

A column generation algorithm for choice-based network revenue management

Juan José Miranda Bront ^{*} Isabel Méndez-Díaz [†] Gustavo Vulcano [‡]

April 30, 2007

Abstract

In the last few years, there has been a trend to enrich traditional revenue management models built upon the independent demand paradigm by accounting for customer choice behavior. This extension involves both modeling and computational challenges.

One way to describe choice behavior is to assume that each customer belongs to a *segment*, which is characterized by a *consideration set*, i.e., a subset of the products provided by the firm that a customer views as options. Customers choose a particular product according to a multinomial-logit criterion, a model widely used in the marketing literature.

In this paper, we consider the choice-based, deterministic, linear programming model (CDLP) of Gallego et al. [6], and the follow-up dynamic programming (DP) decomposition heuristic of van Ryzin and Liu [16], and focus on the more general version of these models, where customers belong to overlapping segments. To solve the CDLP for real-size networks, we need to develop a column generation algorithm. We prove that the associated column generation subproblem is indeed NP-Complete, and propose a simple, greedy heuristic to overcome the complexity of an exact algorithm. Our computational results show that the heuristic is quite effective, and that the overall approach has good practical potential and leads to high quality solutions.

Key words: choice behavior, multinomial logit, capacity control, hyperbolic programming, integer programming.

^{*}Computer Science Department, University of Buenos Aires, Buenos Aires, Argentina, jmiranda@dc.uba.ar

[†]Computer Science Department, University of Buenos Aires, Buenos Aires, Argentina, imendez@dc.uba.ar

[‡]Stern School of Business, New York University, New York, NY, U.S.A., gvulcano@stern.nyu.edu.

1 Introduction

Capacity control-based revenue management (RM) involves controlling a fixed and perishable capacity of resources over a finite horizon, with the objective of maximizing revenues. Applications of RM include service industries like airlines, hotels, railways, cruises, etc. In this paper, we will use the terminology for the airline application as representative of the problem.

Traditionally, RM systems have been built upon the *independent demand model* assumption. This assumption views demand as a sequence of requests for products, which are insensitive to the capacity controls applied by the airline, and to market conditions like price offered by the competitors, frequency of departures, brand preference of the customers, etc (e.g. see Talluri and van Ryzin [13] for further details). Under the independent demand model assumption, it is also generally assumed that low fare demand comes first. Thus, the airline’s problem is posed as “how much capacity should be reserved for the high fare demand that will come later?”. There is wide agreement nowadays about the limitations of this assumption, based on the observation that the sale of a product is indeed the outcome of a customer’s purchase decision subject to market conditions. Furthermore, the expansion of low-cost airlines offering simplified, undifferentiated fare structures, and their usual strategy of saturating a market with several flights during the day, has raised the interest in formally capturing customer choice behavior in RM systems.

Accounting for customer choice behavior involves two main challenges. The first is to model the choice decision of a customer at a particular moment in time, and to estimate the parameters that describe that decision. The second is to incorporate this sophisticated demand model in the optimization module of a RM system, so that the availability controls explicitly account for customer choice behavior.

A very interesting piece of work on this second matter is the paper by Gallego et al. [6]. They propose a customer choice-based linear programming (CDLP) model for network RM, which is the choice-based analog of the deterministic linear programming (DLP) model of traditional RM. Although their formulation is for the case of *flexible product* offerings, where the firm has the flexibility of offering a collection of products to serve market demand, its application for choice-based RM is straightforward: It includes *offer sets*, i.e. sets of products offered by the seller at different points in time during the booking horizon, thus re-stating the problem as “which alternatives should the firm make available to the customers in order to profitably influence their choices?”. The decision variables are the lengths of time during which the different offer sets must be available during the booking horizon. As a limitation, their market demand model does not allow any kind of segmentation.

In a follow-up paper, van Ryzin and Liu [16] present two approaches to implement the outcome of the CDLP: In the first approach, the primal solution of the CDLP is applied right away, and the offer sets remain available for the length of time indicated by the values of the decision variables; and the other approach uses the optimal dual variables in a decomposition scheme that split the global network problem into a collection of computationally tractable leg-level problems. They present a market segmentation model, in which each customer belongs to one specific segment. In their case, the segments are defined by disjoint *consideration sets* of products. Their results show that the decomposition heuristic produces significant improvements in revenue compared to a direct application of the CDLP primal solution. They also compare the revenue gap of both approaches with respect to results obtained under the traditional independent demand assumption, and verify that the decomposition heuristic consistently outperforms the latter, while the performance of CDLP is sort of mixed.

Motivated by the appealing ideas of Gallego et al. [6], and the promising results of van Ryzin and Liu [16], in this paper, we consider the extension of the CDLP -and its derivations- to the general case of overlapping segments (i.e. segments whose corresponding consideration sets have a nonempty intersection). In Gallego

et al. [6], the absence of market segments eliminates the different preference weights that different customers could have for the same product. In van Ryzin and Liu [16], the restriction imposed in the numerical experiments is a strong one, since it partitions the space of products among different customer segments. For instance, consider a simple airline market with two parallel flights (say, one morning flight, and one afternoon flight), and two fare classes (high and low) per flight. One segment could be defined by time-sensitive customers, with strong morning preference, i.e. customers who prefer to pay the low fare, but are eventually willing to pay the high fare in the morning flight. Another segment could be defined by price-sensitive customers, that will go for the lowest available fare, no matter the time. Clearly, the preference order is different for these two types of customers (which is precluded in the model of Gallego et al. [6]), and the product “morning flight, low fare class” belongs simultaneously to both consideration sets (which is not allowed in the model of van Ryzin and Liu [16]). Certainly, allowing for overlapping segments would be a more natural fit for modeling real situations of market segmentation, and constitutes a strong limitation of the previous two papers. The cases known so far where the CDLP model and its derivations can be solved efficiently are very restrictive. In this paper, we present a method that closes the gap between the CDLP framework and its real practical potential.

As in van Ryzin and Liu [16], we also focus on the multinomial logit (MNL) choice model of demand, a model widely used in the marketing and economics literature. The main difficulty, from a computational standpoint, suffered by their approach is solving the CDLP efficiently by column generation. Indeed, it turns out that the column generation subproblem is difficult on its own. A structurally similar problem to the column generation one arises in the operational policy derived from the decomposition algorithm, when the firm needs to dynamically compute the next offer set to exhibit. As van Ryzin and Liu [16] show in Section 6.3, for the case of non-overlapping segments, this can be done in polynomial time. For the more general case of overlapping segments, the problem is of the fractional-programming type. Our theoretical contribution is a negative result in this regard: even the column generation subproblem is NP-Complete. The proof that we present here relies on a polynomial transformation from the graph theory problem *minimum vertex cover*, well-known to be NP-complete.

Being able to solve –or at least approximately solve– the column generation subproblem in an efficient way, has broader implications than the clear benefit that it would bring to our methodology. For example, in a marketing science framework, our column generation subproblem is equivalent to the problem faced by a seller that receives streams of customers from overlapping segments, and needs to decide the optimal assortment of products to offer in order to maximize the instantaneous revenue rate, when the product prices are fixed.

In order to provide a polynomial algorithm for the column generation subproblem, we use a greedy heuristic that performs quite well in practice, from its computational speed as well as quality of solutions obtained perspectives, finding optimal solutions in most of the cases that we considered.

The main practical contribution of our paper is illustrated by computational experiments. We run an exhaustive series of computational tests extending the decomposition heuristic of van Ryzin and Liu [16] to the overlapping segment case, and comparing its performance with CDLP, a randomized variation of CDLP, and a reoptimization scheme that uses elements of both the decomposition heuristic (DCOMP) and the primitive CDLP. Our results are somewhat aligned with the findings of van Ryzin and Liu [16]: DCOMP outperforms the other methods, especially in scarce capacity settings. However, the other methods show a good revenue performance, outperforming the behavior of the independent demand model assumption generally by more than 10%, indeed showing that choice behavior is a first order effect from a revenue management perspective.

Our experiments also show the practical feasibility of this approach by providing the computational times

of the different methods. The speed for solving the column generation subproblem gives a strong support for the potential of the choice-based methods. The decomposition heuristic is clearly the most time consuming method, taking around 8 to 10 minutes in a small to medium network setting (four nodes, seven legs, eleven products, ten customer segments). However, its core is building off-line a big two-dimensional table to be used later on when computing real time the offer sets. This is indeed a batch process that is typically run overnight by the airlines, and hence it is still practically feasible. Nevertheless, if time becomes an issue for larger networks, the computational times of the other methods are at least an order of magnitude shorter, hence constituting an interesting alternative considering the high quality of the solutions obtained.

The remainder of this paper is organized as follows: In Section 2, we review the related literature. In Section 3, we introduce the stochastic model, and the related choice-based linear programming formulation CDLP. Section 4 presents our approach for solving the column generation subproblem derived from CDLP. Our main theoretical result –the NP-Completeness of this subproblem– is included here. Section 5 describes the decomposition heuristic, and explains the difficulty of even solving each of the leg-level problems exactly in polynomial time. Our numerical results are reported in Section 6, and we conclude in Section 7.

2 Literature review

A complete description of traditional network RM models can be found in Talluri and van Ryzin [13, Chapter 3]. As was pointed out before, these methods typically work under the independent demand model paradigm, which is understood nowadays as a serious limitation for modeling customer behavior.

The interest in enriching these models with choice behavior finds its roots in the work by Belobaba and Hopperstad [1], who conduct simulation studies to understand the implications of passenger choice behavior with respect to the independent demand assumption.

On the theoretical side, Talluri and van Ryzin [14] provide an exact analysis of the optimal control policy for a single-leg RM model under a general discrete choice model of demand. They prove that the optimal policy is quite simple, and consists of opening sequentially sets from an ordered family of *efficient sets*. Zhang and Cooper [21] analyze choice among parallel flights in the same O-D market. Their model assumes that customers choose among the same fare class on different flights but not among fare classes. They develop bounds and approximations for the resulting dynamic program. Using a simulation-based optimization approach, but constrained to a parametric class of policies (in their case, virtual nesting controls), van Ryzin and Vulcano [17] report promising computational results for optimizing RM controls under a very general model of network choice behavior. The paper by van Ryzin [15] provides a very clear introduction to demand models for RM.

We describe customer choice behavior by using overlapping segments, where the preferences of each segment are described by a MNL model (e.g. see Ben-Akiva and Lerman [2] for a detailed description of the MNL choice model). The MNL belongs to the wide class of *random utility models*. This class models choice behavior assuming that individual customers' utilities for alternatives are random variables, and that customers are utility maximizers. Our demand model is, in fact, an example of the so-called *mixed multinomial logit* (MMNL) model, first introduced by Cardell and Dunbar [3]. The general MMNL model choice probabilities are defined by mixing a MNL model over a mixing distribution. McFadden and Train [10] establish the important result that, in theory, any random utility model can be captured by a correct specification of G , also providing support to the flexibility of our model in terms of its capability to represent more general customer choice behavior.

As mentioned above, our computational method builds upon the papers of Gallego et al. [6] and van Ryzin and Liu [16]. The recent work by Zhang and Adelman [20] is also related to that line of research.

They approximate the value function with an affine function of the state vector, but the column generation algorithm that they propose is also focused on the disjoint consideration set case.

3 Model

Our model set-up is similar to van Ryzin and Liu [16, Section 3]. Consider a network that has m resources or *legs* which can be used to provide n products (e.g. in an airline network each resource could correspond to a single-leg flight, and a product is defined by an itinerary and fare-class combination). The initial capacities are denoted by $c = (c_1, \dots, c_m)$. The set of products is denoted by $N = \{1, \dots, n\}$. Define the incidence matrix $A = [a_{ij}] \in \{0, 1\}^{m \times n}$. We let $a_{ij} = 1$ if resource i is used by product j , and $a_{ij} = 0$ otherwise. Thus, the j -th column of A , A_j , is the incidence vector for product j ; and the i -th row of A , A^i , is the incidence vector for resource i . We use the notation $i \in A_j$ to indicate that resource i is used by product j ; and $j \in A^i$ to mean that product j uses resource i . The state of the network is described by a vector $x = (x_1, \dots, x_m)$ of remaining resource capacities. If one unit of product j is sold, the state of the network changes to $x - A_j$. The revenue obtained from accepting a request for one unit of product j is r_j .

We consider discrete time periods which are indexed forward in time by t , $t = 1, \dots, T$, for a finite T . We assume that there is at most one customer arrival within each time period, requesting a single unit of a product. The probability of an arrival in each small time period is denoted by λ .

Customers belong to different market segments $l = 1, \dots, L$, and each segment is characterized by one *consideration set* $C_l \subset N$. Note that this is different from Gallego et al. [6], who just allow for a unique segment $C_1 = N$. On the other hand, the distinguishing feature of our model with respect to van Ryzin and Liu [16] is that we allow for overlapping segments. That is, we admit that $C_l \cap C_{l'} \neq \emptyset$ for $l \neq l'$. From the firm's perspective, each arriving customer belongs to segment l with probability p_l , with $\sum_{l=1}^L p_l = 1$. Hence, the arriving stream of segment- l customers is a Poisson process with rate $\lambda_l = \lambda p_l$, and the total arrival rate λ verifies $\lambda = \sum_{l=1}^L \lambda_l$.

The firm's decision is the set of products to make available in each period t . We call this subset $S \subset N$ of available (open) products, the firm's *offer set*. Given an offer set S , an arriving customer chooses product $j \in S$ with probability $P_j(S)$, where $P_j(S) = 0$ if $j \notin S$. We will denote the no-purchase probability by $P_0(S)$, and by total probability, we have that $\sum_{j \in S} P_j(S) + P_0(S) = 1$. As is generally the case in the choice behavior related RM literature, these probabilities will be based on the multinomial logit model (MNL). Under the MNL choice model, the choice probability of a segment- l customer is defined by a preference vector $v_l \geq 0$, that indicates the customer "preference weight" for each product contained in C_l . This vector, together with the no-purchase preference v_{l0} , determine a customer's choice probabilities as follows: If we let $P_{lj}(S)$ denote the probability that a customer from segment l chooses product $j \in C_l \cap S$ when S is offered, then,

$$P_{lj}(S) = \frac{v_{lj}}{\sum_{h \in C_l \cap S} v_{lh} + v_{l0}}$$

If $j \notin C_l \cap S$ or $j \notin C_l$, then $v_{lj} = 0$ (and hence $P_{lj}(S) = 0$). For mathematical tractability, we assume that $v_{l0} > 0$ for $l = 1, \dots, L$. In words, every arrival has a positive probability of not purchasing any product. Noting that from the seller's perspective the segment of a customer is not distinguishable, the probability that an arriving customer chooses product $j \in S$ is given by

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

We assume that the firm is risk neutral and seeks to maximize expected revenues. The firm's decision problem is to find a policy for choosing offer sets S at any time t during the booking horizon.

3.1 Dynamic programming formulation

Following van Ryzin and Liu [16], this problem can be formulated as a dynamic program (DP):

$$\begin{aligned} V_t(x) &= \max_{S \subset N} \left\{ \sum_{j \in S} \lambda P_j(S) (r_j + V_{t+1}(x - A_j)) + (\lambda P_0(S) + 1 - \lambda) V_{t+1}(x) \right\} \\ &= \max_{S \subset N} \left\{ \sum_{j \in S} \lambda P_j(S) (r_j - (V_{t+1}(x) - V_{t+1}(x - A_j))) \right\} + V_{t+1}(x), \end{aligned} \quad (2)$$

with boundary conditions:

$$\begin{aligned} V_t(0) &= 0, \quad t = 1, \dots, T \\ V_{T+1}(x) &= 0, \quad \forall x \geq 0 \end{aligned}$$

Unfortunately this is not manageable for most realistic networks because of the curse of dimensionality of the state space. In order to circumvent this complexity, the most widely used approach in network RM is to approximate the DP with a LP. This is the approach taken in Gallego et al. [6], and in van Ryzin and Liu [16], and it is also the one that we take here.

3.2 Linear programming formulation

Note that we can interpret $P_j(S)$ as the deterministic quantity of product j sold when the set S is offered and a customer arrives. Let $R(S)$ denote the expected revenue generated from an arriving customer when set S is offered, i.e.,

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

Given an arrival, let $Q_i(S)$ denote the conditional probability of using a unit of capacity on leg i when we offer set S . The vector of capacity consumption (conditional) probabilities is given by $Q(S) = AP(S)$, where $P(S) = (P_1(S), \dots, P_n(S))^T$ is the vector of purchase probabilities.

Since the firm knows the market segmentation, but *ex ante* does not know which set each arriving customer belongs to, it just picks a subset S from the set of all available products N . We include here the choice-based LP (corresponding to formulation (2) in van Ryzin and Liu [16]):

$$\begin{aligned} V^{CDLP} &= \max \sum_{S \subset N} \lambda R(S) t(S) \\ &\text{s.t.} \\ &\sum_{S \subset N} \lambda Q(S) t(S) \leq c \\ &\sum_{S \subset N} t(S) \leq T \\ &t(S) \geq 0, \quad \forall S \subset N \end{aligned} \quad (3)$$

Even though the airline must decide at any period t which set S to offer, from an aggregated, deterministic revenue perspective it only matters how many periods each set S is offered. The decision variables $t(S)$

represent the number of periods during which set S is offered. The first (m -dimensional) constraint is about capacity availability; the second (unidimensional) one is about time availability (i.e. length of the booking horizon).

The goodness of this approximation is supported by its asymptotic optimality, verified by van Ryzin and Liu [16, Section 4]. Namely, when capacity and demand are scaled up proportionately, the revenue obtained under CDLP converges to the optimal revenue under the exact DP formulation.

4 Solving the CDLP by column generation

Note that the CDLP (3) has $2^n - 1$ primal variables, corresponding to all the possible non-empty subsets of N . However, as suggested by Gallego et al. [6], we can use column generation techniques to try to circumvent the size of the problem. The sketch of the algorithm is as follows: We start with a limited number of columns (subsets S) and solve a reduced LP using only these columns. We then check if there is any column left out of the problem that has a positive reduced cost relative to the current dual prices of the reduced problem. If so, a positive reduced cost column is added and the LP is resolved. If there is no such positive reduced cost column, then the current solution is optimal.

The reduced primal problem corresponding to the CDLP (3), assuming that we initially consider only a limited number of linearly independent subsets (columns) $\mathcal{N} = \{S_1, \dots, S_k\}$, is described by

$$\begin{aligned} V^{CDLP-R} &= \max \sum_{S \subset \mathcal{N}} \lambda R(S) t(S) \\ &\text{s.t.} \\ &\sum_{S \subset \mathcal{N}} \lambda Q(S) t(S) \leq c \\ &\sum_{S \subset \mathcal{N}} t(S) \leq T \\ &t(S) \geq 0, \quad \forall S \subset \mathcal{N} \end{aligned} \quad (4)$$

Let $\pi \in \mathbf{R}^m$ and $\sigma \in \mathbf{R}$ be the dual prices for this reduced problem. We must check if there is any set $S \notin \mathcal{N}$ that has a positive reduced cost. This is achieved by solving the following column generation subproblem:

$$\max_{S \subset N} \{ \lambda R(S) - \lambda \pi^\top Q(S) \} - \sigma \quad (5)$$

We let the binary vector $y \in \{0, 1\}^n$ be the characteristic vector of set S . It indicates which products are offered at any point in time, so that if S is the set currently offered, $y_j = 1$ if $j \in S$, and $y_j = 0$ otherwise. We can then express (5) in terms of the binary variables y_j for the MNL choice model:

$$\max_{y \in \{0, 1\}^n} \left\{ \sum_{l=1}^L \lambda_l \frac{\sum_{j \in C_l} (r_j - A_j^\top \pi) v_{lj} y_j}{\sum_{i \in C_l} v_{li} y_i + v_{l0}} \right\} - \sigma \quad (6)$$

If the optimal function value for (6) is nonpositive, then π and σ are dual feasible, and our current solution to the reduced problem is in fact optimal for the original CDLP (3). Otherwise, the optimal solution S^* is the new column to add to our reduced formulation (4), and the iteration is repeated. Note that the assumption $v_l \geq 0, v_{l0} > 0, \forall l$, ensures that the optimization problem (6) is always well defined.

4.1 Complexity of the column generation subproblem

Unlike the problem in van Ryzin and Liu [16, Section 6.3.1.], problem (6) is not separable in the variables $y_j, 1 \leq j \leq n$, since each product j could belong simultaneously to two or more consideration sets, and so y_j

could appear in the denominator multiple times, for different segments l .

In fact, problem (6) is a hyperbolic (or fractional) 0–1 programming problem. The general version of hyperbolic binary programming, i.e.

$$\max_{y \in \{0,1\}^n} \sum_{l=1}^L \frac{a_{l0} + \sum_{j=1}^n a_{lj} y_j}{b_{l0} + \sum_{j=1}^n b_{lj} y_j}, \quad (7)$$

where all coefficients are positive, is known to be NP-Hard (see Prokopyev et al. [11]). However, our problem (6) is a particular case of (7), where the variables are linked with less degree of freedom in such a way that problem (7) cannot be reduced to it, even for values $L = 2$, $n \geq 2$. On the other extreme, when the segments are disjoint, recall that we know from van Ryzin and Liu [16, Section 6.3] that problem (6) is polynomial (e.g. see also Hammer and Rudeanu [8], and Hansen et al. [9]). The following theorem establishes that the NP-Hard problem *minimum vertex cover* (see Garey and Johnson [7, problem GT1]) can be reduced to (6), so that our problem (6) is also NP-Hard.

Theorem 1 *The 0–1 fractional programming problem*

$$\max_{y \in \{0,1\}^n} \left\{ \sum_{l=1}^L \lambda_l \frac{\sum_{j \in C_l} w_j v_{lj} y_j}{\sum_{i \in C_l} v_{li} y_i + v_{l0}} \right\} \quad (8)$$

where $w_j, v_{lj} > 0, l = 1, \dots, L, j = 1, \dots, n$, and $C_l \subset N$, is NP-Hard.

Proof. Given a connected graph $G = (V, E)$, with nodes $V = \{1, \dots, v\}, v \geq 2$, and arcs $E \subset \{(i, j) \in V \times V, i < j\}$, a *minimum vertex cover* of G is a subset $V' \subset V$ such that every arc in E is incident to at least one node in V' .

Let I be an instance of *minimum vertex cover*. We will construct an instance J of problem (8) corresponding to I . There are $v + 1$ binary decision variables y_1, \dots, y_v, y_{v+1} , and $v + |E|$ summands in the objective function, where the first v summands are:

$$\frac{2y_{v+1} + y_1}{y_{v+1} + y_1 + 1/v^2} + \frac{2y_{v+1} + y_2}{y_{v+1} + y_2 + 1/v^2} + \dots + \frac{2y_{v+1} + y_v}{y_{v+1} + y_v + 1/v^2}$$

and the last $|E|$ summands are of the form

$$\frac{y_i + y_j}{y_i + y_j + 1/v^2}, \quad \text{for every } (i, j) \in E$$

Note that we can recover formulation (8) by taking $n = v + 1$, $w_{v+1} = 2$, $w_j = 1, 1 \leq j \leq v$, and by defining $L = v + |E|$; $C_i = \{v+1, i\}, i = 1, \dots, v$, and $C_{v+j} = \{i, j\}, j = 1, \dots, |E|$, where the arcs are labeled following the lexicographic order. For $l = 1, \dots, L$, we also need $\lambda_l = 1$, $v_{l0} = 1/v^2$, and $v_{lj} = \mathbb{1}\{j \in V \cap C_l\}$.

The following facts hold:

1. An optimal solution of J must verify $y_{v+1} = 1$, since for all $i \in V$,

$$\frac{2 + y_i}{1 + y_i + 1/v^2} > \frac{y_i}{y_i + 1/v^2}, \quad \forall v \geq 2$$

2. For every arc $(i_0, j_0) \in E$, in the optimum of J , we cannot have $y_{i_0} = y_{j_0} = 0$. To see this, we argue by contradiction: Let S be an optimal solution of J , with $y_{i_0} = y_{j_0} = 0$, for $(i_0, j_0) \in E$, and let S' be the same as S , but with $y_{i_0} = 1$. The affected terms would be:

$$\frac{2y_{v+1} + y_{i_0}}{y_{v+1} + y_{i_0} + 1/v^2}, \quad \text{and} \quad \frac{y_{i_0} + y_j}{y_{i_0} + y_j + 1/v^2}, \quad \text{for all } j \text{ such that } (i_0, j) \in E.$$

For the first one, we decrease the value of the solution S' with respect to the value of the solution S by less than $1/2$ for $v \geq 2$ (recall that from fact 1 above, $y_{v+1} = 1$):

$$\frac{3v^2}{2v^2 + 1} - \frac{2v^2}{v^2 + 1} = -\frac{v^4 - v^2}{2v^4 + 3v^2 + 1} > -\frac{1}{2}$$

For the second one, we increase the value of the solution by at least $1/2$ for $v \geq 2$, for every arc in E that has i_0 as one incident node (recall that we have fixed $y_{j_0} = 0$):

$$\frac{1}{1 + 1/v^2} \geq \frac{1}{2}$$

So, we can increase the objective function value by setting $y_{i_0} = 1$, which is a contradiction.

3. We are left with showing that an optimal solution of J has the smallest possible number of variables y_i , $i = 1, \dots, v$, set at $y_i = 1$. Let S be an optimal solution of J , and S' another optimal solution of J , but with fewer variables y_i set at one, i.e. $|S| > |S'|$. We define $Value_1(S)$ and $Value_2(S)$ (respectively, $Value_1(S')$, $Value_2(S')$) as the partial sum of the first v terms of objective function value in (8) and the last $|E|$ terms when plugging in the corresponding values of the decision variables y .

Comparing $Value_1(S)$ with $Value_1(S')$, give us

$$\begin{aligned} Value_1(S) - Value_1(S') &= |S| \frac{3}{2 + 1/v^2} + (n - |S|) \frac{2}{1 + 1/v^2} - |S'| \frac{3}{2 + 1/v^2} - (n - |S'|) \frac{2}{1 + 1/v^2} \\ &= (|S'| - |S|) \frac{v^4 - v^2}{2v^4 + 3v^2 + 1} \\ &< -\frac{1}{4}, \quad \forall v \geq 2 \end{aligned}$$

Regarding the last $|E|$ terms, the difference between the value of the summand $(k, j) \in S$ and the value of the summand $(k, j) \in S'$ is at most

$$\frac{2}{2 + 1/v^2} - \frac{1}{1 + 1/v^2} = \frac{2v^2}{2v^2 + 1} - \frac{v^2}{v^2 + 1} = \frac{v^2}{2v^4 + 3v^2 + 1}$$

Since the number of arcs verifies $|E| \leq n(n - 1)/2$, then

$$\begin{aligned} Value_2(S) - Value_2(S') &\leq \frac{v(v - 1)}{2} \frac{v^2}{2v^4 + 3v^2 + 1} \\ &= \frac{v^4 - v^3}{4v^4 + 6v^2 + 2} \\ &< \frac{1}{4} \end{aligned}$$

So, $Value_1(S') + Value_2(S') > Value_1(S) + Value_2(S)$, which contradicts the fact that S is an optimal solution of instance J .

Hence, we have found a polynomial transformation from instance I to instance J , such that a solution of J implies a solution to I . \square

We can now state the following corollary:

Corollary 1 *The associated decision problem to the optimization problem (8), i.e., "Given a constant $K > 0$, is there an assignment of variables $y \in \{0, 1\}^n$ such that*

$$\max_{y \in \{0, 1\}^n} \left\{ \sum_{l=1}^L \lambda_l \frac{\sum_{j \in C_l} w_j v_{lj} y_j}{\sum_{i \in C_l} v_{li} y_i + v_{l0}} \right\} > K \quad ? \quad (9)$$

is NP-Complete.

Proof. The decision problem (9) is NP, since given an instance of YES, it clearly takes polynomial time to check that the inequality is satisfied. The NP-Hard feature holds from Theorem 1. \square .

Note that the equivalence between formulation (8) and our column generation problem (6) relies on the fact that $w_j = (r_j - A_j^\top \pi)$, and that $y_j = 0$ in the optimal solution to (6) whenever $w_j \leq 0$.

4.2 Approaches for solving the column generation subproblem

4.2.1 MIP formulation

The hyperbolic column generation problem can be reformulated as a MIP problem (e.g. see Prokopyev et al. [11]). By defining the variables

$$x_l = \frac{1}{\sum_{i \in C_l} v_i y_i + v_{l0}}, \quad l = 1, \dots, L, \quad (10)$$

the column generation problem (6) can be rewritten as:

$$\begin{aligned} \max \quad & \sum_{l=1}^L \sum_{j \in C_l} \lambda_l (r_j - A_j^\top \pi) v_{lj} y_j x_l \\ \text{s.t.} \quad & x_l v_{l0} + \sum_{i \in C_l} v_i y_i x_l = 1, \quad l = 1, \dots, L \\ & y_j \in \{0, 1\}, \quad j \in N, \\ & x_l \geq 0 \quad l = 1, \dots, L \end{aligned} \quad (11)$$

The nonlinear terms $y_i x_l$ can be linearized (see Wu [19]), and a linear mixed 0-1 formulation can be obtained. The linearization is based on the following observation: A polynomial mixed 0-1 term $z = xy$, where x is a continuous variable and y is a 0-1 variable, can be represented by the following linear inequalities: (1) $x - z \leq K - Ky$; (2) $z \leq x$; (3) $z \leq Ky$; and (4) $z \geq 0$, where K is a large number greater than x .

In the context of our problem, if we define variables $z_{li} = x_l y_i$, and apply this result, it is possible to obtain the next MIP formulation:

$$\begin{aligned} \max \quad & \sum_{l=1}^L \sum_{j \in C_l} \lambda_l (r_j - A_j^\top \pi) v_{lj} z_{lj} \\ \text{s.t.} \quad & x_l v_{l0} + \sum_{i \in C_l} v_i z_{li} = 1, \quad \forall l \\ & x_l - z_{li} \leq K - Ky_i, \quad \forall l, i \in C_l \\ & z_{li} \leq x_l, \quad \forall l, i \in C_l \\ & z_{li} \leq Ky_i, \quad \forall l, i \in C_l \\ & y_j \in \{0, 1\}, \quad x_l \geq 0, \quad z_{li} \geq 0 \end{aligned}$$

Since $x_l = \frac{1}{\sum_{i \in C_l} v_i y_i + v_{l0}}$, it is enough to take $K \geq 1/\underline{v}$, where $\underline{v} = \min\{v_{li} : i = 0, 1, \dots, n; l = 1, \dots, L\}$.

The advantage of this reformulation is that any MIP software package can be used to solve the hyperbolic 0-1 problem. In fact, according to our experience, standard built-in methods provided by MIP software packages work pretty fast for the cases that we considered. However, the complexity results presented above show that the column generation problem is a difficult optimization problem, and therefore an alternative approach guaranteed to run in polynomial time is reasonable to implement in order to obtain *quickly a good quality* solution.

4.2.2 Greedy heuristic

Seeking computational simplification, we implement a greedy, constructive heuristic. It starts from an empty set S , and progressively adds products to it according to the maximum marginal contribution to the current solution.¹

Heuristic for the column generation subproblem

1. For all products j such that $r_j - A_j^T \pi \leq 0$, set $y_j = 0$.
2. Let $S' \subset N$ be the set of products j with no assigned value for y_j .
3. Compute $j_1^* = \operatorname{argmax}_{j \in S'} \left\{ \sum_{l=1}^L \frac{(r_j - A_j^T \pi) v_{lj}}{v_{lj} + v_{l0}} \right\}$. Set $S := \{j_1^*\}$, $S' := S' - \{j_1^*\}$.

4. Repeat

(a) Compute

$$j^* := \operatorname{argmax}_{j \in S'} \left\{ \sum_{l=1}^L \lambda_l \frac{\sum_{i \in C_l \cap (S \cup \{j\})} (r_i - A_i^T \pi) v_{li}}{\sum_{i \in C_l \cap (S \cup \{j\})} v_{li} + v_{l0}} \right\}$$

(b) If $\operatorname{Value}(S \cup \{j^*\}) > \operatorname{Value}(S)$, then $S := S \cup \{j^*\}$, and $S' := S' - \{j^*\}$.

until S is not modified.

5. For all $j \in S$, set $y_j = 1$. For $j \notin S$, set $y_j = 0$.

The worst case complexity of this procedure is $O(n^2L)$, where in practice, L is typically smaller than n . According to our experience, this greedy heuristic works pretty fast, and drives towards the optimal solution in most of the cases that we considered. Nevertheless, we show here an example where it fails to give the optimum.

Our approach to solve the CDLP by column generation first uses the greedy heuristic in order to identify an entering column to the base. If this method does not find an entering column, then we use the exact MIP procedure. As we said above, when there is still a column to enter the base, in most of the cases, the heuristic finds it.

5 Decomposition approximation algorithm

In this section, we describe how we extend the decomposition method used in van Ryzin and Liu [16, Section 6.2] to the overlapping-segment case. Following them, the main idea is to decompose the network DP into a collection of single leg DPs, each of which is one-dimensional and therefore does not suffer from the curse of dimensionality.

In order to come up with an operational policy, there are two phases that the firm should undergo. The first phase is an off-line process; it consists of assessing the expected value of the remaining capacity. This value depends on the number of available seats in the network flights, and on the time-to-go in the booking horizon. The second phase is a real time process: in every period, the firm needs to compute, dynamically, the next offer set.

¹This heuristic was also used by Prokopyev et al. [11] as the construction phase of a GRASP to solve a general fractional programming problem.

5.1 Phase 1: Assessing the value of capacity

Let $\pi^* = (\pi_1^*, \dots, \pi_m^*)$ denote the optimal dual prices of (4) (and hence, of (3)), corresponding to the m capacity constraints. Consider the following approximation for the network value function (2) at leg i :

$$V_t(x) \approx \hat{V}_t^i(x_i) + \sum_{k \neq i} \pi_k^* x_k,$$

where $\hat{V}_t^i(x_i)$ is a dynamic (time dependent) and nonlinear approximation of the value of the remaining capacity of leg i , and the $\pi_k^* x_k$ are static (time independent) and linear approximations of the value of capacity of other legs. Thus, an estimate of the opportunity cost of product j when providing a dynamic approximation of the capacity value at leg i is

$$V_t(x) - V_t(x - A_j) \approx \begin{cases} \Delta \hat{V}_t^i(x_i) + \sum_{k \in A_j, k \neq i} \pi_k^* & \text{if } i \in A_j \\ \sum_{k \in A_j} \pi_k^* & \text{if } i \notin A_j, \end{cases}$$

where $\Delta \hat{V}_t^i(x_i) = \hat{V}_t^i(x_i) - \hat{V}_t^i(x_i - 1)$. Note that this can be written as

$$V_t(x) - V_t(x - A_j) \approx (\Delta \hat{V}_{t+1}^i(x_i) - \pi_i^*) \mathbb{1}\{i \in A_j\} + \sum_{k \in A_j} \pi_k^*.$$

Taking this approximation, and substituting it into the exact network DP (2), we get the following one-dimensional DP approximation for the network value at each leg $i = 1, \dots, m$:

$$\hat{V}_t^i(x_i) = \max_{S \subset N} \left\{ \sum_{j \in S} \lambda P_j(S) \left(r_j - (\Delta \hat{V}_{t+1}^i(x_i) - \pi_i^*) \mathbb{1}\{i \in A_j\} - \sum_{k \in A_j} \pi_k^* \right) \right\} + \hat{V}_{t+1}^i(x_i), \quad (12)$$

with boundary conditions:

$$\begin{aligned} \hat{V}_{T+1}^i(x_i) &= 0, \quad \forall x_i \geq 0 \\ \hat{V}_t^i(0) &= 0, \quad \forall t \end{aligned}$$

Each of these leg-level value functions constitutes a dynamic approximation of the network value function at time t , given that there are x_i units of capacity remaining. Therefore, we can average out these approximations, and provide the following network approximation for the true value function (2):

$$V_t(x) \approx \frac{1}{m} \sum_{i=1}^m \left(\hat{V}_t^i(x_i) + \sum_{k \neq i} \pi_k^* x_k \right) \quad (13)$$

The computation of the table $\Delta \hat{V}_t^i(x_i), i = 1, \dots, m; t = 1, \dots, T$, is the building block for providing, dynamically, the offer sets S . Typically, airlines compute this table overnight, in a batch process. This task could be weekly if we are far from the departure date of the flights, and daily during the last month in the booking horizon.

5.2 Phase 2: Computing the offer sets dynamically

Based on the assessed value of capacity, the real time operational policy consists of deciding the set of products to offer in the next time period.

From (13), we can compute an estimate for the marginal value of capacity on leg i as a convex combination of the dynamic approximation and the corresponding dual variable of the CDLP:

$$\Delta V_t^i(x) \approx \frac{1}{m} \Delta \hat{V}_t^i(x_i) + \frac{m-1}{m} \pi_i^*$$

This approximation is slightly different from the one in van Ryzin and Liu [16, Section 6.2], since it explicitly expresses the marginal value of capacity as a convex combination of the dynamic and the static estimates. Note that for big values of m (i.e. for networks with several legs), $\Delta V_t^i(x) \approx \pi_i^*$. One could alternatively consider other heuristic approximations, of the form

$$\Delta V_t^i(x) \approx \Delta \bar{V}_t^i(x) := \beta \Delta \hat{V}_t^i(x_i) + (1 - \beta) \pi_i^*, \quad \text{for } 0 \leq \beta \leq 1. \quad (14)$$

Fine tuning the parameter β in a real world application should deserve particular attention for an airline.

During the booking horizon, the firm should select a set S dynamically at each time t by solving

$$\max_{S \subset N} \left\{ \sum_{j \in S, j \text{ available}} \lambda P_j(S) (r_j - \Delta \bar{V}_t^\top(x) A_j) \right\}, \quad (15)$$

where $\Delta \bar{V}_t(x) = (\Delta \bar{V}_{t+1}^1(x), \dots, \Delta \bar{V}_{t+1}^m(x))$ is the vector of approximate displacement costs. Our candidate products to include in S are the available products $j \in N$, in the sense that they must satisfy $x_i > 0$, for all $i \in A_j$.

Note that problem (15) has the same structure as the one in (8), since it is also driven by the complexity of solving the hyperbolic 0–1 programming problem determined by the MNL probabilities $P_j(S)$. Hence, it is also NP-Complete, driving towards the need of experimenting with heuristics or approximation schemes, as the one discussed in Section 4.2.2.

Problem (15) is also particularly relevant to the marketing science literature related to customer-choice behavior. It represents the optimization problem faced by a seller that serves a multi-segment arriving stream of customers, who choose according to a MNL model. Prices are fixed, and represented by $p_j = r_j - \Delta \bar{V}_t^\top(x) A_j$ in equation (15). Here, the objective is maximizing the instantaneous revenue rate.

6 Computational results

In this section, we evaluate different methods based on the CDLP to solve in practice the network revenue management problem when customers belonging to overlapping segments choose according to the MNL model. We assess the convenience of each method based on the quality of the solutions obtained, and on the computational feasibility in terms of the time consumed.

Our computational experiments were conducted in a SUN UltraSparc III server (CPU of 1Ghz, RAM of 2GB, operating system SunOS 5.9). The algorithms were coded in C++, and compiled using *gcc* version 3.4.6. The code was linked to the Cplex 8.1 optimization routines.

We use Monte Carlo simulation to mimic customer choice behavior. The simulation of each stream of customers starts in period $t = 0$. Depending on the method used (to be explained below), we determine the next set S^* to offer. The probability of selling product $j \in S$ in a small time period is given by $\lambda P_j(S)$. To determine the product sold (if any), we generate a Unif[0,1] r.v. and use the discrete cumulative distribution for $j \in S$. If a product j is sold, we subtract one unit of capacity from the legs $i \in A_j$, and proceed to the next period where we show the new S^* , until reaching period T .

We tested the following methods:

1. *DCOMP*: This is the choice-based decomposition heuristic discussed in Section 5. First, we solve the CDLP in (3) by using the column generation algorithm. Then, we use the leg-level decomposition in (12), and build off-line the table $\hat{V}_t^i(x_i)$, for all $t = 1, \dots, T$, and for all $x_i = 1, \dots, c_i$. We do this for all legs $i = 1, \dots, m$. The marginal value of capacity is computed by setting $\beta = 1$ in (14). When

processing the arriving stream of customers, upon each arrival, we solve (15) and determine the next set S^* to offer.

2. *DCOMP-0.5*: This method is analogous to DCOMP, but setting $\beta = 0.5$ in equation (14).
3. *CDLP*: This method implements the primal solution of the CDLP formulation described in Section 3.2. Recall that the optimal solution to (3) indicates the total time to offer each set S . However, the sequence in which the sets are offered is, in principle, ambiguous. In this case, we offer the sets following the lexicographic order of the indices of the LP variables. Since the variables $t(S)$ in (3) could take fractional values, we round them to the nearest integer. We truncate the booking horizon at T in case sets are offered beyond T after rounding.
4. *RCDLP*: This is a randomized version of CDLP. Namely, we shuffle the order of the indices of the primal variables of the LP (3).²
5. *ROPT-freq*: In this case, we reoptimize the CDLP with a predetermined frequency, in order to get updated values for the dual variables. We then use those values in the dynamic calculation (15) of the offer sets, which corresponds to setting $\beta = 0$ in equation (14). The idea of this method is to provide static estimates of the marginal value of capacity, but update frequently enough such that the estimation accounts for a more accurate state of the network. The idea is to somehow capture the dynamic approximation of DCOMP, but at a lower computational cost since ROPT-freq does not require computation of any table like $V_t(x)$ beforehand.

We use values 0.1 and 0.01 for the parameter *freq*, representing the frequency of the reoptimization of the CDLP to get new dual variables. That is, we partition the booking horizon into 10 and 100 evenly split time slots, respectively. When reoptimizing (3), we use the remaining time, capacity, and estimated demand-to-come as input parameters.³

Note that the offer set remains the same within each time slot except when a product j belonging to the current offer set becomes unavailable. In that case, we need to recompute (15).

6. *INDEP*: This method implements the traditional independent demand model assumption, and serves as a benchmark for explicitly accounting for customer choice behavior. This method involves first solving a DLP, for which the mean demand of product j is set at $\lambda P_j(N) T$, for all $j \in N$. That is, we assume that all products are simultaneously offered. We next use the dual variables in a dynamic programming decomposition scheme as described in Talluri and van Ryzin [13, Chapter 3], which is equivalent to the decomposition algorithm of Section 5, but replacing $P_j(S)$ with $P_j(N)$, for all j and S .

In the tables below we report an upper bound (UB) on the optimal expected revenue, given by the objective function value of (3). The suboptimality gaps of the different methods are calculated with respect to DCOMP.

In each of the examples, we test different network load factors. We use a parameter α to scale all leg capacities, where $\alpha = 1$ corresponds to the original base case. We alter the preferences in the choice behavior by varying the no-purchase preference vector $v_0 = (v_{10}, \dots, v_{L0})$.

²Similar to what was observed in van Ryzin and Liu [16], the performance of CDLP and RCDLP is very similar, and it is not clear whether one dominates the other. For this reason, we only report revenues for CDLP.

³Given that for this method we obtain better results by reoptimizing more frequently, we only report the revenue generated by ROPT-0.01.

6.1 Example 1: Parallel Flights

This example is illustrated in Figure 1. It is based on a network with three parallel flights, and two fare classes on each flight (high (H), and low (L)), producing a total of six products. We assume there are four customer segments: customers that are willing to choose only among high-class products; customers that are willing to choose only among low-class products; customers that are considering all products, but with more preference for those in class H; and finally customers that are considering all products, but with more preference for those in class L. This is a generalization of the model studied by Zhang and Cooper [21], and also by van Ryzin and Liu [16, Section 7.1].

The table in Figure 1 and Table 1 show respectively the product and segment definitions. Leg 1 corresponds to a morning flight, leg 2 to an afternoon flight, and leg 3 to an evening flight. The initial capacities are $c = (30, 50, 40)$. The booking horizon consists of $T = 300$ time periods, with an average of 150 arrivals per stream of demand. We generate 2,000 streams of demand, and process each stream using all the methods mentioned above.

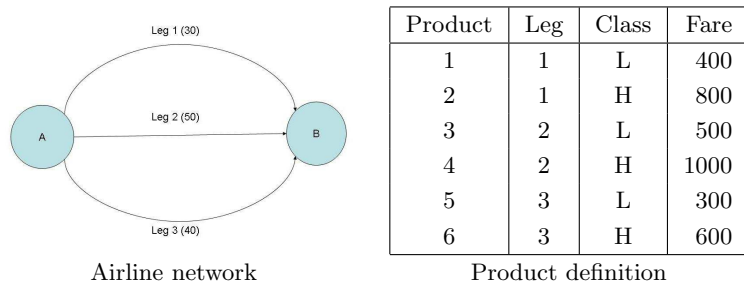


Figure 1: Parallel flights example

Segment	Consideration set	Pref. vector	λ_l	Description
1	{2,4,6}	(5,10,1)	0.1	Price insensitive, afternoon preference
2	{1,3,5}	(5,1,10)	0.15	Price sensitive, evening preference
3	{1,2,3,4,5,6}	(10,8,6,4,3,1)	0.2	Early preference, price sensitive
4	{1,2,3,4,5,6}	(8,10,4,6,1,3)	0.05	Price insensitive, early preference

Table 1: Segment definitions for Parallel Flights instance

Table 2 summarizes the expected revenues obtained under the different policies, and Table 3 summarizes the suboptimality gaps of the other methods with respect to DCOMP. For instance, the first row in the tables represents the case $c_{0.6} = 0.6 \times (30, 50, 40) = (18, 30, 24)$, and $v_0 = (1, 5, 5, 1)$, namely the weight for the no-purchase option for segments 1 and 4 is 1, and for segments 2 and 3 is 5.

We observe that the performance of DCOMP and DCOMP-0.5 is very similar, and that they both outperform CDLP in most cases, with confidence intervals for the revenue gaps with respect to DCOMP that exclude zero in 10 of the 15 cases. Differences are more significant in the high load factor cases (i.e. $\alpha = 0.6, 0.8$, and 1.0). The gains of the choice-based methods with respect to INDEP are generally quite significant.

For the ample capacity cases (say, $\alpha = 1.2$ and 1.4), all choice-based methods perform similarly in terms of expected revenues. In this example, we do not observe evidence of the risk of reoptimizing frequently (see Cooper [4], Cooper et al [5], and Secomandi [12] for discussions on the *spiral down effect*). In fact, given that $T = 300$, we are recomputing the dual variables once every three booking periods for the ROPT-0.01 case. Looking at the revenue differences achieved between ROPT-0.01 and ROPT-0.1, we could assess that

α	v_0	UB	DCOMP		DCOMP-0.5		CDLP		ROPT-0.01		INDEP	
			Mean	%LF	Mean	%LF	Mean	%LF	Mean	%LF	Mean	%LF
0.6	(1,5,5,1)	56,884	55,948	98.60	55,872	97.82	54,177	94.60	51,632	95.83	49,794	97.41
	(1,10,5,1)	56,848	55,882	98.36	55,698	97.29	54,051	94.45	51,946	93.81	49,655	96.77
	(5,20,10,5)	53,819	51,326	94.59	51,338	94.23	50,058	92.94	48,602	97.35	46,246	92.53
0.8	(1,5,5,1)	71,936	69,533	95.78	69,163	94.79	68,105	94.69	66,336	94.57	60,346	94.14
	(1,10,5,1)	71,794	69,129	95.24	68,863	94.50	67,896	94.58	66,237	93.82	59,552	91.40
	(5,20,10,5)	61,868	60,147	90.48	60,222	90.49	59,073	90.82	56,724	91.98	53,044	81.94
1.0	(1,5,5,1)	79,155	76,954	95.65	77,096	95.91	75,726	94.96	77,106	95.36	66,224	85.38
	(1,10,5,1)	76,866	75,639	90.88	75,655	90.97	73,788	90.26	75,399	91.26	64,831	81.76
	(5,20,10,5)	63,255	62,775	78.09	62,792	78.15	62,702	78.41	62,040	79.94	56,203	72.44
1.2	(1,5,5,1)	80,371	79,817	84.28	79,818	84.24	79,666	84.33	79,834	84.58	68,970	76.84
	(1,10,5,1)	78,045	77,520	79.06	77,526	79.07	77,348	79.26	77,529	79.49	67,570	73.71
	(5,20,10,5)	63,296	63,111	67.52	63,113	67.52	62,491	68.88	62,422	70.20	58,543	65.88
1.4	(1,5,5,1)	81,066	80,408	73.08	80,376	72.83	80,362	72.75	80,439	73.27	71,418	70.60
	(1,10,5,1)	78,816	78,123	68.56	78,097	68.32	78,091	68.24	78,136	68.80	69,949	67.87
	(5,20,10,5)	63,337	63,211	60.54	63,212	60.54	62,553	62.04	62,822	62.82	60,732	60.82

Table 2: Revenue results for the Parallel Flights example.

resolving the CDLP to obtain updated dual variables works very well, specially for the low to moderate load factor cases (i.e. $\alpha \geq 1.0$). On the other hand, for the highly constrained capacity cases, the combination between the dynamic and static components of DCOMP-0.5 more than compensates the frequent update of the dual variables.

α	v_0	% DCOMP-0.5	% CDLP	% ROPT-0.01	% INDEP
0.6	(1,5,5,1)	0.13 ± 0.18	3.16 ± 0.31	7.71 ± 0.24	10.99 ± 0.23
	(1,10,5,1)	0.32 ± 0.20	3.27 ± 0.32	7.04 ± 0.27	11.14 ± 0.23
	(5,20,10,5)	-0.02 ± 0.39	2.47 ± 0.41	5.30 ± 0.36	9.89 ± 0.37
0.8	(1,5,5,1)	0.53 ± 0.33	2.05 ± 0.33	4.59 ± 0.30	13.21 ± 0.33
	(1,10,5,1)	0.38 ± 0.35	1.78 ± 0.34	4.18 ± 0.32	13.85 ± 0.34
	(5,20,10,5)	-0.12 ± 0.47	1.78 ± 0.44	5.69 ± 0.42	11.80 ± 0.46
1.0	(1,5,5,1)	-0.18 ± 0.34	1.59 ± 0.33	-0.19 ± 0.34	13.94 ± 0.38
	(1,10,5,1)	-0.02 ± 0.42	2.44 ± 0.38	0.31 ± 0.23	14.28 ± 0.41
	(5,20,10,5)	-0.53 ± 0.56	0.11 ± 0.54	1.17 ± 0.54	10.46 ± 0.52
1.2	(1,5,5,1)	0.00 ± 0.46	0.18 ± 0.45	-0.02 ± 0.29	13.58 ± 0.43
	(1,10,5,1)	0.00 ± 0.48	0.22 ± 0.47	-0.01 ± 0.48	12.83 ± 0.45
	(5,20,10,5)	0.00 ± 0.57	0.98 ± 0.56	1.09 ± 0.57	7.23 ± 0.53
1.4	(1,5,5,1)	0.03 ± 0.48	0.05 ± 0.47	-0.03 ± 0.49	11.18 ± 0.45
	(1,10,5,1)	0.03 ± 0.50	0.04 ± 0.50	-0.01 ± 0.50	10.46 ± 0.46
	(5,20,10,5)	0.00 ± 0.56	1.04 ± 0.56	0.61 ± 0.56	3.92 ± 0.54

Table 3: 95% confidence intervals for the suboptimality gaps with respect to DCOMP in the Parallel Flights example.

The example also shows the impact of the customer preferences on the expected revenues. Specifically, for a given α , the revenues decrease with v_0 (i.e., the higher the values of the coordinates of v_0 , the higher the no-purchase likelihood of an arrival, and hence the lower the expected revenue).

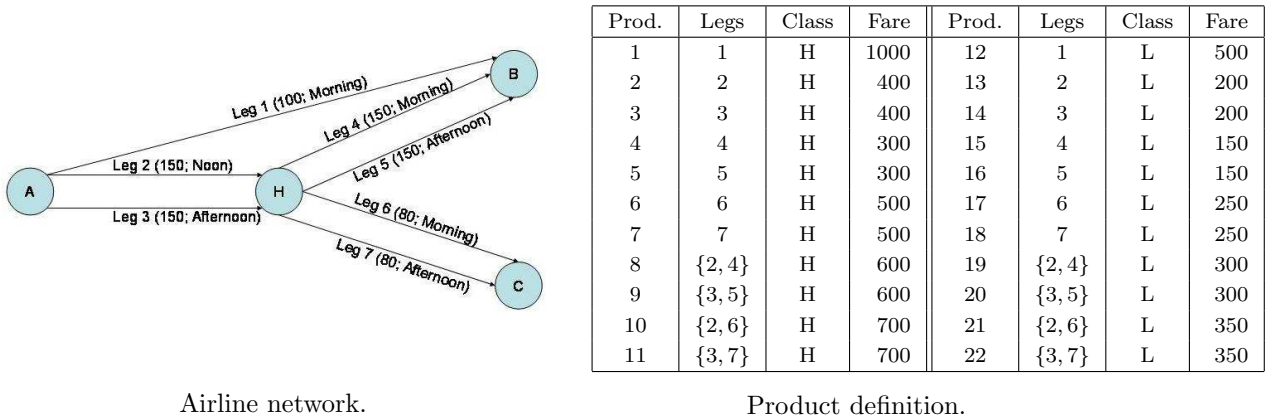
One important fact regarding the CDLP is that the column generation algorithm iterates on average just 5 times, when there are $2^6 - 1 = 63$ variables in the model. Recall that the approach that we follow first uses the heuristic described in Section 4.2.2 to find an entering column, and if it fails, then it calls the exact MIP procedure described in Section 4.2.1. In this case, the latter manages to find an entering column in 4.16% of the cases, meaning that the greedy heuristic provides good quality solutions more than 95% of the times.

6.2 Example 2: Small Network

This example is based on the same small network considered in van Ryzin and Liu [16, Section 7.2]. It consists of 7 flight legs with two fare classes per leg, and a total of 22 products. Figure 2 shows the network with the initial leg capacities and describes the products. There is local traffic, and also passengers connecting through the hub H . There are ten overlapping consideration sets, corresponding respectively to different customer segments. They are defined in terms of their origin-destination market, fare-class, and time preferences. Table 4 describes the ten customer segments.

To simplify notation, we have reduced the length of vector v_0 . In this case, $L = 10$, and a complete description of this vector consists of 10 coordinates. Since the segments can be classified in 5 pairs according to the O-D, we report just two coordinates for v_0 , assuming that the no-purchase preference weight is repeated for the different pairs. For example, $v_0 = (1, 5, 1, 5, 1, 5, 1, 5, 1, 5)$ is represented as $v_0 = (1, 5)$.

The booking horizon consists of $T = 1,000$ time periods, with an average of 910 arrivals per stream of demand. We generate 2,000 streams of demand per case (i.e., per combination (α, v_0)), and process each stream using all the methods mentioned above.



Airline network.

Product definition.

Figure 2: Network definition for the Small Network instance.

Segment	O-D	Consideration set	Pref. vector	λ_l	Description
1	$A \rightarrow B$	{1,8,9,12,19,20}	(10,8,8,6,4,4)	0.08	Price insensitive, early pref.
2	$A \rightarrow B$	{1,8,9,12,19,20}	(1,2,2,8,10,10)	0.2	Price sensitive
3	$A \rightarrow H$	{2,3,13,14}	(10,10,5,5)	0.05	Price insensitive
4	$A \rightarrow H$	{2,3,13,14}	(2,2,10,10)	0.2	Price sensitive
5	$H \rightarrow B$	{4,5,15,16}	(10,10,5,5)	0.1	Price insensitive
6	$H \rightarrow B$	{4,5,15,16}	(2,2,10,8)	0.15	Price sensitive, slight early pref.
7	$H \rightarrow C$	{6,7,17,18}	(10,8,5,5)	0.02	Price insensitive, slight early pref.
8	$H \rightarrow C$	{6,7,17,18}	(2,2,10,8)	0.05	Price sensitive
9	$A \rightarrow C$	{10,11,21,22}	(10,8,5,5)	0.02	Price insensitive, slight early pref.
10	$A \rightarrow C$	{10,11,21,22}	(2,2,10,10)	0.04	Price sensitive

Table 4: Segment definitions for the Small Network instance.

Table 5 summarizes the expected revenues obtained under the different policies, and Table 6 summarizes the suboptimality gaps of the other methods with respect to DCOMP. We observe again that DCOMP and DCOMP-0.5 exhibit a similar performance. We observe that DCOMP outperforms CDLP in nine of the fifteen cases, and is comparable to it in the other six.

The first row in Table 5 shows a poor performance of DCOMP. A similar behavior is observed for a couple of cases in van Ryzin and Liu [16, Section 7.3]. In their case, this phenomenon occurs in the extreme case

where the probability of no-purchase is zero, and its reason is the degeneracy of the optimal primal solution. Neither of these observations applies in our case, and further investigation would be needed.

α	v_0	UB	DCOMP		DCOMP-0.5		CDLP		ROPT-0.01		INDEP	
			Mean	%LF	Mean	%LF	Mean	%LF	Mean	%LF	Mean	%LF
0.6	(1,5)	215,793	197,038	88.90	196,920	88.06	207,890	91.27	200,444	93.37	172,362	97.71
	(5,10)	200,515	194,146	93.35	191,443	92.09	194,393	91.90	192,896	95.07	163,905	96.76
	(10,20)	170,137	167,866	92.68	167,902	92.79	164,089	91.45	166,919	93.07	151,801	92.48
0.8	(1,5)	266,934	262,823	86.79	263,023	86.37	261,264	85.62	252,013	86.90	204,572	94.60
	(5,10)	223,173	220,891	90.48	221,012	90.51	215,884	89.38	217,073	90.75	191,066	90.22
	(10,20)	188,574	186,219	85.29	185,969	85.27	184,182	84.86	186,325	85.61	172,246	84.09
1.0	(1,5)	281,967	279,506	81.34	279,536	81.36	277,738	80.80	278,344	81.04	226,002	87.71
	(5,10)	235,284	233,929	84.41	233,891	84.27	230,342	83.86	233,138	84.09	209,701	83.64
	(10,20)	192,038	191,646	76.10	191,623	76.05	190,283	76.34	191,727	75.89	188,058	76.73
1.2	(1,5)	284,772	284,736	71.85	284,747	71.85	282,842	71.55	283,280	72.47	243,930	82.48
	(5,10)	238,562	238,539	72.38	238,502	72.26	238,299	72.03	238,548	72.35	225,691	77.65
	(10,20)	192,373	192,530	65.86	192,524	65.87	192,511	65.88	192,532	65.87	192,416	65.80
1.4	(1,5)	287,076	286,743	62.14	286,629	62.16	285,417	61.96	286,160	62.24	259,039	76.96
	(5,10)	238,562	238,843	61.80	238,843	61.80	238,843	61.80	238,843	61.80	231,937	68.82
	(10,20)	192,373	192,541	56.48	192,541	56.48	192,541	56.48	192,541	56.48	192,468	56.42

Table 5: Revenue results for the Small Network example.

Note that when there is a high chance of not buying (i.e., $v_0 = (10, 20)$), ROPT (specially ROPT-0.01) performs very well, even slightly better than DCOMP for $\alpha = 0.8$ and $\alpha = 1$. This could be due to the fact that the marginal value of capacity does not change from one small period to the other one, since residual capacity is more likely to remain the same, given the lack of transactions. Hence, the dual variables are providing an accurate estimate for it.

Like in the previous example, the gains of the choice-based methods with respect to INDEP are quite significant, even though in several cases the load factor of INDEP is higher than the load factor of DCOMP. This indicates that the choice based methods and INDEP induce a different passenger mix. Namely, INDEP sells more tickets, but in a less smart way.

Note that for the ample capacity cases $\alpha = 1.2$ and $\alpha = 1.4$, all choice-based methods perform similarly in terms of expected revenues. This is due to the fact that for these ample cases, the capacity exceeds the demand over the different legs. Then, offering all (or almost all) the products is an appropriate strategy.

Table 7 shows computational time statistics (in seconds) for the different methods under the choice scenarios studied. The column COLGEN shows the time elapsed for solving the CDLP, including the column generation subproblem. The fourth column represents the time for filling the table $V_t(x)$ used in the decomposition heuristic DCOMP (see the description of phase 1 in Section 5), while the next is the average time needed to dynamically solve problem (15) once. In this example, we used the exact linearization procedure discussed in Section 4.2.1. The reported times for ROPT-freq are the total cumulative time to reoptimize across the whole booking horizon. The times for INDEP correspond to the solution of the associated DLP.

The main message from Table 7 is that all the choice-based methods are computationally feasible. Certainly, building the DP table $V_t(x)$ is the most time consuming. But this is a batch process that will be run periodically by the airline. The solution of the dynamic problem (15) is noticeably fast using the exact MIP method (just a small fraction of a second), and hence the operational policy is indeed feasible.

Eventually, if the computational time for filling table $V_t(x)$ becomes an issue in a larger network, the airline could safely use the less time-consuming CDLP or ROPT, and still get high quality solutions.

The noticeable short times observed for COLGEN are due to the fact that the number of iterations

α	v_0	% DCOMP-0.5	% CDLP	% ROPT-0.01	% INDEP
0.6	(1,5)	0.05 ± 0.16	-5.50 ± 0.18	-1.72 ± 0.17	12.52 ± 0.15
	(5,10)	1.39 ± 0.17	-0.12 ± 0.18	0.64 ± 0.16	15.57 ± 0.15
	(10,20)	-0.02 ± 0.18	2.25 ± 0.19	0.56 ± 0.17	9.57 ± 0.16
0.8	(1,5)	-0.07 ± 0.13	0.59 ± 0.15	4.11 ± 0.14	22.16 ± 0.13
	(5,10)	-0.05 ± 0.15	2.26 ± 0.15	1.72 ± 0.15	13.50 ± 0.16
	(10,20)	0.13 ± 0.20	1.09 ± 0.19	-0.05 ± 0.20	7.50 ± 0.20
1.0	(1,5)	-0.01 ± 0.14	0.63 ± 0.15	0.41 ± 0.15	19.14 ± 0.15
	(5,10)	0.01 ± 0.16	1.53 ± 0.16	0.33 ± 0.19	10.35 ± 0.17
	(10,20)	0.01 ± 0.22	0.71 ± 0.22	-0.04 ± 0.30	1.87 ± 0.22
1.2	(1,5)	0.00 ± 0.16	0.66 ± 0.16	0.51 ± 0.17	14.33 ± 0.16
	(5,10)	0.01 ± 0.19	0.10 ± 0.18	0.00 ± 0.19	5.38 ± 0.18
	(10,20)	0.00 ± 0.22	0.00 ± 0.23	0.00 ± 0.22	0.05 ± 0.21
1.4	(1,5)	0.03 ± 0.17	0.46 ± 0.17	0.20 ± 0.17	9.66 ± 0.16
	(5,10)	0.00 ± 0.19	0.00 ± 0.19	0.00 ± 0.19	2.89 ± 0.19
	(10,20)	0.00 ± 0.22	0.00 ± 0.22	0.00 ± 0.22	0.03 ± 0.22

Table 6: 95% confidence intervals for the suboptimality gaps with respect to DCOMP in the Small Network example.

α	v_0	COLGEN	V_t^i	Solution of (15)	ROPT-0.01	INDEP
0.6	(1,5)	0.05	171.81	0.01	6.64	0.30
	(5,10)	0.11	246.14	0.02	11.00	0.31
	(10,20)	0.07	282.52	0.02	6.08	0.31
0.8	(1,5)	0.07	285.43	0.03	7.71	0.42
	(5,10)	0.15	372.04	0.09	10.41	0.42
	(10,20)	0.05	406.98	0.02	4.19	0.43
1.0	(1,5)	0.09	409.9	0.06	7.43	0.54
	(5,10)	0.08	498.79	0.04	4.52	0.53
	(10,20)	0.05	518.30	0.02	2.74	0.54
1.2	(1,5)	0.09	506.72	0.06	7.09	0.64
	(5,10)	0.04	616.86	0.02	2.74	0.64
	(10,20)	0.04	636.43	0.01	2.29	0.67
1.4	(1,5)	0.07	599.47	0.05	6.53	0.75
	(5,10)	0.04	729.83	0.02	2.62	0.78
	(10,20)	0.03	737.17	0.01	2.29	0.79

Table 7: Computational times (in seconds) for Small Network instance

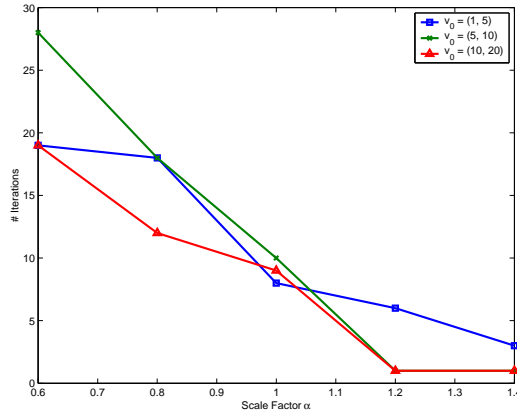


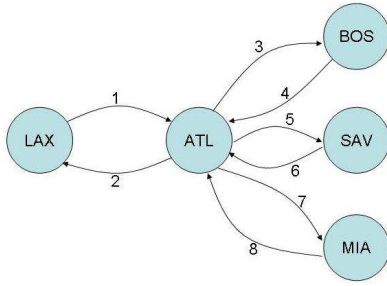
Figure 3: Number of iterations of COLGEN as a function of α and v_0 for the Small Network example.

that it performs is very small relative to the number of variables of the CDLP (see Figure 3). In this example, there are 22 products, giving $2^{22} - 1 = 4,194,303$ variables. Nevertheless, the maximum number of iterations observed across all the cases is 28, a number several various of magnitude smaller than the number of variables. Furthermore, the greedy heuristic finds an entering column in more than 99% of the iterations of COLGEN.

Another interesting observation is that for $\alpha = 1.2$ or 1.4 , and $v_0 = (5, 10)$ or $v_0 = (10, 20)$, COLGEN iterates just once, and the optimal solution has only one variable in the base: $t(S) = N$, i.e., the complete set of products. This is due to the fact that for these ample capacity cases, the only relevant constraint is the time constraint. Figure 3 also shows that the number of iterations for each v_0 decreases as α increases. Again, this is explained by the fact that the complexity of COLGEN reduces as the leg capacities increase.

6.3 Example 3: Hub-and-Spoke Network

This example is based on the five airport network of Figure 4, and itinerary/fare data from Williamson [18]. There are ten roundtrip itineraries. Each itinerary is segmented into 4 different classes: Y, M, B and Q (see Figure 4). Each product is a combination of one-way itinerary and class. For example: product 1 is the one-way itinerary ATLBOS for class Y, which has revenue $r_1 = 310$ and uses leg 3; product 2 corresponds to ATLBOS, class M, with revenue $r_2 = 290$. Product 2 also uses leg 3. Product 5 is BOSATL, class Y, with associated $r_5 = 310$; it uses leg 4. This gives eight legs in the network and eighty products.



Airline network.

O-D Market	Legs	Revenue			
		Y	M	B	Q
ATLBOS/BOSATL	3/4	310	290	95	69
ATLLAX/LAXATL	2/1	455	391	142	122
ATLMIA/MIAATL	7/8	280	209	94	59
ATLSAV/SAVATL	5/6	159	140	64	49
BOSLAX/LAXBOS	4-2/1-3	575	380	159	139
BOSMIA/MIABOS	4-7/8-3	403	314	124	89
BOSSAV/SAVBOS	4-5/6-3	319	250	109	69
LAXMIA/MIALAX	1-7/8-2	477	239	139	119
LAXSAV/SAVLAX	1-5/6-2	502	450	154	134
MIASAV/SAVMIA	8-5/6-7	226	168	84	59

Product definition.

Figure 4: Network definition for the Hub and Spoke Network instance.

There are 40 overlapping customer segments in this example, as described in Table 8. For each possible O-D pair, we define a segment of price sensitive customers that just considers classes B and Q; and another segment of price sensitive customers, but who are eventually willing to buy up to highest class Y. For instance, for the ATL/BOS market, the segments defined are:

- ATL/BOS H, characterized by a consideration set that includes all possible products between ATL and BOS, with preference order for classes $Q > B > M > Y$.
- ATL/BOS L, characterized by a consideration set that includes classes B and Q between ATL and BOS, with preference order for classes $Q > B$.

The booking horizon consists of $T = 2,000$ time periods, with an average of 1,732 arrivals per stream of demand. We generate 500 streams of arrivals per case, and process each stream using all the methods mentioned before. Table 9 summarizes the expected revenues and load factors obtained under DCOMP,

Segment	C_l	v_l	λ_l	Segment	C_l	v_l	λ_l
ATL/BOS H	{1,2,3,4}	{6,7,9,10}	0.015	BOS/MIA H	{41,42,43,44}	{6,7,10,10}	0.008
ATL/BOS L	{3,4}	{8,10}	0.035	BOS/MIA L	{43,44}	{8,10}	0.03
BOS/ATL H	{5,6,7,8}	{6,7,9,10}	0.015	MIA/BOS H	{45,46,47,48}	{6,7,10,10}	0.008
BOS/ATL L	{7,8}	{8,10}	0.035	MIA/BOS L	{47,48}	{8,10}	0.03
ATL/LAX H	{9,10,11,12}	{5,6,9,10}	0.01	BOS/SAV H	{49,50,51,52}	{5,6,9,10}	0.01
ATL/LAX L	{11,12}	{10,10}	0.04	BOS/SAV L	{51,52}	{8,10}	0.035
LAX/ATL H	{13,14,15,16}	{5,6,9,10}	0.01	SAV/BOS H	{53,54,55,56}	{5,6,9,10}	0.01
LAX/ATL L	{15,16}	{10,10}	0.04	SAV/BOS L	{55,56}	{8,10}	0.035
ATL/MIA H	{17,18,19,20}	{5,5,10,10}	0.012	LAX/MIA H	{57,58,59,60}	{5,6,10,10}	0.012
ATL/MIA L	{19,20}	{8,10}	0.035	LAX/MIA L	{59,60}	{9,10}	0.028
MIA/ATL H	{21,22,23,24}	{5,5,10,10}	0.012	MIA/LAX H	{61,62,63,64}	{5,6,10,10}	0.012
MIA/ATL L	{23,24}	{8,10}	0.035	MIA/LAX L	{63,64}	{9,10}	0.028
ATL/SAV H	{25,26,27,28}	{4,5,8,9}	0.01	LAX/SAV H	{65,66,67,68}	{6,7,10,10}	0.016
ATL/SAV L	{27,28}	{7,10}	0.03	LAX/SAV L	{67,68}	{10,10}	0.03
SAV/ATL H	{29,30,31,32}	{4,5,8,9}	0.01	SAV/LAX H	{69,70,71,72}	{6,7,10,10}	0.016
SAV/ATL L	{31,32}	{7,10}	0.03	SAV/LAX L	{71,72}	{10,10}	0.03
BOS/LAX H	{33,34,35,36}	{5,5,7,10}	0.01	MIA/SAV H	{73,74,75,76}	{6,7,8,10}	0.01
BOS/LAX L	{35,36}	{9,10}	0.032	MIA/SAV L	{75,76}	{9,10}	0.025
LAX/BOS H	{37,38,39,40}	{5,5,7,10}	0.01	MIA/SAV H	{77,78,79,80}	{6,7,8,10}	0.01
LAX/BOS L	{39,40}	{9,10}	0.032	MIA/SAV L	{79,80}	{9,10}	0.025

Table 8: Segment definitions for Hub-and-Spoke Network example.

DCOMP-0.5, CDLP, and INDEP.⁴ Table 10 summarizes the suboptimality gaps of the other methods with respect to DCOMP. The notation for the no-purchase preference vector v_0 is the same as the notation used in Example 6.2.

α	v_0	UB	DCOMP		DCOMP-0.5		CDLP		INDEP	
			Mean	%LF	Mean	%LF	Mean	%LF	Mean	%LF
0.6	(1,5)	163,897	160,624	97.10	160,206	95.03	156,557	95.70	110,471	98.64
	(5,10)	132,674	130,971	97.68	130,875	97.29	126,425	95.45	104,330	98.30
	(10,20)	111,897	110,314	97.61	110,209	96.93	106,688	95.53	96,661	97.85
0.8	(1,5)	177,384	175,598	97.70	173,520	93.66	170,301	96.05	130,841	98.72
	(5,10)	146,338	144,597	97.44	144,377	96.99	140,857	95.90	123,399	98.37
	(10,20)	122,464	121,062	96.24	120,985	96.14	117,621	96.03	114,012	97.53
1.0	(1,5)	187,270	185,384	96.43	184,785	95.99	181,673	95.57	149,246	98.63
	(5,10)	156,243	154,718	94.52	154,508	94.21	151,907	95.03	140,161	98.03
	(10,20)	128,386	127,343	91.65	127,255	91.88	125,811	92.27	126,091	92.09
1.2	(1,5)	195,269	193,511	94.88	192,953	94.89	190,000	93.71	165,880	98.29
	(5,10)	160,206	159,386	87.28	159,354	87.37	157,877	87.36	154,210	95.57
	(10,20)	128,448	128,336	78.36	128,336	78.36	128,336	78.36	128,361	78.38
1.4	(1,5)	197,113	196,886	86.70	196,860	86.77	196,639	86.79	179,983	96.54
	(5,10)	160,453	160,350	76.28	160,352	76.28	160,350	76.28	159,435	85.54
	(10,20)	128,448	128,336	68.22	128,336	68.22	128,336	68.22	128,363	68.24

Table 9: Simulation results for Hub-and-Spoke Network example.

Once again, the results obtained for DCOMP and DCOMP-0.5 are very similar. However, there is a clear dominance of these two policies with respect to CDLP, of the order of 2%, which is significant for revenue management standards. It is also clear the advantage of using choice-based models with respect to

⁴We do not report results for ROPT in this example. Given the size of the problem, and the fact that we should reoptimize frequently in every single simulated stream of demand, it would have taken a long computational time to get the results for the confidence intervals.

the standard INDEP paradigm, specially for the scarce capacity cases. The magnitude of the improvement could be as high as 30%, and is generally between 3% and 15%, indicating the importance of accounting for choice behavior in revenue management systems.

α	v_0	% DCOMP-0.5	% CDLP	% INDEP
0.6	(1,5)	0.26 \pm 0.42	2.53 \pm 0.38	31.22 \pm 0.33
	(5,10)	0.07 \pm 0.41	3.47 \pm 0.40	20.34 \pm 0.35
	(10,20)	0.09 \pm 0.41	3.28 \pm 0.40	12.37 \pm 0.37
0.8	(1,5)	1.18 \pm 0.38	3.01 \pm 0.36	25.48 \pm 0.31
	(5,10)	0.15 \pm 0.37	2.58 \pm 0.36	14.66 \pm 0.31
	(10,20)	0.06 \pm 0.37	2.84 \pm 0.36	5.82 \pm 0.33
1.0	(1,5)	0.32 \pm 0.33	2.00 \pm 0.32	19.49 \pm 0.27
	(5,10)	0.13 \pm 0.34	1.81 \pm 0.32	9.40 \pm 0.29
	(10,20)	0.06 \pm 0.36	1.20 \pm 0.35	0.98 \pm 0.35
1.2	(1,5)	0.28 \pm 0.30	1.81 \pm 0.29	14.27 \pm 0.25
	(5,10)	0.02 \pm 0.32	0.94 \pm 0.31	3.24 \pm 0.28
	(10,20)	0.00 \pm 0.39	0.00 \pm 0.39	-0.01 \pm 0.40
1.4	(1,5)	0.00 \pm 0.31	0.11 \pm 0.31	8.57 \pm 0.27
	(5,10)	0.00 \pm 0.35	0.00 \pm 0.35	0.57 \pm 0.32
	(10,20)	0.00 \pm 0.39	0.00 \pm 0.39	-0.02 \pm 0.40

Table 10: 95% confidence intervals for the suboptimality gaps with respect to DCOMP in the Hub-and-Spoke Network example.

Table 11 shows computational time statistics for the methods under evaluation. The most time consuming process is the computation of the table V_t^i , which varies between 2.5 and 10 hours. This is explained by the size of the problem: 2,000 time periods, and eight legs with capacities between 120 and 180 seats (depending on the scale factor α). Nevertheless, they may be acceptable times for an overnight batch process.

Differently from the previous examples, and given that here there are 80 products, we use the greedy heuristic to solve dynamically problem (15) in order to decide the next offer set. The fifth column in Table 11 shows that it takes less than a tenth of a second. The high quality of the solutions obtained indicates that this option does not hurt the performance of the policies.

α	v_0	COLGEN (seconds)	V_t^i (minutes)	Solution of (15) (seconds)	INDEP (seconds)
0.6	(1,5)	73.31	168.28	0.005	3.62
	(5,10)	0.70	205.11	0.006	3.66
	(10,20)	1.53	228.25	0.007	3.73
0.8	(1,5)	851.36	245.9	0.006	4.93
	(5,10)	5.75	295.40	0.007	4.99
	(10,20)	10.60	327.94	0.008	5.10
1.0	(1,5)	3184.22	341.30	0.006	6.31
	(5,10)	3.01	393.24	0.008	6.40
	(10,20)	0.25	422.22	0.008	6.39
1.2	(1,5)	1188.89	442.15	0.007	7.67
	(5,10)	10.55	490.50	0.007	7.67
	(10,20)	0.13	516.08	0.008	7.68
1.4	(1,5)	107.52	548.21	0.007	8.94
	(5,10)	0.85	576.21	0.007	8.96
	(10,20)	0.13	584.09	0.007	8.85

Table 11: Computational times for the Hub-and-Spoke Network.

In this example, the times reported for COLGEN are significantly higher than in previous examples, specially for $v_0 = (1, 5)$. However, in 10 of the 15 cases, a solution is obtained in less than 11 seconds, when there are $2^{80} - 1$ variables $t(S)$. Figure 5 shows the number of iterations of COLGEN for each v_0 , as a function of α . Again, we see that the number of iterations exhibits a decreasing trend in α . In the extreme case $\alpha = 1.4$, all the cases converge to a single iteration, because the solutions consist of a unique offer set $t(S)$. But the difference in times is also influenced by the computational overhead of solving the column generation subproblem. In this example, the greedy heuristic fails 12.36% of the times in finding an entering column, and the exact MIP linearization procedure is then called. In fact, the call to the MIP procedure occurs most of the times when $v_0 = (1, 5)$ (actually, in 11% of the times).

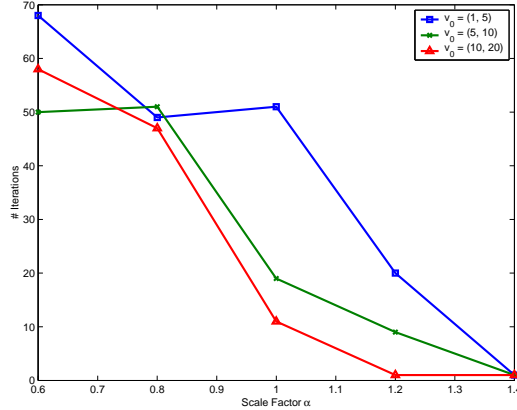


Figure 5: Number of iterations of COLGEN as a function of α and v_0 for the Hub-and-Spoke Network example.

7 Conclusions

In this paper, we consider the choice-based, deterministic, linear programming model (CDLP) of Gallego et al. [6], and then explored by van Ryzin and Liu [16]. We study the more general case where customers belonging to overlapping segments choose according to the multinomial logit model. In order to solve the CDLP for real-size networks, we need to develop a column generation algorithm. We prove that even the column generation subproblem –formulated as a 0-1 fractional programming problem– is NP-Complete, and propose a polynomial time, greedy heuristic to solve it. This heuristic performs remarkably well in practice, both from its computational time and the quality of the solutions derived. We also present a MIP formulation of the 0-1 fractional programming, column generation problem, to be used when solving the master CDLP problem and the greedy heuristic does not find an entering column. According to our experience, though, in most of the cases a routine call to the heuristic from the master CDLP is enough to get a new entering column.

Being able to efficiently solve the column generation subproblem has broader implications. For example, the column generation problem shares the structure of the marketing science problem defined by a seller that receives streams of customers from overlapping segments, and needs to decide what the optimal assortment of products to offer is in order to maximize the instantaneous revenue rate, when the prices of the products are fixed. The greedy heuristic constitutes a useful practical technique for solving it.

Going back to our revenue management problem, in this paper we also extend the decomposition heuristic

(DCOMP) proposed in van Ryzin and Liu [16] to the overlapping segment case. We run an exhaustive series of computational tests, and compare its performance with CDLP, a randomized variation of CDLP, and a reoptimization scheme that uses elements of both DCOMP and CDLP. Our results are somewhat aligned with the findings of van Ryzin and Liu [16]: DCOMP outperforms the other methods, especially in high load factor settings. However, the other methods show a remarkably good revenue performance, consistently outperforming the behavior of the independent demand model assumption, in several cases by more than 10% (especially in high load factor cases), confirming that choice behavior is a first order effect from a revenue management perspective.

Our experiments also show the practical feasibility of the column generation approach by providing the computational times of the different methods. The decomposition heuristic is clearly the most time consuming method. However, its core is about building off-line a big two-dimensional table to be used afterwards when computing real time the set of products to offer. This is indeed a batch process that is typically run overnight by the airlines, and hence it is still practically feasible. Nevertheless, if time becomes an issue for larger networks, the computational times of the other methods (i.e. CDLP, RCDLP, and ROPT) are at least an order of magnitude shorter, hence constituting an interesting alternative considering the high quality of the solutions obtained. Among these other methods, considering the tradeoff between computational time and quality of the solutions obtained, both CDLP and the randomized version of it seem to be the most promising approaches to implement in practice.

References

- [1] BELOBABA, P., AND HOPPERSTAD, C. Boeing/mit simulation study: Pods results update. In *1999 AGIFORS Reservations and Yield Management Study Group Symposium Proceedings* (London, UK, 1999).
- [2] BEN-AKIVA, M., AND LERMAN, S. *Discrete Choice Analysis: Theory and Applications to Travel Demand*, sixth ed. The MIT Press, Cambridge, MA, 1994.
- [3] CARDELL, N., AND DUNBAR, F. Measuring the societal impacts of automobile downsizing. *Transportation Research A 14* (1980), 423–434.
- [4] COOPER, W. Asymptotic behavior of an allocation policy for revenue management. *Operations Research 50* (2002), 720–727.
- [5] COOPER, W., DE MELLO, T. H., AND KLEYWEGT, A. Models of the spiral-down effect in revenue management. *Operations Research 54* (2006), 968–987.
- [6] GALLEGO, G., IYENGAR, G., PHILLIPS, R., AND DUBEY, A. Managing flexible products on a network. Tech. Rep. CORC TR-2004-01, Department of Industrial Engineering and Operations Research, Columbia University, 2004.
- [7] GAREY, M., AND JOHNSON, D. *Computers and intractability: A guide to the theory of NP-Completeness*. W.H. Freeman and Company, San Francisco, CA, 1979.
- [8] HAMMER, P., AND RUDEANU, S. *Boolean methods in operations research and related areas*. Springer, Berlin, Germany, 1968.
- [9] HANSEN, P., DE ARAGÃO, M. P., AND RIBEIRO, C. C. Hyperbolic 0–1 programming and query optimization in information retrieval. *Mathematical Programming 52* (1991), 256–263.

- [10] MCFADDEN, D., AND TRAIN, K. Mixed MNL models for discrete response. *Journal of Applied Econometrics* 15 (2000), 447–470.
- [11] PROKOPYEV, O., HUANG, H., AND PARDALOS, P. On complexity of unconstrained hyperbolic 0–1 programming problems. *Operations Research Letters* 33 (2005), 312–318.
- [12] SECOMANDI, N. An analysis of the control-algorithm re-solving issue in inventory and revenue management. Technical report, Tepper School of Business, Carnegie Mellon University, 2005.
- [13] TALLURI, K., AND VAN RYZIN, G. *The Theory and Practice of Revenue Management*. Kluwer Academic Publishers, New York, NY, 2004.
- [14] TALLURI, K. T., AND VAN RYZIN, G. J. Revenue management under a general discrete choice model of consumer behavior. *Management Science* 50 (2004), 15–33.
- [15] VAN RYZIN, G. Models of demand. *Journal of Revenue and Pricing Management* 4 (2005), 204–210.
- [16] VAN RYZIN, G., AND LIU, Q. On the choice-based linear programming model for network revenue management. Working paper, Graduate School of Business, Columbia University, New York, 2004.
- [17] VAN RYZIN, G., AND VULCANO, G. Computing virtual nesting controls for network revenue management under customer choice behavior. Working paper, Graduate School of Business, Columbia University, New York, NY, 2004.
- [18] WILLIAMSON, E. *Airline Network Seat Inventory Control: Methodologies and Revenue Impacts*. PhD thesis, Flight Transportation Laboratory, MIT, Cambridge, MA, 1992.
- [19] WU, T. A note on a global approach for general 0–1 fractional programming. *European Journal of Operational Research* 101 (1997), 220–223.
- [20] ZHANG, D., AND ADELMAN, D. An approximate dynamic programming approach to network revenue management with customer choice. Working paper, Graduate School of Business, University of Chicago, Chicago, IL, 2006.
- [21] ZHANG, D., AND COOPER, W. L. Revenue management for parallel flights with customer-choice behavior. *Operations Research* 53 (2005), 415–431.