low fare, but are allowed to switch to high fare products, while price insensitive (business) customers stick to high fares. The features of each segment of the population are shown in Table 3.9.

We solve the problem in different booking horizons. Indeed, if the capacity of legs exceeds the corresponding demand, the problem becomes much easier to solve and the firm could offer

Table 3.8 Market fares

market	low fare	high fare
PAR $\rightarrow$ BRU	200	400
PAR $\rightarrow$ RTA	300	500
PAR $\rightarrow$ SCH	350	525
PAR $\rightarrow$ AMA	350	525
BRU $\rightarrow$ RTA	150	250
BRU $\rightarrow$ SCH	175	275
BRU $\rightarrow$ AMA	200	300
RTA $\rightarrow$ SCH	50	100
RTA $\rightarrow$ AMA	175	300
SCH $\rightarrow$ AMA	50	100

almost all of its products. To better evaluate algorithms, we consider different capacities by multiplying a scale factor γ to the capacity of legs c.

Besides altering the capacity of legs, we also perturbed the length of the booking horizon, thus significantly increasing the number of variables. Nevertheless, even for the largest instances, the CPU was kept in check. Actually, the algorithm scaled quite well, with running times increasing linearly with the length of the booking horizon (see Table 3.10).

3.5.4 Summary of numerical results

The simulation results have provided an unbiased account in favor of our deterministic model. A close runner-up is the stochastic model DB^{SP} (Chen and Homem-de Mello (2010)), involving or not a backup heuristic, which unfortunately does not scale well with problem size. While Chen and Homem-de Mello (2010) do not provide the CPU requested by their method, ours consistently requires less than one minute of CPU for solving the parallel flight examples, and less than three minutes for the larger Hub and Spoke instances. This is a good indication that our approach is able to determine the optimal assortment of products to be offered over the booking horizon, even for realistic problem sizes.

Table 3.9 OPL settings for the railroad problem.

OPL	ordered itineraries	demand	description
1	8,6,9,10,5,1,12,11,4,13,3,2,17,19,18,16,20, 15,7,14,	20	PAR-BRU(L)
2	17,16,15,18,14,13,19,20,12,11,	25	PAR-BRU(H)
3	28,23,26,22,27,21,34,40,39,36,35,33,38,37, 24,31,30,29,25,32,	10	PAR-RTA(L)
4	35,34,36,33,37,38,39,32,31,40,	20	PAR-RTA(H)
5	46,45,44,48,47,60,49,42,41,50,54,53,52,57, 56,55,59,58,43,51,	30	PAR-SCH(L)
6	54,53,55,56,58,57,59,52,60,51,	10	PAR-SCH(H)
7	64,63,66,65,68,67,69,70,61,62,75,74,72,71, 80,79,77,76,73,78,	20	PAR-AMA(L)
8	74,73,76,75,77,78,72,71,79,80,	20	PAR-AMA(H)
9	85,84,86,83,89,91,96,95,94,97,93,99,98,92, 100,87,88,90,82,81,	10	BRU-RTA(L)
10	95,96,97,93,94,98,92,99,100,91,	30	BRU-RTA(H)
11	104,103,106,105,110,109,102,108,107,101, 115,114,113,117,116,112,120,119,118,111,	15	BRU-SCH(L)
12	115,114,116,117,113,118,112,119,111,120,	30	BRU-SCH(H)
13	126,140,128,127,130,129,121,135,134,133, 132,136,131,125,124,137,123,122,139,138,	20	BRU-AMA(L)
14	134,135,133,136,132,137,131,140,138,139,	20	BRU-AMA(H)
15	147,144,143,148,149,150,142,141,155,154, 157,156,153,146,145,160,159,158,152,151,	30	RTA-SCH(L)
16	155,154,153,157,156,158,159,160,151,152,	10	RTA-SCH(H)
17	166,164,180,168,167,163,169,162,170,161, 173,176,175,174,172,177,165,179,178,171,	20	PAR-AMA(L)
18	174,175,176,178,177,173,172,179,180,171,	20	RTA-AMA(H)
19	199,187,183,188,189,190,182,181,194,196, 195,193,198,197,192,191,186,185,184,200,	10	SCA-AMA(L)
20	186,185,200,199,187,183,188,189,190,182,	30	SCA-AMA(H)

Table 3.10 Average processing time for the railroad example (seconds)

scale factor	booking horizon		
	T=300	T=400	T=500
γ=0.5	830	1100	1630
γ=1	440	770	950
γ=1.5	400	630	700

3.6 Conclusion and further work

It has long been recognized that any reliable revenue management system must integrate a sound model of customer behavior. In this paper, we have adopted a nonparametric approach that mimics the choice process of individual customers, and dispenses with the costly statistical process that consists in estimating the parameters underlying a discrete choice model. This results in a simple model that lends itself to an efficient and scalable decomposition algorithm.

Of course, nonparametric models have drawbacks. For instance, it is a nontrivial task to partition the population into potentially large number of segments, one per preference list, and this segmentation might actually be highly sensible to fare variations, which we did not consider in our model. Nevertheless, an interesting feature of the model is that it can accommodate hybrid choice models. To do so, we need only define a simulator compatible with the parameters of the choice model, and obtain OPLs by ranking the utilities of available choices. This information can then be used as an input to the optimization model. Moreover, parametric models can be used to calibrate available historical data and improve the accuracy of OPLs.

On the algorithmic side, we note that the column-generation subproblem could be efficiently addressed by an off-the-shelf solver. To deal with much larger instances, algorithms adapted to the specific structure of the subproblems are required. This will be the topic of further research.

CHAPTER 4

ARTICLE 3 : COMPUTING BOOKING LIMITS UNDER A NON-PARAMETRIC DEMAND MODEL : A MATHEMATICAL PROGRAMMING APPROACH

Chapter Information : An article based on this chapter is submitted for publication M. Hosseinalifam, P. Marcotte, and G. Savard.

In this paper, we develop a new customer choice-based framework for computing nested booking limits that yield the highest possible return, within a given non-parametric customer choice environment.

ABSTRACT

In revenue management, booking limits are commonly used to restrict access to classes of products, and subsequently make way for more profitable ones. Frequently, this inventory control policy assumes that products are nested in decreasing order of revenue, and that less profitable products are denied access first. In this paper, we propose a flexible mathematical programming framework for the computation of nested booking limits, under the assumption that customers are characterized by ordered lists of their preferences. This yields a mixed integer program that is amenable to an efficient and scalable column generation algorithm. Numerical tests illustrate the improved performance of the resulting policies, which are numerically tested against alternative proposals from the current literature, the latter being admittedly scarce.

Key words : Revenue management, network capacity control, booking limits, customer choice behavior, mathematical programming.

4.1 Introduction

A revenue management system consists of a set of scientific tools for managing, over a finite horizon, demand requests issued from heterogeneous customers, given that resources are in fixed supply. Demand forecasting, pricing and structural information constitute the input data for the inventory control system, and output is the decision of accepting or rejecting the arriving requests, with the aim to maximize the firm's revenue (Talluri and Van Ryzin (2005)).

In this paper, although we adopt the vocabulary common in the airline industry, our results easily translate to the rail, media or hospitality environments. In the airline industry, resources are matched with actual seats on the flight legs making up an itinerary, and products correspond to tickets on flight leg(s) associated with specific origin-destination pairs.

For any given flight, the corresponding product is associated with a *control class* (*i.e.*, a group of products with similar features like revenue, restrictions, etc) on the legs that it uses : first, leisure, coach, etc. Most Central Reservation Systems (CRS) adopt control policies that fall into one of two categories : booking limits and bid prices (Chaneton and Vulcano (2011b)), the first one being prevalent. A booking limit policy sets, for each control class, an upper bound on the number of seats that are allocated to the control classes on the leg that seats belong. Its aim is to control access to products, in order to avoid rejecting future customers characterized by a high 'willingness-to-pay' (van Ryzin and Vulcano (2008a)).

Two main categories of booking limits have been considered : partitioned and nested as seen in Figure 4.1 In the partitioned case (left figure), a specific amount of the resources is allocated exclusively to each control class, *i.e.*, distinct classes do not have access to shared capacity. In contrast for the nested case (right figure), control classes with higher revenue ('parent' classes) have access to the assigned capacity of all control classes with lower revenue ('child' classes). In this paper, we consider full nesting by setting booking limits on control classes associated to each leg.

In this example (Figure 4.1), with nested booking control policy, control class Y is a parent control class for classes M and K and have access to all of their seats. There are 5, 10 and 5 seats assigned for control classes Y, M and K, with booking limits 20, 15 and 5, respectively. Moreover, 5 and 10 seats are protected for class Y and M with respect to their child classes and no seat are protected for control class K, *i.e.,* all parent control classes have recursively access to the seats of their children classes.

This strategy avoids selling out lower control class products while there is still unused capacity in the higher ones (van Ryzin and Vulcano (2008b)).

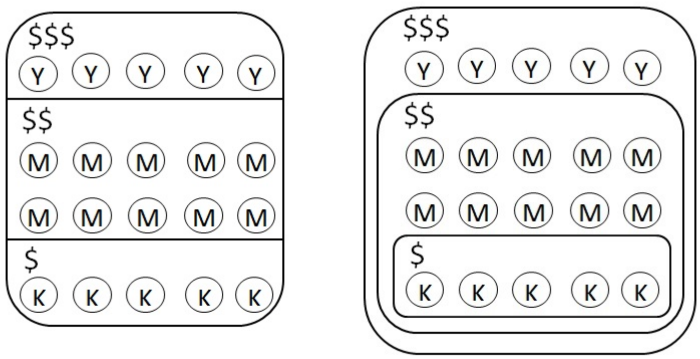

Figure 4.1 Partitioned VS nested booking limits.

The calculation of booking limits for the 'standard' revenue problem was initiated by Littlewood (1972), followed by Belobaba (1987) and Belobaba and Hopperstad (1999), who developed two heuristic approaches based on expected marginal seat revenue (EMSRa and EMSRb). These were applied to problems involving multiple products and a single resource. More recently, Bertsimas and De Boer (2005) proposed a simulation-based approach to compute booking limits over a network. As a first step, these authors used simulation to estimate the first order gradient of the revenue function. Next, this estimation was used to improve an initial set of booking limits via a stochastic steepest ascent algorithm. van Ryzin and Vulcano (2008b) also proposed, based on a continuous capacity-demand model, a simulation-based approach, which was extended to a general choice model. A drawback of this approach is the difficulty of embedding specific constraints.

As a ‘dual’ alternative to booking controls, bid price policies set threshold prices on each leg (resource), and a request is accepted only if its revenue exceeds the sum of bid prices of its constituent legs. Although it may yield suboptimal solutions, this policy frequently achieves high revenue, and is furthermore very easy to understand and implement (Chaneton and Vulcano (2011b)). Despite those qualities, many reservation systems still use booking limits, which provide a direct control of the arriving requests.

Besides bid prices and booking limits, another policy has been introduced, mainly in the academic literature. It consists in determining, over a booking horizon, the optimal allocation of resources through the solution of a choice-based linear program (CDLP, see Liu and van Ryzin (2008) and Bront et al. (2009)). Although it is theoretically superior to the other two methods, CDLP has made few inroads into the industry, mainly due to its complexity and non-intuitiveness.

In this context, the paper's focus is to develop a new CDLP-based framework for computing nested booking limits that yield the highest possible return, within a given customer choice environment. With respect to the latter, we assume that customers rank the products from most to least preferred, and select the one that sits highest on this preference list (Chaneton and Vulcano (2011b), Chen and Homem-de Mello (2010)). This nonparametric approach is in contrast with extreme value models, such as the multinomial logit, which require the estimation of several parameters.

The paper is structured as follows. In Section 2, we introduce the notation, and formulate a mathematical program that yields optimal resource allocation within a nonparametric demand environment. In Section 3, we provide a detailed account of the column generation approach used for its solution. In Section 4 we illustrate the efficiency of the method, both from the quality and computational points of view ; this is achieved via benchmarks from the literature, as well as through numerical simulation. Finally, the concluding section mentions avenues for further research.

4.2 Problem formulation

In this section, we introduce two mathematical programming formulations for computing optimal booking limits, under the requirement that they be nested. From the modelling point of view, the main challenge lies in enforcing the compatibility of the nestedness property within a CDLP-based framework, while taking into account customer behavior.

4.2.1 General definitions and notations

Let us consider a system where supply of resources is limited, and where demand is characterized by Poisson processes associated with customer classes (segments), each one endowed with its own intensity and ordered preference list (OPL). Recall the assumption that customers rank the products from most to least preferred, and select the one that sits highest on this preference list. If her first preferred product was not offered, then with a transition (buy up/down) probability, she will move to her next most preferred choice. Decisions concerning the subset of products to offer are made at discrete time steps, and we assume that the width of the time intervals is small enough that the probability of more than one arrival within each interval is negligible.

The notation is based on that of the companion paper Hosseinalifam et al. (2014), where additional information is provided. Throughout, we denote by $|E|$ the cardinality of a generic set E.

T : ordered set of booking periods indexing forward

C : ordered set of control classes indexed forward in decreasing hierarchical order. *e.g.*, class 1 is parent for class 2, class 2 for class 3, etc.

J : set of products

r_j : revenue associated with product $j \in J$

I : set of resources

x_i : initial amount of resource $i \in I$

L : set of customer segments

λ : total arrival rate

$\lambda_l = \lambda p_l$: arrival rate of segment $l \in L$

p_l : proportion of customers belonging to segment l

$P_l = \exp(-\lambda p_l)$: probability of an arrival issued from segment $l \in L$ within an arbitrary time period

$O^l = \{j_1^l, j_2^l, \ldots, j_{K_l}^l\}$: ordered preference list (OPL) of products associated with customer segment $l \in L$

a_{ij} : Boolean constant indicating whether resource i is used by product j ($a_{ij} = 1$) or not ($a_{ij} = 0$). The matrix A whose elements are the a_{ij}'s is referred to as the product-resource incidence matrix or, simply, the incidence matrix.

a_j^c : Boolean constant indicating whether product j belongs to control class c ($a_j^c = 1$) or not ($a_j^c = 0$).

$S \in 2^J$: set of products, possibly including the 'null' product

$O^l(S) = \{j_1^l(S), j_2^l(S), \ldots, j_{K_l(S)}^l(S)\} \subseteq O^l$: ordered preference list (OPL) associated with offer set S and customer segment $l \in L$

P_{kl}^+ : transition probability from the product in rank k to the next one in rank $k+1$ in the OPL l.

$P_{kl}^{++} = \prod_{q=1}^{k-1} P_{ql}^+$: total transition probability from the customer's most preferred product to the one in rank k in the OPL l.

The decision variables of the model, denoted $X_t(S)$, refer to the fraction of period t in which set S is offered. If set S is offered, the probability of choosing product j is equal to

$$P_j(S) = \sum_{l: j_1^l(S) = j} P_{kl}^{++} P_l, \tag{4.1}$$

where $j_1^l(S)$ is the available highest ranked product of segment l belonging to set S with actual rank of k. This yields the expected revenue

$$R(S) = \sum_{j \in S} P_j(S) r_j \tag{4.2}$$

and the expected capacity usage

$$Q_i(S) = \sum_{j \in S} P_j(S) a_{ij}. \tag{4.3}$$

We then obtain a generic linear programming formulation of the revenue management problem

$$\text{LP}: \qquad \max_X \lambda \sum_{t \in T} \sum_{S \in 2^J} R(S) X_t(S)$$

subject to

$$\sum_{t \in T} \sum_{S \in 2^J} \lambda Q_i(S) X_t(S) \leq x_i \qquad \forall i \in I, \tag{4.4}$$

$$\sum_{S \in 2^J} X_t(S) \leq 1, \qquad \forall t \in T, \tag{4.5}$$

$$0 \leq X_t(S) \leq 1 \qquad \forall t \in T, S \in 2^J, \tag{4.6}$$

which can be adapted to any customer choice model, and has actually been considered, in an equivalent formulation by Liu and van Ryzin (2008) and Bront et al. (2009) in the specific multinomial logit case.

4.2.2 Embedding CRS rules

In practical applications, firms apply restrictions to the set of products offered. The following rules are standard :

– Each product belongs to a unique control class c on its constituent legs.

- A product can be offered if and only if its associated control class on all of its constituent leg(s) is open, *e.g.*, if a product uses more than one resource, it can be offered if and only if its associated control class be open on all its constituent legs.
- If a control class is open, all products belonging to that control class are offered.
- A lower (child) class can be open only if the higher (parent) classes on that leg are open.
- Closed control classes cannot be re-opened.
- We may have specific upper bounds on booking limits on legs.

Clearly, the optimal solution of CDLP might violate one or several of the above constraints. This leads to the extended models described below.

4.2.3 A disaggregate model

The constraints that are most difficult to embed are the logical constraints linking the opening of parent and child classes. To this aim, we introduce binary variables Z_{it}^c which specify whether or not the control class c on leg (resource) i at time t is open or not. We link the availability of products in the offer sets and opening or closing a control class by constraints (4.7)–(4.9) and we call them *CRS-state* constraints.

$$a_{ij}a_j^c X_t(S) \leq Z_{it}^c \qquad \forall t \in T, i \in I, c \in C, S \in 2^J, j \in S, \tag{4.7}$$

$$X_t(S) \leq \sum_{i \in I} a_{ij} - \sum_{c \in C, i \in I | (a_{ij}, a_j^c \neq 0)} Z_{it}^c \qquad \forall t \in T, S \in 2^J, j \notin S, \tag{4.8}$$

$$1 + \sum_{c \in C, i \in I | (a_{ij}, a_j^c \neq 0)} Z_{it}^c - \sum_i a_{ij} \leq \sum_{S \in 2^J | j \in S} X_t(S) \qquad \forall t \in T, j \in J, \tag{4.9}$$

The set of constraints (4.7) ensures that if a product j belonging to set S is going to be offered in a specific booking period t (*i.e.*, $a_{ij}a_j^c X_t(S) > 0$), its corresponding control class on all of its constituent legs should be open (*i.e.*, Z^c_{it} =1), and if a control class is closed (*i.e.*, Z^c_{it} =0), all of products belonging to that control class which are using that leg can not be offered (*i.e.*, $a_{ij}a^c_j X_t(S) = 0$).

The set of constraints (4.8) enforces that if a set S is offered at time t (*i.e.*, $X_t(S) > 0$), for a product j that is not present in S, its corresponding control class on at least one of its constituent legs should be closed. The difference between the number of its constituent legs and the number of open control classes corresponding to that product on its constituent legs should be strictly positive.

Finally, constraints (4.9) ensure that if a control class is open, all products belonging to that control class should be offered. For a product j, if the number of its corresponding open control classes on its constituent legs meets the number of legs which it actually uses, product j then should be present in one or more offered sets with positive value at time t.

Next, constraints (4.10)–(4.11) implement the non-opening rules over the booking horizon, together with the relationships between parent and child control classes.

$$Z^c_{it+1} \leq Z^c_{it} \qquad \forall i \in I, c \in C, t \in \{1, ..., |T| - 1\} \tag{4.10}$$

$$Z^{c+1}_{it} \leq Z^c_{it} \qquad \forall i \in I, t \in T, c \in \{1, ..., |C| - 1\} \tag{4.11}$$

Constraints (4.10) ensure that if a control class is closed at a booking period t, it can not be re-opened at any future booking period. We call constraints (4.10) along with state constraints (4.7)–(4.9) as *CRS-non-reopening* constraints. Constraints (4.11) ensure that if a control class is closed, all child classes with lower revenue are also closed. Similarly, we call constraints (4.11) along with state constraints (4.7)–(4.9) the *CRS-nesting-order* constraints.

Now, let $Q^c_i(S)$ denote the expected usage of products of control class c on resource i if the product set S is offered, b^c_i denote the assignment for control class c on leg i. Note that booking limit for each control class can be computed by summing its assignments and those of its child control classes. The set of profit-maximizing offer sets and booking limits consistent with the CRS rules is then obtained by solving the following disaggregate nested booking limit (DNBL) program :

DNBL :

$$\max_{X,b,Z} \sum_{t\in T} \sum_{S\in 2^J} \lambda R(S) X_t(S) \tag{4.12}$$

subject to

$$\sum_{t\in T} \sum_{S\in 2^J} \lambda Q_i(S) X_t(S) \leq x_i \qquad \forall i \in I,$$

$$\sum_{S\in 2^J} X_t(S) \leq 1, \qquad \forall t \in T,$$

$$\sum_{t\in T} \sum_{S\in 2^J} Q_i^c(S) X_t(S) \leq b_i^c, \qquad \forall i \in I, c \in C,$$

$$X_t(S) \leq \sum_{i\in I} a_{ij} - \sum_{c\in C, i\in I|(a_{ij}, a_j^c \neq 0)} Z_{it}^c \qquad \forall t \in T, S \in 2^J, j \notin S,$$

$$1 + \sum_{c\in C, i\in I|(a_{ij}, a_j^c \neq 0)} Z_{it}^c - \sum_{i} a_{ij} \leq \sum_{S\in 2^J | j\in S} X_t(S) \qquad \forall t \in T, j \in J,$$

$$a_{ij} a_j^c X_t(S) \leq Z_{it}^c \qquad \forall t \in T, i \in I, c \in C, S \in 2^J, j \in S,$$

$$Z_{it+1}^c \leq Z_{it}^c \qquad \forall i \in I, c \in C, t \in \{1, \ldots, |T|-1\},$$

$$Z_{it}^{c+1} \leq Z_{it}^c \qquad \forall i \in I, t \in T, c \in \{1, \ldots, |C|-1\},$$

$$0 \leq b_i^c \leq \mu_i^c \qquad \forall i \in I, c \in C$$

$$0 \leq X_t(S) \leq 1 \qquad \forall t \in T, S \in 2^J,$$

$$Z_{it}^c \in \{0, 1\} \qquad \forall t \in T, i \in I, c \in C,$$

Note that, besides the value of booking limits, the solution of the above mixed integer linear program yields the exact periods when specific control classes are open, together with the corresponding sets of products.

4.2.4 An aggregate model

The size of Program DNBL , which is proportional to the number of booking periods, precludes its solution on anything but very small instances. In order to derive a more tractable, we aggregate booking periods in the set of aggregated booking periods H. More specifically, the horizon is partitioned into $|H|$ sub-horizons. Any sub-horizon h, is an aggregation of several consecutive booking periods and, at the end of any booking sub-horizon h, one or more control classes are closed. Slightly abusing notation, we now let $X_h(S)$ denote the number of periods in which the offer set S in the booking sub-horizon h is offered. This value will be the length of the sub-horizon h. The structure of the model is then in line, modulo the CRS compatibility constraints, with the formulation of Bront et al. (2009). The modified mathematical program ANBL is as follows :

ANBL :

$$\max_{X,b,Z} \sum_{h \in H} \sum_{S \in 2^J} \lambda R(S) X_h(S) \tag{4.13}$$

subject to

$$\sum_{h \in H} \sum_{S \in 2^J} \lambda Q_i(S) X_h(S) \leq x_i \qquad \forall i \in I,$$

$$\sum_{S \in 2^J} X_h(S) \leq |T|, \qquad \forall h \in H,$$

$$\sum_{h \in H} \sum_{S \in 2^J} Q_i^c(S) X_h(S) \leq b_i^c, \qquad \forall i \in I, c \in C,$$

$$X_h(S) \leq \Big(\sum_{i \in I} a_{ij} - \sum_{c \in C, i \in I | (a_{ij}, a_j^c \neq 0)} Z_{ih}^c\Big) M, \qquad \forall h \in H, S \in 2^J, j \notin S,$$

$$1 + \sum_{c \in C, i \in I | (a_{ij}, a_j^c \neq 0)} Z_{ih}^c - \sum_i a_{ij} \leq \sum_{S \in 2^J | j \in S} X_h(S), \qquad \forall h \in H, j \in J,$$

$$a_{ij} a_j^c X_h(S) \leq M Z_{ih}^c, \qquad \forall h \in H, i \in I, c \in C, S \in 2^J, \; j \in S,$$

$$Z_{ih+1}^c \leq Z_{ih}^c, \qquad \forall i \in I, c \in C, \; h \in \{1, \dots, |H| - 1\},$$

$$Z_{ih}^{c+1} \leq Z_{ih}^c, \qquad \forall i \in I, h \in H, \; c \in \{1, \dots, |C| - 1\},$$

$$0 \leq b_i^c \leq \mu_i^c, \qquad \forall i \in I, c \in C,$$

$$0 \leq X_h(S), \qquad \forall h \in H, S \in 2^J,$$

$$Z_{ih}^c \in \{0, 1\}, \qquad \forall h \in H, i \in I, c \in C,$$

In the above, a suitable value for the big-M constant M is simply given by $\max\{X_h(S)\}$, which is unknown but less or equal to $|T|$. In the next section, we present a solution approach that can be adapted to either ANBL or DNBL. The performance of both models will be appraised in the numerical examples.

4.3 Solution approaches

Model (ANBL) has an exponential number of variables, which corresponds to the $2^{|J|} - 1$ non-empty possible combinations of products to offer. Small instances can be solved by commercial MIP solvers. Actually, filtering techniques may help to reduce problem size by focusing on offer sets compatible with the CRS rules. For addressing large instances, we implement a column generation scheme.

4.3.1 Offline filtering

As mentioned before, the solution of the model CDLP is not necessarily feasible under CRS rules. For instance, for the 16-product problem considered in Section (4.4.2), only 242 are CRS-compatible, out of $2^{16} - 1 = 65535$ combinations. To check compatibility, let us introduce the variables z_i^c, which indicate whether control class c is open on leg i or not. Then, we can ensure that if a product j is going to be included in the offer set S, the corresponding control class on that resource should be open. Otherwise, if a control class is closed, all products belonging to that control class that use the resource are not offered. This constraint is mathematically expressed as :

$$a_{ij} a_j^c \leq z_i^c, \qquad \forall i \in I, c \in C, j \in S. \tag{4.14}$$

A product should not be offered in the offer set S, if its corresponding control class on constituent resource is closed. This can be verified by the following constraint :

$$1 + \sum_{c \in C, i \in I | (a_{ij}, a_j^c \neq 0)} z_i^c - \sum_{i \in I} a_{ij} \leq 0, \qquad \forall j \notin S. \tag{4.15}$$

Finally, in the offer set S, a child class can not be open if its parent class is closed. This is verified by the following constraint :

$$z_i^{c+1} \leq z_i^c, \qquad \forall i \in I, c \in C \tag{4.16}$$

Considering the constraints mentioned above, the feasibility of a set S under the CRS rules can be verified by solving the mathematical program

$$\max_{z} 0 \tag{4.17}$$

subject to

$$a_{ij} a_j^c \leq z_i^c, \qquad \forall i \in I, c \in C, j \in S,$$

$$z_i^{c+1} \leq z_i^c, \qquad \forall i \in I, c \in C,$$

$$1 + \sum_{c \in C, i \in I | (a_{ij}, a_j^c \neq 0)} z_i^c - \sum_{i \in I} a_{ij} \leq 0, \qquad \forall j \notin S.$$

Note that additional constraints based on the CRS rules could easily be appended to the model, if required.

4.3.2 Column generation-based heuristic

To obtain an exact solution of the practical (ANBL) model, which is a large-scale mixed integer problem, we have to develop a branch-and-price algorithm. In terms of processing time, this is very costly approach and in the practical problems, we mostly need a solution with a good quality in an acceptable processing time. In this section, we develop a column generation-based heuristic approach to solve model (ANBL) following with detailed steps of the algorithm.

In the column generation algorithm, we start by solving a reduced linear problem (RLP) ; that involves a limited number of columns. Then we construct a subproblem by using the dual solution of RLP, to find a column with positive reduced cost. Then we add this column to the RLP and solve it again. We continue these steps until we could not find any new column with a positive reduced cost, then the current solution is optimal (Bront et al. (2009)).

The RLP for the model (ANBL) is presented as follows. Let Ω denote the subset of products compatible with CRS rules. For simplicity, we consider there is no upper bound imposed on booking limits of resources.

To solve model (ANBL) with column generation algorithm, we have to deal with both

CRS-non-reopening and CRS-nesting-order constraints with binary variables. Since at any step of the column generation, the sub problem generates a unique subset of products, we can modify the subproblem to only generate subsets compatible with CRS-nesting-order constraints. However, to consider the CRS-non-reopening constraints during the booking horizon, we have to keep them in the master problem. So, the RLP will be as follows :

RLP :

$$\max_{X,b,Z} \sum_{h\in H}\sum_{S\in\Omega} \lambda R(S) X_h(S) \tag{4.18}$$

subject to

$$\sum_{h\in H}\sum_{S\in\Omega} \lambda Q_i(S) X_h(S) \leq x_i \qquad \forall i \in I,$$

$$\sum_{S\in\Omega} X_h(S) \leq |T|, \qquad \forall h \in H,$$

$$\sum_{h\in H}\sum_{S\in\Omega} Q_i^c(S) X_h(S) \leq b_i^c, \qquad \forall i \in I, c \in C,$$

$$X_h(S) \leq (\sum_{i\in I} a_{ij} - \sum_{c\in C, i\in I|(a_{ij}, a_j^c \neq 0)} Z_{ih}^c) M, \qquad \forall h \in H, S \in \Omega, j \notin S,$$

$$1 + \sum_{c\in C, i\in I|(a_{ij}, a_j^c \neq 0)} Z_{ih}^c - \sum_i a_{ij} \leq \sum_{S\in\Omega|j\in S} X_h(S), \qquad \forall h \in H, j \in J,$$

$$a_{ij} a_j^c X_h(S) \leq M Z_{ih}^c, \qquad \forall h \in H, i \in I, c \in C, S \in \Omega, j \in S,$$

$$Z_{ih+1}^c \leq Z_{ih}^c, \qquad \forall i \in I, c \in C, h \in \{1, \ldots, |H|-1\},$$

$$0 \leq X_h(S), \qquad \forall h \in H, S \in \Omega,$$

$$Z_{ih}^c \in \{0,1\}, \qquad \forall h \in H, i \in I, c \in C,$$

To tackle this problem, we develop the following heuristic approach :

Algorithm CG

Step 1 (initialization)

- Let Ω be a limited subset of CRS compatible sets.
- Let ζ be the set of all binary variables.
- Let Φ be an empty set of binary variables with specified fixed values.

Step 2 (variable fixing)

Fix the binary variables in Φ to their specified values.

Step 3 (solving RLP)

Solve RLP as a linear model with given Ω and Φ, and extract the dual variables.

Step 4 (solving subproblem)

Construct subproblem with obtained dual variables and generate a new CRS-nesting-order compatible subset.

Step 5 (stopping criterion)

– If there is no improvement in the value of objective function, then stop.

– If there is no new subset with positive reduced cost, then goto Step 8.

Step 6 (updating RLP)

Add new subset to set Ω.

Step 7 (loop)

Return to Step 2.

Step 8 (re-introducing CRS-non-reopening)

Solve RLP with all binary variables in ζ as a MIP.

Step 9 (updating Φ)

add all binary variables in ζ to Φ with their values of MIP solution.

Step 10 (loop)

Return to Step 2.

4.3.3 Subproblem formulation

Let π and σ be the dual prices of the capacity and time constraints of RLP, respectively. Then, a column with the most positive reduced cost can be obtained by solving the following subproblem.

$$\max_{S \in 2^J} R(S) - \sum_{i \in I} \pi_i Q_i(S) - \sum_{h \in H} \sigma_h. \tag{4.19}$$

The order of the products in the customer's OPL is playing the crucial role in our choice model. Let us introduce binary variable $y_{j_k^l}$ in the terms of the rank of the products, that assumes value 1 if the product ranked k in OPL l belongs to S, and to 0 otherwise. Note that j_k^l represents a unique index $j = j_k^l$. If a product in rank k from OPL l has been selected and activated, the rest of products with ranks higher than k should be inactive. (For more details see Hosseinalifam et al. (2014)).

To do so, we introduce binary variables w_k^l that denote the rank of the customer's most preferred available product. Once a product in rank $\bar{k}$ of OPL l is selected, we set variable w_k^l to value 1 for all $k > \bar{k}$.

Now, we can control the activation state of any product in the OPL l, by the product $y_{j_k^l}(1 - w_k^l)$. *i.e.,* the highest ranked available product is selected, while the remaining ones are not considered any more.

The definition of the variables w_k^l, together with following constraints (4.20) and (4.21),

$$y_{j_k^l} \leq w_{k+1}^l, \qquad \forall l \in L, k \in \{1, \dots, |O^l| - 1\}, \tag{4.20}$$

$$w_k^l \leq w_{k+1}^l, \qquad \forall l \in L, k \in \{1, \dots, |O^l| - 1\}, \tag{4.21}$$

ensures that any arriving customer purchases his highest ranked product within S.

Next, we need to check the feasibility of generated offer set under CRS-state and CRS-nesting-order rules. This is achieved by adding the following complementary constraints to the sub problem.

$$a_{ij} a_j^c y_j \leq z_i^c \qquad \forall i \in I, j \in J, c \in C. \tag{4.22}$$

Constraints (4.22) ensures that if a product is going to be included in the offer set, constructed by the subproblem, its corresponding control class c on leg i is open, and if a control class is closed, all products belonging to that control class could not be offered any more.

Constraints (4.23) ensure that if a control class is open, all products belonging to that control class should be offered and vice versa.

$$1 + \sum_{i \in I, c \in C} z_i^c - \sum_i a_{ij} \leq y_j \qquad \forall j \in J. \tag{4.23}$$

Finally, following constraints are to validate CRS-nesting-order rule. They ensure that if a control class is closed, all its child control classes should be closed. And, if a control class is open, all of its parent control classes should be open as well.

$$z_i^{c+1} \leq z_i^c \qquad \forall i \in I, j \in J, c \in C. \tag{4.24}$$

Recalling that P_{kl}^{++} denotes the total transition probability from the customer's most preferred product to the one in rank k in the OPL l, and re-stating binary variable y_j in the terms of order of products in the OPLs, $y_{j_k^l}$, the column generation subproblem can be formulated as follows. For notation simplicity, for the indicator parameters like a_{ij} and a_j^c we let $j = j_k^l$ be the unique index for product j located in rank k of OPL l.

$$\begin{aligned}
&\text{SUB}: \\
&\quad \max_{y,w,z} \sum_{l \in L} \sum_{k=1}^{|O^l|} P_{kl}^{++} P_l (r_j - \sum_{i \in I} a_{ij}\pi_i) y_{j_k^l} (1 - w_k^l) \\
&\quad \text{subject to} \\
&\quad y_{j_k^l} \leq w_{k+1}^l, && \forall l \in L, k \in \{1, \ldots, |O^l| - 1\}, \\
&\quad w_k^l \leq w_{k+1}^l, && \forall l \in L, k \in \{1, \ldots, |O^l| - 1\}, \\
&\quad z_i^{c+1} \leq z_i^c && \forall i \in I, c \in \{1, \ldots, |C| - 1\}, \\
&\quad a_{ij} a_j^c y_{j_k^l} \leq z_i^c && \forall i \in I, \forall l \in L, k \in \{1, \ldots, |O^l|\}, \\
&\quad 1 + \sum_{i \in I, c \in C} z_i^c - \sum_i a_{ij} \leq y_{j_k^l} && \forall l \in L, k \in \{1, \ldots, |O^l|\}, \\
&\quad y_{j_k^l}, w_k^l, z_i^c \in \{0, 1\} && \forall i \in I, c \in C, l \in L, k \in \{1, \ldots, |O^l|\}.
\end{aligned} \tag{4.25}$$

The bilinear terms $y_{j_k^l} w_k^l$ can be linearized through the introduction of variables $u_k^l = y_{j_k^l} w_k^l$ and additional inequalities $u_k^l \le y_{j_k^l}$, $u_k^l \le w_k^l$, $u_k^l \ge y_{j_k^l} + w_k^l - 1$. This yields the equivalent mixed integer formulation :

SUB′

$$\begin{aligned}
&\max_{y,w,z} \sum_{l\in L} \sum_{k=1}^{|O^l|} P_{kl}^{++} P_l (r_j - \sum_{i\in I} a_{ij}\pi_i)(y_{j_k^l} - u_k^l) \\
&\text{subject to} && (4.26)\\
&y_{j_k^l} \le w_{k+1}^l, && \forall l \in L, k \in \{1,\ldots,|O^l|-1\},\\
&w_k^l \le w_{k+1}^l, && \forall l \in L, k \in \{1,\ldots,|O^l|-1\},\\
&u_k^l \le w_k^l && \forall l \in L, k \in \{1,\ldots,|O^l|\},\\
&u_k^l \le y_{j_k^l} && \forall l \in L, k \in \{1,\ldots,|O^l|\},\\
&z_i^{c+1} \le z_i^c && \forall i \in I, c \in \{1,\ldots,|C|-1\},\\
&a_{ij} a_j^c y_{j_k^l} \le z_i^c && \forall i \in I, \forall l \in L, k \in \{1,\ldots,|O^l|\},\\
&u_k^l \ge y_{j_k^l} + w_k^l - 1 && \forall l \in L, k \in \{1,\ldots,|O^l|\},\\
&1 + \sum_{i\in I, c\in C} z_i^c - \sum_i a_{ij} \le y_{j_k^l} && \forall l \in L, k \in \{1,\ldots,|O^l|\},\\
&y_{j_k^l}, w_k^l, z_i^c \in \{0,1\} && \forall i \in I, c \in C, l \in L, k \in \{1,\ldots,|O^l|\}.
\end{aligned}$$

Model SUB', Even for large scale instances, can be solved efficiently by an off-the-shelf solver. This is mainly because of its compact formulation. Note also that the formulation is quite flexible and allows to incorporate additional constraints whenever required.

4.4 Numerical results

In this section, we evaluate the performance of the proposed models and algorithm on classic " Parallel Flights " and two "Hub and Spoke " networks with different sizes as benchmarks.

Moreover, we use Bertsimas-de Boer's simulation-based approach (Bertsimas and De Boer (2005)) to compare the performance of proposed approaches with one other well-known existing approach in the literature.

Bertsimas-de Boer algorithm is a simulation-based approach to compute booking limits over a network. It starts by an initial set of booking limits. Afterwards, using simulation to estimate the first order gradient of the revenue function, they develop a stochastic steepest ascent algorithm to improve the initial set of booking limits.

The computational results have been carried out on a computer with 2.4 GHz CPU and 4 GB of RAM and 2 cores. We have used the FICO Xpress-Mosel 7.2.1 to formulate and obtain the results of the column generation algorithm.

We do simulation to consider stochastic nature of demand in evaluation of different approaches. We generate 2000 streams of demand in the simulation process to compute the expected revenue obtained with the booking limits. At any booking period, we may have a customer from a segment l based on the arrival rate for that segment λ_l. The arriving customer will make a choice based on the order of products in his OPL and the availability based on the booking limits. The average of the computed results will denote the expected revenue obtained by the given assignment of resources and choice behaviour of segmented customers. All simulation results have a relative error of less than 0.5 % with 95% confidence.

4.4.1 Parallel flights

In this example (Chen and Homem-de Mello (2010)), we have a network with three parallel flight legs (morning, afternoon, evening) with initial capacities of 30, 50 and 40, respectively. A total number of six products are induced by these resources with two High (H) and Low (L) fare classes on each flight leg. Products' information and resource usage are shown in Figure 4.2 and Table 4.1.

Customers' OPL information are shown in Table 4.2. First and second columns denote the OPLs with the ordered products in them. Third and forth columns indicate transition

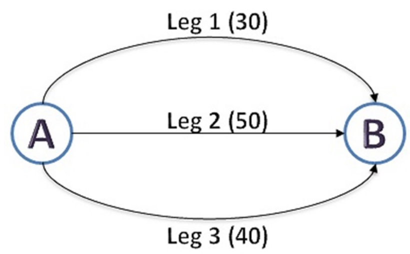

Figure 4.2 Parallel flight example : network.

Table 4.1 Parallel flight example : products

product	leg	class	fare
1	1	L	400
2	1	H	800
3	2	L	500
4	2	H	1000
5	3	L	300
6	3	H	600

probabilities and arrival rate for each OPL, respectively. We have four types of OPLs with corresponding transition probabilities. The booking horizon is divided into $|T| = 300$ periods with an average of 150 arrivals. To have a better evaluation of algorithms, we alter the demand level with the scale factor of α=0.75, 1 and 1.25. In this context, α=0.75 indicates that we multiply the supposed demand to 0.75 and we only let 75 % of demand come to the network.

Table 4.2 OPL setting for the parallel flights

OPL	ordered itineraries	transition probability	arrival rate
1	4, 2, 6	1, 0.5, 0.2	0.10
2	5, 1, 3	1, 0.5, 0.2	0.15
3	1, 2, 3, 4, 5, 6	1, 0.8, 0.75, 0.66, 0.75, 0.33	0.20
4	2, 1, 4, 3, 6, 5	1, 0.8, 0.75, 0.66, 0.75, 0.33	0.05

The total Expected Revenue (ER), Upper Bound (UB), and processing times (sec.) are presented in Table (4.3). Note that ER is the revenue obtained by the simulation, and UB is the value of the objective function of the exact formulation. The third and fourth columns show the result of the two exact formulations DNBL and ANBL, respectively. The fifth and sixth columns are the solutions of the column generation-based heuristic on the models DNBL and ANBL, respectively. The column "Bertsimas" indicates the obtained revenue by simulation-based approach of Bertsimas-de Boer. We set the full capacity of the resources as initial booking value of booking limits for each resource for the Bertsimas-de Boer algorithm. The next column denotes the expected revenue obtained by simulation-based approach with the solution of the heuristic approach as an initial solution. Finally, the last column indicates the upper-bound without considering the CRS restrictions.

Table 4.3 comparison of the results of the parallel flights example

α		DNBL	ANBL	HDNBL	HANBL	Bertsimas.	HANBL+B.	No CRS
	UB	65574	65595	65574	65595	-	-	65595
0.75	*ER*	*65026*	***65026***	***65026***	***65026***	*62609*	***65026***	-
	CPU	45	< 1	18	< 1	260	41	-
	UB	83079	83160	83079	83160	-	-	83410
1	*ER*	*81541*	***81712***	*81551*	*81587*	*79585*	*81587*	-
	CPU	133	< 1	87	< 1	338	38	-
	UB	86558	86819	86389	86819	-	-	86819
1.25	*ER*	*86226*	***86360***	*86108*	*86357*	*86118*	*86302*	-
	CPU	342	< 1	253	< 1	522	84	-

As it can be expected, both exact approaches have better revenue performances comparing to heuristic approaches. In the exact formulations, aggregated formulation has an slightly better upper bound comparing to the disaggregated formulation, because of the flexible length of booking horizons and then its finer control on state of the control classes and offering products. In the terms of the processing times, aggregated formulation, has better performance comparing to all other approaches in both exact and heuristic approaches.

We also observe that Performance of the Bertsimas-de Boer algorithm, is dependent to its initial solution. If we start this algorithm with a good initial solution, processing time significantly reduces to obtain an improved revenue.

4.4.2 Hub and Spoke network I

This network is based on the networks illustrated in the Figure 4.3. In this example, we have five flight legs and three cities inducing the total number of sixteen products. The flight legs' initial capacities are x=(12, 8, 8, 8, 8). Customers are divided into nine overlapping segments. For all segments, three different scenarios of low, medium and high level of overlap have been considered. The booking horizon is divided to $|T| = 80$ small booking periods and the arrival rate λ has been altered by a scale factor $\alpha = 0.9, 1, 1.1$.

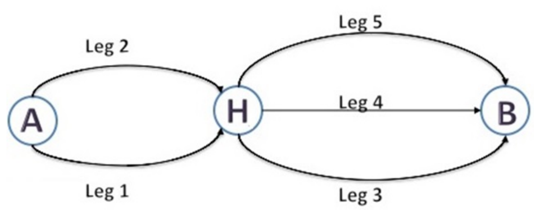

Figure 4.3 Hub and Spoke example I : network

Table 4.4 Hub and Spoke example I : products

product	leg	class	fare	product	leg	class	fare
1	1	L	400	9	5	L	400
2	1	H	800	10	5	H	800
3	2	L	300	11	1,4	L	500
4	2	H	600	12	1,4	H	1000
5	3	L	400	13	1,5	L	450
6	3	H	800	14	1,5	H	900
7	4	L	300	15	2,5	L	400
8	4	H	600	16	2,5	H	800

OPL informations for the cases with low, medium and high overlap are shown in Tables (4.5), (4.7) and (4.9) with corresponding transition probabilities and arrival rates in the third and fourth columns respectively. Afterwards, the following Tables (4.6), (4.8) and (4.10) denotes the obtained results by different approaches : low, medium and high overlap, respectively.

To obtain a good quality upper bound to evaluate the performance of the heuristic approach we implement the filtering approach. Number of the possible combinations is reduced from 65536 to only 242 feasible subsets under CRS rules by implementing proposed filtering approaches.

Table 4.5 OPL setting for the Hub and Spoke network I with low overlap

OPL	ordered itineraries	transition probability	arrival rate
1	2, 4	1, 0.5	0.07
2	4, 2	1, 0.5	0.05
3	1, 3	1, 0.8	0.15
4	6, 8, 10	1, 0.5, 0.2	0.07
5	10, 8, 6	1, 0.5, 0.2	0.05
6	7, 5, 9	1, 0.5, 1	0.15
7	12, 14	1, 0.5	0.07
8	16	1	0.05
9	15, 13, 11	1, 0.8, 0.7	0.15

Table 4.6 comparison of the different results of Hub and Spoke network I (low overlap)

α		DNBL	ANBL	HDNBL	HANBL	Bertsimas.	HANBL+B.	No CRS
	UB	23958	24103	23134	23271	-	-	24424
0.9	*ER*	*20924*	*21018*	*20967*	*21011*	*20902*	***21024***	-
	CPU	3600	35	212	11	45	35	-
	UB	25256	25302	21232	24499	-	-	25360
1	*ER*	*22106*	***22240***	*20723*	*22042*	*22034*	*22214*	-
	CPU	618	26	256	14	78	34	-
	UB	25762	25936	22501	25936	-	-	26011
1.1	*ER*	*22970*	***23257***	*21036*	*23168*	*22971*	*23142*	-
	CPU	3600	63	211	32	147	41	-

Filtering, in moderate size problems, enables us to solve the models with exact approaches to optimality with all possible feasible offer sets, when it is not possible to use exact ap-

Table 4.7 OPL setting for the Hub and Spoke example I with medium overlap

OPL	ordered itineraries	transition probability	arrival rate
1	2, 1, 4	1, 1, 0.5	0.07
2	4, 3, 2	1, 1, 0.5	0.05
3	1, 3	1, 0.8	0.15
4	6, 5, 8, 10	1, 1, 0.5, 0.2	0.07
5	10, 9, 8, 6	1, 1, 0.5, 0.2	0.05
6	7, 5, 9	1, 0.5, 1	0.15
7	12, 11, 14	1, 1, 0.5	0.07
8	16, 15	1,1	0.05
9	15, 13, 11	1, 0.8, 0.7	0.15

Table 4.8 comparison of the different results of Hub and Spoke network I (medium overlap)

α		DNBL	ANBL	HDNBL	HANBL	Bertsimas.	HANBL+B.	No CRS
	UB	23958	24103	23134	23271	-	-	24424
0.9	*ER*	*20909*	***21184***	*20987*	*21017*	*21070*	*21162*	-
	CPU	3600	34	151	7	163	103	-
	UB	25256	25302	24470	24499	-	-	25360
1	*ER*	*22169*	***22218***	*22025*	*22095*	*22204*	*22193*	-
	CPU	616	26	93	8	116	35	-
	UB	25762	25930	23364	25315	-	-	26011
1.1	*ER*	*23004*	***23218***	*22310*	*23217*	*23024*	*23204*	-
	CPU	1305	62	135	5	228	54	-

Table 4.9 OPL settings for Hub and Spoke network I with high overlap

OPL	ordered itineraries	transition probability	arrival rate
1	2, 1, 3, 4	1, 1, 0.7, 0.7	0.07
2	4, 3, 1, 2	1, 1, 0.7, 0.7	0.05
3	1, 3	1, 0.8	0.15
4	6, 5, 7, 8, 9, 10	1, 1, 0.7, 0.7, 0.6, 0.2	0.07
5	10, 9, 7, 8, 5, 6	1, 1, 0.7, 0.7, 0.6, 0.2	0.05
6	7, 5, 9	1, 0.5, 1	0.15
7	12, 11, 13, 14	1, 1, 0.7, 0.7	0.07
8	16, 15, 11, 12	1, 1, 0.7, 0.7	0.05
9	15, 13, 11	1, 0.8, 0.7	0.15

Table 4.10 comparison of the results of the Hub and Spoke network I with high overlap

α		DNBL	ANBL	HDNBL	HANBL	Bertsimas.	HANBL+B.	No CRS
	UB	23886	24103	23044	23230	-	-	24424
0.9	*ER*	*21062*	*21173*	*21006*	*21119*	***21208***	*21131*	-
	CPU	3600	32	162	7	247	47	-
	UB	25256	25302	23367	23567	-	-	25360
1	*ER*	*22171*	***22301***	*21800*	*21911*	*22181*	*22270*	-
	CPU	615	16	161	10	101	36	-
	UB	24318	25922	25142	25322	-	-	26008
1.1	*ER*	*23276*	***23385***	*23325*	*23357*	*23316*	*23360*	-
	CPU	3600	42	108	5	138	72	-

proaches with considering all possible offer sets. However, even after filtering, the exact formulation (DNBL) cannot finish in the one hour time limit all the time. Note that the filtering time has not been considered in the presented processing time.

This example, more precisely shows the time and revenue performance of aggregated formulation comparing to disaggregated one with a meaningful difference. It can be denoted that the column generation-based heuristic approach has better performance in aggregated formulation rather the disaggregated one. Like previous example, Bertsimas-de Boer algorithm with results of HANBL as initial solution has a very slight improvement which denotes high quality of HANBL's solution.

4.4.3 Hub and Spoke network II

In this example (Chen and Homem-de Mello (2010)), we consider a network (see Figure 4.4 and Table 4.11) with 7 flight legs inducing the total number of 22 products.

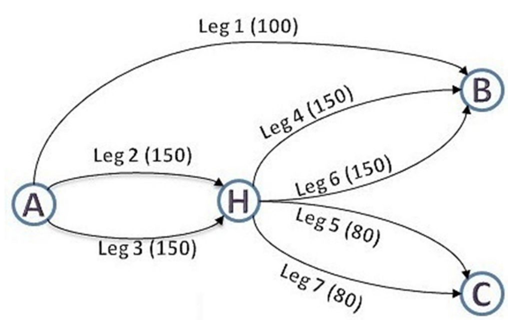

Figure 4.4 Hub and Spoke example II : network.

This network has a larger hub and spoke structure compared to the previous example and like before, we have some products using more than one resource. The resources have the initial capacities of x :=(100, 150, 150, 150, 150, 80, 80), respectively. The booking horizon consists of $|T| = 1000$ periods which with arrival rate $\lambda = 0.91$ we have the expected arrival rate of 910 customers per stream.

OPL information are shown in Table (4.12). We have ten OPLs with corresponding tran-

Table 4.11 Hub and Spoke example II : products

product	leg	class	fare	product	leg	class	fare
1	1	H	1000	12	1	L	500
2	2	H	400	13	2	L	200
3	3	H	400	14	3	L	200
4	4	H	300	15	4	L	150
5	5	H	300	16	5	L	150
6	6	H	500	27	6	L	250
7	7	H	500	28	7	L	250
8	{2,4}	H	600	19	{2,4}	L	300
9	{3,5}	H	600	20	{3,5}	L	300
10	{2,6}	H	700	21	{2,6}	L	350
11	{3,7}	H	700	22	{3,7}	L	350

Table 4.12 OPL settings for the Hub and Spoke example II.

OPL	ordered itineraries	transition probability	arrival rate
1	1, 8, 9, 12, 19, 20	1, 0.8, 1, 0.75, 0.66, 1	0.08
2	20, 19, 12, 9, 8, 1	1, 1, 0.8, 0.25, 1, 0.5	0.20
3	2, 3, 13, 14	1, 1, 0.5, 1	0.05
4	14, 13, 3, 2	1, 1, 0.2, 1	0.20
5	4, 5, 15, 16	1, 1, 0.5, 1	0.10
6	16, 15, 5, 4	1, 1, 0.2, 0.8	0.15
7	6, 7, 17, 18	1, 0.8, 0.625, 1	0.02
8	18, 17, 7, 6	1, 1, 0.2, 0.8	0.05
9	10, 11, 21, 22	1, 0.8, 0.625, 1	0.02
10	22, 21, 11, 10	1, 1, 0.2, 1	0.04

sition probabilities provided in the third column. We use a scale factor α to scale the arriving demand and evaluate the algorithm. The summary of the performance of different algorithms is shown in the Table (4.13).

This example more clearly denotes the differences between approaches. First, we should denote that for a problem with 22 products, it is possible to have more than 4 millions combinations of products to offer which makes it completely impossible to consider any commercial MIP solver to directly solve the problem.

In order to be able to have a good quality upper bound to evaluate the performance of the heuristic approach we implement the filtering approach. By implementing the filtering we reduce the number of the possible combinations from more that 4 millions to the only 2186 subsets which are possible to be offered under CRS rules in the nested booking policy. This filtering enables us to solve the exact formulation with all possible combinations with a time limit of maximum one hour.

Booking limits of the exact formulation DNBL, even by using the filtered data, cannot be computed any more. The large size of the problem does not let solver to be executed. Like the previous examples, for the initial solution for the Bertsimas-de Boer algorithm, we put all available capacity for the parent class. In this example, we have a meaningful increasing in processing time of Bertsimas-de Boer algorithm. This is mainly because of increasing the simulation time for a larger example compared to the previous ones. It can be clearly seen that HANBL still have much better performance in the terms of processing time and quality of the solution comparing to other approaches.

4.4.4 Summary of numerical results

The obtained results for the two exact approaches show that even though both models have almost the same revenue, there is a big difference in the processing time between them. This is mainly because of the aggregation of the booking periods in the model (ANBL) which makes problem smaller and easier to solve. Results show that in some cases we are unable to

Table 4.13 comparison of the different results of Hub and Spoke network II

α		DNBL	ANBL	HDNBL	HANBL	Bertsimas.	HANBL+B.	No CRS
	UB	-	224640	197815	218631	-	-	226766
0.75	*ER*	-	***216820***	*189914*	*214773*	*190695*	*216799*	-
	CPU	-	322	3600	23	1542	280	-
	UB	-	262250	203271	245365	-	-	267250
1	*ER*	-	***256952***	*192189*	*239526*	*228467*	*255281*	-
	CPU	-	650	3600	29	1672	484	-
	UB	-	297899	265116	288529	-	-	303516
1.25	*ER*	-	***265569***	*248096*	*258622*	*225293*	*260745*	-
	CPU	-	271	3600	12	1508	502	-

execute the model (DNBL) in 1 hour time limit, or even in the Example 3, Xpress cannot start because of the large size of the problem.

We remind that in both of the hub and spoke examples, we do preprocessing before implementing the exact approaches. This preprocessing consists of a filtering approach to distinguish subsets of products which can be offered under CRS rules. Regarding to the exponential number of possible combinations of products, filtering enables us to implement commercial solvers directly on our models.

Between two heuristic approaches also we can see a significant improvement in processing time in aggregated formulation. Besides the improvements in processing times, we obtain solutions with better quality in aggregated formulation.

The simulation-based approach has also a good performance ; however, since it is a local approach ; the quality of the obtained solution and specially the processing time is mainly dependent to the initial solution. Moreover, as it can be shown in the Hub and Spoke II Example, for a large-scale problems, the processing time with a large number of products, resources and booking periods the simulation-based approach by itself will take a long time to find a local solution. However, embedding the heuristic approach with improved simulation-based method will be a good candidate to have a good quality solution.

4.5 Conclusion

Our main contribution in this work is to propose a new mathematical programming approach to estimate better values of nested booking limits in a customer choice-based network revenue management problem. The proposed approach uses the strength of the model CDLP with a completely general choice-based model of demand to consider the network effect.

With the flexibility to take into account various CRS rules, the proposed approach computes the nested booking limits which generate the highest possible revenue. Besides the estimation of nested booking limits, the proposed approach provides offer sets corresponding to them. These data provide very useful information and has a vital rule for analyzers of the revenue management system.

In this paper, we proposed a new mathematical programming approach to estimate nested booking limits in a customer choice-based network revenue management problem. This approach not only provides the estimated values of nested booking limits, but also it provides the corresponding offer set to the estimate booking limits and opening and closing time for each control class on the different resources. This results can be widely used to implement as business rules in most of reservation systems.

We use a very general and powerful non-parametric choice model to estimate customer's choice behavior. Each arriving customer chooses from available alternatives according to an ordered preference list of products. In case of the non-availability of customer's preferred product, he substitutes it with a next product with a lower rank in his ordered preference list.

To solve the proposed model for the real world practical problems, a column generation algorithm has been developed to generate only set of products which are feasible under CRS rules. Our numerical results show that the proposed algorithm in most of the cases dominates other nested booking limit control policies. Moreover, in the terms of processing times, our algorithm is quite fast and in the most of the cases and does not need more than few minutes.

An appealing further work to this research could be considering hybrid nesting structure instead of general serial nesting. Another works could be improving algorithms and heuristics to aggregate a larger number of customers ordered preference lists (OPL) to increase efficiency of algorithms. Improving a joint optimization and simulation-based approaches could be another avenue for further researches.

CHAPTER 5

GENERAL DISCUSSION AND CONCLUSION

In this thesis, we have investigated novel approaches for the dynamic allocation of resources, in the context of network revenue management. These can be cast into two categories : bid price and booking limit control policies. In particular, we have shown how a column generation framework can accommodate real-life features, which are key to a successful implementation. Our numerical results are proof of the good performance of our algorithms, both in terms of actual revenue and processing time.

In chapter 2, we developed a joint seat allocation and bid pricing model that derives the value of time-dependent bid prices and the corresponding resource allocation in the customer choice-based revenue management framework.

In chapter 3, we proposed a nonparametric model for choice-based revenue maximization with corresponding algorithmic framework to solve practical large-scale problems. Finally, in chapter 4, we developed a new customer choice-based framework for computing nested booking limits that yield the highest possible return, within a given non-parametric customer choice environment.

Further research should focus on refining the choice model, for instance by considering hybrid, mixed logit environments. The success of our approach prompts us to include yet more features encountered in real world applications, mostly in the transportation industry (airlines, rail).

Finally, for the next generation of RM systems, researchers and practitioners will focus on most challenging features of these systems : modeling, forecasting and optimization. This involves more sophisticated systems to estimate precise customer choice behavior and implement efficient large scale optimization techniques to obtain better capacity control policies.

The presence of large amounts of detailed historical data and recent advances in big data analytics are also going to play a crucial role in the future of revenue management systems.

REFERENCES

Almogy, Y. and Levin, O. (1971). A class of fractional programming problems. *Operations Research*, 19 :57–67.

Belobaba, P. (1987). Air travel demand and airline seat inventory management. Technical report, Cambridge, MA : Flight Transportation Laboratory, Massachusetts Institute of Technology,[1987].

Belobaba, P. and Hopperstad, C. (1999). Boeing. MIT simulation study : PODS results update. In 1999 AGIFORS Reservations and Yield Management Study Group Symposium, April.

Bertsimas, D. and De Boer, S. (2005). Simulation-based booking limits for airline revenue management. *Operations Research*, 53(1) :90–106.

Bront, J., Méndez-Díaz, I., and Vulcano, G. (2009). A column generation algorithm for choice-based network revenue management. *Operations Research*, 57(3) :769–784.

Chaneton, J., Méndez-Díaz, I., Vulcano, G., and Zabala, P. (2010). Cdlp-based bid prices for network revenue management. *INFORMS Annual Meeting, Austin, TX.*

Chaneton, J. M. and Vulcano, G. (2011a). Computing bid prices for revenue management under customer choice behavior. *Manufacturing & Service Operations Management*, 13(4) :452–470.

Chaneton, J. M. and Vulcano, G. (2011b). Computing bid prices for revenue management under customer choice behavior. *Manufacturing & Service Operations Management*, 13(4) :452–470.

Chen, L. and Homem-de Mello, T. (2010). Mathematical programming models for revenue management under customer choice. *European Journal of Operational Research*, 203(2) :294–305.

Falk, J. and Palocsay, S. (1992). Optimizing the sum of linear fractional functions, in C.A. Floudas and P.M. Pardalos (eds.). *Recent Advances in Global Optimization*, pages 221–258.

Farias, V. F., Jagabathula, S., and Shah, D. (2011). Assortment optimization under general choice. *arXiv preprint arXiv :1108.3596*.

Farias, V. F., Jagabathula, S., and Shah, D. (2013). A nonparametric approach to modeling choice with limited data. *Management Science*, 59(2) :305–322.

Gallego, G., Iyengar, G., Phillips, R., and Dubey, A. (2004). Managing Flexible Products on a Network. Technical report, IEOR Department, Columbia University, Stanford University, School of Operations Research and Industrial Engineering, Cornell University.

Hosseinalifam, M. (2009). A fractional programming approach for choice-based network revenue management. Mémoire de maïtrise (Ecole polytechnique de Montréal).

Hosseinalifam, M., Marcotte, P., and Savard, G. (2014). Network capacity control under a nonparametric demand choice model. *preprint*.

Kunnumkal, S. and Topaloglu, H. (2008). A refined deterministic linear program for the network revenue management problem with customer choice behavior. *Naval Research Logistics*, 55(6) :563–580.

Littlewood, K. (1972). Forecasting and control of passengers. In *12th AGIFORS Symposium Proceedings*, volume 95, page 128.

Liu, Q. and Van Ryzin, G. (2008). On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management*, 10(2) :288–310.

Liu, Q. and van Ryzin, G. J. (2008). On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management*, 10(2) :288–310.

Mahajan, S. and van Ryzin, G. (2001). Stocking retail assortments under dynamic consumer substitution. *Operations Research*, 49(3) :334–351.

McFadden, D. (2000). Disaggregate behavioral travel demand's rum side. *Travel Behaviour Research*, pages 17–63.

Meissner, J. and Strauss, A. (2008). Network revenue management with inventory-sensitive bid prices and customer choice. *preprint.*

Meissner, J. and Strauss, A. (2012). Improved bid prices for choice-based network revenue management. *European Journal of Operational Research*, 217(2) :417–427.

Simpson, R. W. (1989). *Using network flow techniques to find shadow prices for market demands and seat inventory control.* MIT, Department of Aeronautics and Astronautics, Flight Transportation Laboratory.

Talluri, K. and van Ryzin, G. (1998). An Analysis of Bid-Price Controls for Network Revenue Management. *Management Science*, 44(11-Part-1) :1577–1593.

Talluri, K. and Van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1) :15–33.

Talluri, K. and Van Ryzin, G. (2005). *The theory and practice of revenue management*, volume 68. Springer.

Topaloglu, H. (2009). Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research*, 57(3) :637–649.

Train, K. (1986). *Qualitative choice analysis : Theory, econometrics, and an application to automobile demand*, volume 10. The MIT press.

van Ryzin, G. and Vulcano, G. (2008a). Computing virtual nesting controls for network revenue management under customer choice behavior. *Manufacturing & Service Operations Management*, 10(3) :448–467.

van Ryzin, G. and Vulcano, G. (2008b). Simulation-based optimization of virtual nesting controls for network revenue management. *Operations Research*, 56(4) :865–880.

van Ryzin, G. and Vulcano, G. (2011). An expectation-maximization algorithm to estimate a general class of non-parametric choice-models. *preprint.*

Williamson, E. L. (1992). Airline network seat inventory control : Methodologies and revenue impacts. *preprint*.

Zhang, D. and Adelman, D. (2009). An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science*, 43(3) :381–394.