Author(s): Meissner, J., Strauss, A.K., and Talluri, K.

Article Title: An Enhanced Concave Program Relaxation for Choice Network Revenue Management

Year of publication: Forthcoming

Link to published article:
http://www.wiley.com/bw/journal.asp?ref=1059-1478

Publisher statement: The definitive version is available at www3.interscience.wiley.com

An Enhanced Concave Program Relaxation for Choice Network Revenue Management

Joern Meissner[*], Arne Strauss[†], Kalyan Talluri[‡]

29 November 2011

Abstract

The network choice revenue management problem models customers as choosing from an offer set, and the firm decides the best subset to offer at any given moment to maximize expected revenue. The resulting dynamic program for the firm is intractable and approximated by a deterministic linear program called the $CDLP$ which has an exponential number of columns. However, under the choice-set paradigm when the segment consideration sets overlap, the $CDLP$ is difficult to solve. Column generation has been proposed but finding an entering column has been shown to be NP-hard. In this paper, starting with a concave program formulation called $SDCP$ that is based on segment-level consideration sets, we add a class of constraints called product constraints (σPC), that project onto subsets of intersections. In addition we propose a natural direct tightening of the $SDCP$ called $ESDCP_\kappa$, and compare the performance of both methods on the benchmark data sets in the literature. In our computational testing on the benchmark data sets in the literature, $2PC$ achieves the $CDLP$ value at a fraction of the CPU time taken by column generation. For a large network our $2PC$ procedure runs under 70 seconds to come within 0.02% of the $CDLP$ value, while column generation takes around 1 hour; for an even larger network with 68 legs, column generation does not converge even in 10 hours for most of the scenarios while $2PC$ runs under 9 minutes. Thus we believe our approach is very promising for quickly approximating $CDLP$ when segment consideration sets overlap and the consideration sets themselves are relatively small.

Keywords: discrete-choice models, network revenue management, optimization

[*]Kuehne Logistics University, Hamburg, Germany.
[†]Warwick Business School, University of Warwick, Coventry CV4 7AL, United Kingdom.
[‡]ICREA & Universitat Pompeu Fabra, Ramon Trias Fargas 25-27, 08005 Barcelona, Spain.

1 Introduction and literature review

Revenue management (RM) is the control of the sale of a limited quantity of a resource (hotel rooms for a night, airline seats, advertising slots etc.) to a heterogenous population with different valuations for a unit of the resource. The resource is perishable, and for simplicity sake, we assume that it perishes at a fixed point of time in the future. Customers are independent of each other and arrive randomly during a sale period, and demand one unit of resource each. Sale is online, and the firm has to decide which products at what price it should offer, the tradeoff being selling too much at too low a price early and running out of capacity, or, losing too many price sensitive customers and ending up with excess unsold inventory.

In industries such as hotels, airlines and media, the products consume bundles of different resources (multi-night stays, multi-leg itineraries) and the decision on whether to offer a particular product at a certain price depends on the expected future demand and current inventory levels for all the resources used by the product (and also indirectly, all the resources in the network). Network revenue management (network RM) is control based on the demands for the entire network. Chapter 3 of Talluri and van Ryzin (2004b) contains all the necessary background on network RM.

RM incorporating more realistic models of customer behavior, as customers choosing from an offer set, have recently become popular (Talluri and van Ryzin (2004a), Gallego, Iyengar, Phillips, and Dubey (2004), Liu and van Ryzin (2008), Kunnumkal and Topaloglu (2010), Zhang and Adelman (2009), Meissner and Strauss (2012), Bodea, Ferguson, and Garrow (2009), Bront, Méndez-Díaz, and Vulcano (2009), Méndez-Díaz, Bront, Vulcano, and Zabala (2012), Kunnumkal (2011)).

The network versions of choice RM are usually modifications of older methods proposed for network RM with the so-called independent-class assumption—for instance the choice-based deterministic linear program ($CDLP$) that we study in this paper can be considered the equivalent of the deterministic linear programming (DLP) formulation, a practical and widely used approximation for the dynamic program under the independent-class assumption. However, accounting for customer choice behavior makes the approximations considerably more difficult to solve. So while the DLP approximation is easy, the corresponding $CDLP$ approximation is NP-hard even for relatively simple models of customer choice. The $CDLP$ formulation has an exponential number of columns and the solution strategy is to use column generation; but finding an entering column is computationally easy only in restrictive cases (multinomial logit (MNL) model of choice with non-overlapping segment consideration sets).

Various mathematical programming approaches have been proposed (e.g. Kunnumkal and Topaloglu (2010), Zhang and Adelman (2009), Meissner and Strauss (2012)) that are tighter relaxations of the dynamic program characterizing the underlying decision problem than the $CDLP$, however, their solution is more difficult. In this paper we concentrate on the simpler deterministic linear program $CDLP$.

Given the hardness results for $CDLP$ for overlapping consideration sets, we have to scale back our ambitions of solving even this approximation to dynamic program for large problems. Another alternative is to consider somewhat restrictive situations which still have wide applicability in practice. Along this latter line of research, Talluri (2010) proposed the so-called segment-based deterministic concave program ($SDCP$) that is weaker than the upper bound resulting from the $CDLP$, but coincides for non-overlapping segments (Gallego, Ratliff, and Shebalov (2010) pursue a similar approach to the $CDLP$). The advantage is that the method is tractable for any choice model whenever the number of elements in a segment's consideration set is not too large.

Small consideration sets can be justified in the airline setting where a segment's consideration set consists of choices (on one airline) for travel for an origin and destination, and typically there are only a few such alternatives on a given date (Talluri (2001)). Note that currently airlines solve the network problem for a single date due to its computational complexity—so even if a customer considers multiple days of travel, as far as the optimization model goes, a customer's choice is for the day's offerings.

Our model and methodology applies also to what is called the assortment optimization problem in retail (Kök, Fisher, and Vaidyanathan (2009), Rusmevichientong, Shmoys, and Topaloglu (2010)) since network choice RM can be considered a dynamic assortment optimization problem with an additional network structure for the resources. For this reason we mention the research on consideration sets in the marketing area. There is a large body of literature that empirically and experimentally verifies the formation of consideration sets and the choice of an item in the consideration set. See for instance, Lussier and Olshavsky (1979), Payne (1976), Wright and Barbour (1977). Hauser and Wernerfelt (1990) report average consideration set sizes of less than 4 for common items such as deodorants, shampoos, air fresheners, laundry detergents and coffees.

$SDCP$ is tractable, but its performance is poor when segment consideration sets overlap (i.e., the bound is significantly looser than $CDLP$). In this paper we extend the $SDCP$ formulation to obtain progressively tighter relaxations of $CDLP$ for the case of overlapping consideration sets. We add a novel class of constraints called *product constraints* that interpret the linear programming decision variables as randomization rules. These constraints are easy to generate and work for general discrete choice models—in fact this is the only approach that we know of that can handle general discrete choice models and overlapping segment consideration sets. We report extensive computational results showing their performance on various types of networks. We contrast the results with an extension of $SDCP$ called $ESDCP_\kappa$. In our numerical testing, $SDCP$ with product constraints achieves the $CDLP$ value at a fraction of the CPU time taken by column generation (Tables 16, 17, 18).

The remainder of the paper is organized as follows: In §2 we introduce the notation, the demand model and the basic dynamic program. In §3 we state the $CDLP$ and $SDCP$ approximations of the dynamic program, followed by the presentation of the main computational approaches that we propose in this paper in §4. §5 contains our numerical results using the new methods, and we present our conclusions in §6.

2 Model and notation

A product is a specification of a price and a combination of resources to be consumed. For example, a product could be an itinerary-fare class combination for an airline network, where an itinerary is a combination of flight legs; in a hotel network, a product would be a multi-night stay for a particular room type at a certain price point. Time is discrete and assumed to consist of T intervals, indexed by t. We assume that the booking horizon begins at time 0 and that all the resources perish instantaneously at time T. We make the standard assumption that the time intervals are fine enough so that the probability of more than one customer arriving in any single time period is negligible. The underlying network has m resources (indexed by i) and n products (indexed by j), and we refer to the set of all resources as I and the set of all products as J. A product j uses a subset of resources, and is identified (possibly) with a set of sale restrictions or features and a revenue of r_j. A resource i is said to be in product j ($i \in j$) if j uses resource i. The resources used by j are represented by $a_{ij} = 1$ if $i \in j$, and $a_{ij} = 0$ if $i \notin j$, or alternately with the 0-1 incidence vector A_j of product j. Let A denote the resource-product incidence matrix; columns of A are then A_j. We denote capacity on resource i at time t as $c_{i,t}$ and the vector of capacities as $\vec{c}_t$, so the initial set of capacities at time 0 is $\vec{c}_0$. The vector $\vec{1}$ is a vector of all ones, and $\vec{0}$ is a vector of all zeroes (dimension appropriate to the context).

Whenever it is clear from the context, we represent a mathematical program or a dynamic program by a label that also serves as the optimal value of the program. For example, ($CDLP$) represents the choice-based deterministic linear program (described below) but can also represent the model or the objective function value of the linear program depending on the context.

2.1 Demand model

We assume there are $\mathcal{L} := \{1, \ldots, L\}$ customer segments, each with distinct purchase behavior. In each period, there is a customer arrival with probability λ. A customer belongs to segment l with probability p_l. We denote $\lambda_l = p_l \lambda$ and assume $\sum_l p_l = 1$, so $\lambda = \sum_l \lambda_l$. We are assuming time-homogenous arrivals (homogenous in rates and segment mix), but the model and all solution methods in this paper can be transparently extended to the case when rates and mix change by period. Each segment l has a *consideration set* $C_l \subseteq J$ of products that it considers for purchase. We assume this consideration set is known to the firm (by a previous process of estimation and analysis), and the consideration sets for different segments can overlap.

In each period the firm offers a subset S of its products for sale, called the *offer set*. Given an offer set S, an arriving customer purchases a product j in the set S or decides not to purchase. The no-purchase option is indexed by 0 and is always present for the customer.

A segment-l customer is indifferent to a product outside his consideration set; i.e., his choice probabilities are not affected by products offered not in the consideration set. A segment-l customer purchases $j \in S$ with given probability $P_j^l(S)$. This is a set-function defined on all subsets of J. For the moment we assume these set functions are given by an oracle; it could conceivably be given by a simple formula such as the Multinomial Logit (MNL) model. Whenever we specify probabilities for a segment l for a given offer set S, we just write it with respect to $S_l := C_l \cap S$ (note that $P_j^l(S) = P_j^l(S_l)$). We define the vector $\vec{P}^l(S) = [P_1^l(S_l), \ldots, P_n^l(S_l)]$ (recall the no-purchase option is indexed by 0, so it is not included in this vector).

Given a customer arrival, and an offer set S, the probability that the firm sells $j \in S$ is then given by $P_j(S) = \sum_l p_l P_j^l(S_l)$ and makes no sale with probability $P_0(S) = 1 - \sum_{j \in S} P_j(S)$. We define the vector $\vec{P}(S) = [P_1(S), \ldots, P_n(S)]$. Notice that $\vec{P}(S) = \sum_l p_l \vec{P}^l(S)$. We define the vectors $\vec{Q}^l(S) = A\vec{P}^l(S)$ and $\vec{Q}(S) = A\vec{P}(S)$. The revenue functions can be written as $R^l(S) = \sum_{j \in S_l} r_j P_j^l(S_l)$ and $R(S) = \sum_{j \in S} r_j P_j(S)$.

In our notation and demand model we broadly follow Bront et al. (2009) and Liu and van Ryzin (2008). The motivation for the design of our solution procedures comes from the following premise: The number of elements in a segment's consideration set is usually small. It sounds unlikely that a customer can process hundreds of choices in making a decision. So the problem for a single segment might be tractable by just brute-force enumeration, i.e., the number of subsets of C_l for a segment l can be enumerated explicitly as if say, $|C_l| \sim 10$, we can easily compute all the $2^{10} = 1024$ subsets of C_l. The segment is indifferent to products outside its consideration set, hence the airline would only consider offering some subset of C_l when optimizing revenue from this segment l.

The empirical work of Hauser and Wernerfelt (1990) in marketing assortment optimization and route-set model of Talluri (2001) in the airline context motivate this approach. The difficulty of $CDLP$ is that it is based on subsets of the set of all products J; in contrast, basing the formulation on segment consideration sets allows us to exploit the relatively small size of each segment's consideration set. We remark that all the above-mentioned articles in the literature concentrate only on the MNL model of choice so that understanding of optimization with other choice models is rather limited at this stage. Our assumption of small consideration sets at least allows us a tractable approach for more general discrete choice models.

2.2 Dynamic program

The dynamic program (DP) to determine optimal controls can be written down as follows. Let $V_t(\vec{c}_t)$ denote the maximum expected revenue to go, given remaining capacity $\vec{c}_t$ in period t. Then $V_t(\vec{c}_t)$ must satisfy the well-known Bellman equation

$$V_t(\vec{c}_t) = \max_{S \subseteq J} \left\{ \sum_{j \in S} \lambda P_j(S)(r_j + V_{t+1}(\vec{c}_t - A_j)) + (\lambda P_0(S) + 1 - \lambda) V_{t+1}(\vec{c}_t) \right\} \tag{1}$$

with the boundary condition $V_T(\vec{c}_T) = V_t(\vec{0}) = 0$ for all $\vec{c}_T$ and for all t. Recall that $P_j(S)$ is the total purchase probability (across all the segments, in one time period) of product j and $P_0(S)$ is the total no-purchase probability when the firm offers set S. Let $V^{DP} = V_0(\vec{c}_0)$ denote the optimal value of this dynamic program from 0 to T, for the given initial capacity vector $\vec{c}_0$.

3 Approximations and upper bounds

The dynamic program (1) is computationally intractable, hence we are interested in approximating the value function. In the following, we outline two recently proposed approaches to that end.

3.1 Choice deterministic linear program ($CDLP$)

The choice-based deterministic linear program ($CDLP$) defined in Gallego et al. (2004) and Liu and van Ryzin (2008) is as follows:

$$
(CDLP) \qquad \begin{aligned}
\max \quad & \sum_{S \subseteq J} \lambda R(S) w_S \\
\text{s.t.} \quad & \sum_{S \subseteq J} \lambda w_S \vec{Q}(S) \leq \vec{c}_0 \\
& \sum_{S \subseteq J} w_S = T \\
& 0 \leq w_S, \quad \forall S \subseteq J.
\end{aligned} \tag{2}
$$

The formulation has 2^n variables w_S that can be interpreted as the number of time periods each set is offered (including $w_\emptyset$). Liu and van Ryzin (2008) show that the optimal objective value is an upper bound on V^{DP}. They also show that the problem can be solved efficiently by column-generation for the MNL model and non-overlapping segments. Bront et al. (2009) and Rusmevichientong et al. (2010) investigate this further and show that column generation is NP-hard whenever the consideration sets for the segments overlap, even for the MNL choice model.

3.2 Segment-based deterministic concave program ($SDCP$)

Talluri (2010) proposed the following formulation that coincides with the $CDLP$ when the segments do not overlap. For segment l, define a capacity vector $\vec{0} \leq \vec{y}_{lt} \leq \vec{1}$ that we reserve for sale to segment l in period t (even if we cannot identify this segment at the time of purchase). Given $\vec{y}_{lt}$, let $R_l^*(\vec{y}_{lt})$ represent the optimal revenue we can obtain offering some convex combination of product sets to segment l. $R_l^*(\vec{y}_{lt})$ can be obtained by solving the following linear program:

$$
(Rgen) \qquad \begin{aligned}
R_l^*(\vec{y}_{lt}) = \max \quad & \sum_{S_l \subseteq C_l} \lambda_l R^l(S_l) w^l_{S_l} \\
\text{s.t.} \quad & \sum_{S_l \subseteq C_l} \lambda_l w^l_{S_l} \vec{Q}^l(S_l) \leq \vec{y}_{lt} \\
& \sum_{S_l \subseteq C_l} w^l_{S_l} \leq 1 \\
& w^l_{S_l} \geq 0, \quad \forall S_l \subseteq C_l.
\end{aligned} \tag{3}
$$

Note that $R_l^*(\vec{y}_{lt})$ is a concave function of $\vec{y}_{lt}$. The linear program ($Rgen$) has an exponential number of columns but can be solved by column generation, and the column generation is often easier than that of

$CDLP$ as it is segment specific and $(Rgen)$ considers only subsets of the consideration set of a single segment at a time—for instance, in the case of latent-segment multinomial-logit demand model of choice, the column generation of $(Rgen)$ is tractable. If the number of considered products $|C_l|$ for each segment is small, say 10 or 12, we can just enumerate the columns.

We now define the following concave programming problem over the capacity vectors:

$$
\begin{aligned}
& \max && \sum_{t=1}^{T}\sum_{l=1}^{L} R_l^*(\vec{y}_{lt}) && (4)\\
& \text{s.t.} && \sum_{t=1}^{T}\sum_{l=1}^{L} \vec{y}_{lt} \le \vec{c}_0 && \\
(SDCP)\quad & && \vec{y}_{lt} \le \lambda_l \vec{1}, \quad \forall\, l,t && \\
& && \vec{y}_{lt} \ge \vec{0}. &&
\end{aligned}
$$

The above formulation of $SDCP$ assumes uniform arrival rates and segment mix for simplicity, but can be modified transparently by using time-dependent arrival rates λ_t. This discrete-time formulation can be made compact by merging periods with the same arrival rates.

$(SDCP)$ is a compact formulation compared to $(CDLP)$, and can be solved by any number of standard concave-programming methods generating the objective function values by solving $(Rgen)$. So the critical computation lies in the calculation of $R_l^*(\vec{y}_{lt})$.

The relation between $CDLP$ and $SDCP$ is shown in (Talluri, 2010) and we repeat the connection here to show the validity of the product constraints (i.e., $SDCP$ with product constraints still leads to an upper bound for the dynamic program). First, we formulate $CDLP$ as follows:

$$
\begin{aligned}
& \max && \textstyle\sum_l \lambda_l \sum_{S_l\subseteq C_l} R^l(S_l) w^l_{S_l} && (5)\\
(CDLP_{\mathcal{W}})\quad & && \textstyle\sum_l \lambda_l \sum_{S_l\subseteq C_l} \vec{Q}^l(S_l) w^l_{S_l} \le \vec{c}_0 && (6)\\
& && (w^l_{S_l}) \in \mathrm{Proj}(\mathcal{W}), && (7)
\end{aligned}
$$

where $(w^l_{S_l})$ denotes the vector with (l, S_l)th component being $w^l_{S_l}$ (likewise for (w_S)), $\mathcal{W}$ is the polytope $\{\sum_{S\subseteq J} w_S = 1, w_S \ge 0\,\forall\, S\}$ representing probability distributions (w_S) over all subsets S, and $\mathrm{Proj}(\mathcal{W})$ is the projection of $\mathcal{W}$ onto the space of $(w^l_{S_l})$ via $w^l_{S_l} := \sum_{S:S\cap C_l=S_l} w_S$ for all $S_l \subseteq C_l$ for all l. This projection can be re-written in a more convenient form: We define a subset incidence matrix B with rows for all $S_l \subseteq C_l, l = 1, 2, \ldots, L$ and columns $S \subseteq J$, and $B_{S_l S} := 1$ if subset $S_l = S \cap C_l$ and 0 otherwise.

With that notation, $(w^l_{S_l}) \in \text{Proj}(\mathcal{W})$ if there exists a feasible solution to the following system:

$$\begin{aligned} & \sum_{S\subseteq J} B_{S_l S} w_S &=& \; w^l_{S_l}, \quad \forall\, S_l \subseteq C_l, \forall\, l & (8)\\ (\mathcal{X}) \quad & \sum_{S\subseteq J} w_S &=& \; 1 & (9)\\ & w_S &\geq& \; 0, \quad \forall\, S \subseteq J. & \end{aligned}$$

The $w^l_{S_l}$'s in the above formulation can be thought of as the marginal distribution on subsets of C_l for a distribution of w_S on all subsets $S \subseteq J$.

Proposition 1 (Talluri (2010))**.** $CDLP_{\mathcal{W}} = CDLP$.

Proof

For a feasible $(w^l_{S_l})$ of $(CDLP_{\mathcal{W}})$, $(w^l_{S_l}) \in \text{Proj}(\mathcal{W})$ implies, there exists a (w_S) satisfying (8). Now notice that

$$\sum_{l=1}^{L} \lambda_l \sum_{S_l\subseteq C_l} \vec{Q}_l(S_l) \sum_{S\subseteq J} B_{S_l S} w_S = \sum_{S\subseteq J} \lambda w_S \vec{Q}(S), \tag{10}$$

and therefore (w_S) satisfies $(CDLP)$ with the same objective value (the objective value is the same by a calculation identical to that of (10)).

Likewise, equation (10) also shows that if (w_S) is a feasible solution to $(CDLP)$ we derive a feasible solution $(w^l_{S_l})$ for $(CDLP_{\mathcal{W}})$ by $w^l_{S_l} = B_{S_l S} w_S$, and this has the same objective value.

□

Talluri (2010) shows that $(SDCP)$ overestimates revenue compared to $(CDLP)$, i.e., $CDLP \leq SDCP$, and the objective values of both formulations coincide for the case of non-overlapping segments.

Theorem 1 (Talluri (2010))**.** $V^{SDCP} \geq V^{CDLP}$.

Proof

The matrix B has the property that every column, corresponding to a set S, has at most one element equal to 1 amongst the rows corresponding to the subsets of a segment l. This implies that a feasible solution to $(CDLP_{\mathcal{W}})$ satisfies $\sum_{S_l} w^l_{S_l} \leq 1$ as $\sum w_S = 1$ (recall that we are normalizing $T = 1$). Hence we add these redundant constraints and relax constraints (8) to obtain $SDCP$.

□

4 Tightening $SDCP$

In the most general setting, the segments' consideration sets can overlap in a variety of ways, and the choice probabilities depend on the offer set, and need not follow any structure. Indeed, Rusmevichientong et al. (2010) show that generating the columns of ($CDLP$) even in a very restrictive setting (MNL model of probabilities, two segments) is NP-hard.

In this section we describe the two computational approaches that we propose in this paper.

4.1 Product constraints

The first method is based on consistency of projections onto the intersections of the considerations sets, that we call *product constraints* (the name comes from the interpretation as a restriction arising from the marginal product probabilities). The constraints are called *valid* if adding them still results in an upper bound for the dynamic program (we show that in fact it results in an upper bound on the $CDLP$). We work with a general discrete-choice model of customer behavior as in (Talluri and van Ryzin, 2004a), and we make no assumptions on the (overlapping) structure of the consideration sets. Throughout we assume that choice probabilities are given by an oracle for every segment l and offer set S.

We first describe the intuition behind our constraints: For any product $j \in C_l \cap C_k$, the length of time that product j is offered to segment l must be equal to the length of time that it is being offered to segment k. In order to derive a corresponding constraint, we first normalize $T = 1$ in (CDLP) without loss of generality. So (w_S) can be interpreted as a distribution over subsets of J, and can be considered a randomization rule—at each point choose a subset based on this distribution. The distribution in turn induces a distribution for each one of the segments l, via the matrix B (recall $B_{S_l S} := 1$ if subset $S_l = S \cap C_l$ and 0 otherwise), $w^l_{S_l} := \sum_S B_{S_l S} w_S$ (alternately, $(w^l_{S_l})$ is the marginal distribution of (w_S)).

Let X_j be a Bernoulli random variable which takes the value $X_j = 1$ if $j \in S$ for an offer set S sampled from the w_S distribution, and $X_j = 0$ otherwise. The expectation $E[X_j]$ is then the probability that product j is offered under this randomized rule. Consider a similar sampling from another distribution given by $w^l_{S_l}$'s. This would also lead to a Bernoulli random variable, and if the $w^l_{S_l}$ are induced by the w_S's, the expectations of these random variables should coincide across the segments; i.e., the $E[X_j]$ should be the same for two segments l and k whose consideration sets contain the product j, leading to the constraint:

$$\sum_{\{S_l \subseteq C_l | S_l \ni \{j\}\}} w^l_{S_l} = \sum_{\{S_k \subseteq C_k | S_k \ni \{j\}\}} w^k_{S_k} = E[X_j].$$

Now the space of w_S's is prohibitively large, and the matrix B has almost no structure as the considerations sets are arbitrary. So we choose to work in the smaller space of $w^l_{S_l}$'s as in $SDCP$ (actually $(Rgen)$ of the $SDCP$ formulation) which however are not induced by the w_S's. So we impose consistency conditions that arise *if* the segment-level distributions were generated by a common set of w_S's. We would like these consistency conditions to be linear and to be easily generated.

One can extend this to subsets of products. As the X_j's are Bernoulli random variables, for any pair of segments l and k that contain two products j_1 and j_2 the following equation should hold if we were to offer the same offer set to j_1 and j_2 (the $CDLP$ condition that $SDCP$ relaxes):

$$\sum_{S_l \ni \{j_1, j_2\}} w^l_{S_l} = \sum_{S_k \ni \{j_1, j_2\}} w^k_{S_k} \quad \Big(= E[X_{j_1} X_{j_2}] = \sum_{S \ni \{j_1, j_2\}} w_S \Big).$$

So we can add linear constraints to $(SDCP)$ of the form $\sum_{S_l \ni \{j_1, j_2\}} w^l_{S_l} = \sum_{S_k \ni \{j_1, j_2\}} w^k_{S_k}$ for all segments l, k such that $C_l, C_k \ni \{j_1, j_2\}$. This extends to triples of products $\{j_1, j_2, j_3\}$ via $\sum_{S_l \ni \{j_1, j_2, j_3\}} w^l_{S_l} = \sum_{S_k \ni \{j_1, j_2, j_3\}} w^k_{S_k}$, and so on.

An alternate way of viewing this idea is that the distributions w_S's and $w^l_{S_l}$'s have to be consistent once we project them onto the subsets of the intersection of the consideration sets. Since our premise is that consideration sets are relatively small, intersections of consideration sets are small also (definitely less than the smaller of the two consideration sets), and if the consideration sets are not too large, we can enumerate all subsets of the intersections without much computational effort (§4.1.2 discusses this further).

The difficulty of solving $(CDLP)$ for overlapping segment considerations sets lies in solving $(\mathcal{X})$ as its columns are indexed by all subsets S and the matrix B has almost no structure when the segment consideration sets overlap.

Let us consider the following generalization of $SDCP$ (note that we moved the objective function to the right-hand side by introducing variables z_{lt}):

$$\begin{aligned}
& & \max_{\vec{y}_{lt}} \quad & \sum_{t=1}^{T} \sum_{l=1}^{L} z_{lt} & \\
& & \text{s.t.} \quad & \sum_{t=1}^{T} \sum_{l=1}^{L} \vec{y}_{lt} \leq \vec{c}_0 & \\
& (SDCP_\kappa) & & \vec{y}_{lt} \leq \lambda_l \vec{1} \quad \forall\, l, t & \\
& & & \sum_{l \in \mathcal{L}} z_{lt} \leq R^*_{\mathcal{L}}(\vec{y}_{lt}) \quad \forall\, t, \forall\, \mathcal{L} \subset \{1, \ldots, L\}, |\mathcal{L}| = \kappa & \qquad (11) \\
& & & \vec{y}_{lt} \geq \vec{0}, \quad \forall\, l, t, &
\end{aligned}$$

and

$$
\begin{aligned}
R^*_{\mathcal{L}}(\vec{y}_{lt}) = \quad & \max \quad \sum_{l\in\mathcal{L}} \sum_{S_l\subseteq C_l} \lambda_l R^l(S_l) w^l_{S_l} \\
& \text{s.t.} \quad \sum_{S_l\subseteq C_l} \lambda_l \vec{Q}^l(S_l) w^l_{S_l} \le \vec{y}_{lt} \quad \forall l \in \mathcal{L}, \\
(Rgen_{\mathcal{L}}) \qquad & \qquad \sum_{S_l\subseteq C_l} w^l_{S_l} \le 1 \quad \forall l \in \mathcal{L}, \\
& \qquad w^l_{S_l} \ge 0,\ \forall\ l \in \mathcal{L}, \forall\ S_l \subseteq C_l.
\end{aligned}
$$

If we consider $\kappa = 1$, i.e., only subsets of the form $\mathcal{L} := \{l\}$, we recover $(SDCP)$. However, if we define $\mathcal{L} := \{l, k\}$ to contain two segments, i.e. $\kappa = 2$, and say the segment consideration sets overlap, then we can tighten the formulation by adding the following constraints to $(Rgen_{\mathcal{L}})$:

$$
\sum_{S_l \supseteq S_{lk}} w^l_{S_l} = \sum_{S_k \supseteq S_{lk}} w^k_{S_k}, \quad \forall S_{lk} \subseteq C_l \cap C_k. \tag{12}
$$

We call these the *product constraints (PC)* and if we restrict $|S_{lk}| = \sigma$, we refer to them as σPC constraints. We refer to $(SDCP_2)$ with σPC constraints added to $R^*_{\mathcal{L}}$ as the σPC formulation.

One can combine $(SDCP_\kappa)$ with $(Rgen_{\mathcal{L}})$ and the σPC into a single linear program, or if the problem is too big to fit into memory, we can implement this by obtaining the dual solution $(\vec{\pi}, \vec{\mu})$ to $(Rgen_{\mathcal{L}})$ with the additional constraints (12) for the current $\vec{y}_{lt}$, and adding the cut $\sum_{l\in\mathcal{L}} z_{lt} \le \sum_{l\in\mathcal{L}} \vec{\pi}_l^\top \vec{y}_{lt} + \mu_l$ to $(SDCP_2)$ iteratively.

Proposition 2. *Suppose we add σPC constraints to $(Rgen_{\mathcal{L}})$ and solve $SDCP_2$ as described above (namely, the σPC formulation), then the value of the resulting linear program is greater than or equal to $CDLP$.*

Proof

Suppose $w^l_{S_l}$ and $w^k_{S_k}$'s are feasible solutions of $(CDLP_{\mathcal{W}})$, then there exists a set of w_S's such that $w^l_{S_l} = \sum_S B_{S_l S} w_S$ as $w^l_{S_l} \in \text{Proj}(\mathcal{W})$. Then, for any fixed $S_{lk} \subseteq C_l \cap C_k$, we have:

$$
\sum_{S_l \supseteq S_{lk}} w^l_{S_l} = \sum_{S_l \supseteq S_{lk}} \sum_S B_{S_l S} w_S = \sum_{S | S\cap C_l \supseteq S_{lk}} w_S = \sum_{S | S\cap C_k \supseteq S_{lk}} w_S = \sum_{S_k \supseteq S_{lk}} w^k_{S_k}.
$$

So the $w^l_{S_l}$'s should satisfy the product constraints (12) and a solution of $(CDLP_{\mathcal{W}})$ leads to a feasible solution of $(SDCP_2)$ with the product constraints (12) added. Thus the value of the maximization problem $(SDCP_2)$ is higher than than of $(CDLP_{\mathcal{W}})$.

□

Thus we obtain an upper bound on $CDLP$ $(= CDLP_{\mathcal{W}})$ by adding product constraints that promises to be a considerable tightening of $SDCP$. In our computational testing on the benchmark data sets from the literature that contain significant overlap in consideration sets, we obtain the $CDLP$ value rapidly just by adding constraints with small values of σ.

4.1.1 A small example

We illustrate the procedure with a small example. Suppose we have a single resource with capacity 10 and five products that use this resource: a, b, c, d, e. Assume there are two segments and segment 1's consideration set is $C_1 = \{a, b, c\}$ and segment 2's consideration set is $C_2 = \{b, c, d, e\}$, so $C_1 \cap C_2 = \{b, c\}$. Assume $T = 1$ (only to reduce notation and size of the example).

The $1PC$ constraints are as follows. Corresponding to $S_{12} = \{b\}$, we have the constraint:

$$w^1_{\{b\}} + w^1_{\{b,c\}} + w^1_{\{a,b\}} + w^1_{\{a,b,c\}} = w^2_{\{b\}} + w^2_{\{b,c\}} + w^2_{\{b,d\}} + w^2_{\{b,e\}} + w^2_{\{b,c,d\}} + w^2_{\{b,d,e\}} + w^2_{\{b,c,e\}} + w^2_{\{b,c,d,e\}}.$$

Corresponding to $S_{12} = \{c\}$, we have the constraint:

$$w^1_{\{c\}} + w^1_{\{b,c\}} + w^1_{\{a,c\}} + w^1_{\{a,b,c\}} = w^2_{\{c\}} + w^2_{\{b,c\}} + w^2_{\{c,d\}} + w^2_{\{c,e\}} + w^2_{\{b,c,d\}} + w^2_{\{b,c,e\}} + w^2_{\{d,c,e\}} + w^2_{\{b,c,d,e\}}.$$

The $2PC$ constraints are as follows. Corresponding to $S_{12} = \{b, c\}$, we have the constraint:

$$w^1_{\{b,c\}} + w^1_{\{a,b,c\}} = w^2_{\{b,c\}} + w^2_{\{b,c,d\}} + w^2_{\{b,c,e\}} + w^2_{\{b,c,d,e\}}.$$

Since the problem is small, we formulate it as a single linear program combining ($SDCP_\kappa$) with ($Rgen_{\mathcal{L}}$) and the σPC constraints (which makes the y and z variables unnecessary):

$$
(\sigma PC) \quad
\begin{aligned}
\max_{(w^1_{S_1}),(w^2_{S_2})} \quad & \sum_{S_1 \subseteq C_1} \lambda_1 R^1_{S_1} w^1_{S_1} + \sum_{S_2 \subseteq C_2} \lambda_2 R^2_{S_2} w^2_{S_2} \\
\text{s.t.} \quad & \sum_{S_1 \subseteq C_1} \lambda_1 Q^1_{S_1} w^1_{S_1} + \sum_{S_2 \subseteq C_2} \lambda_2 Q^2_{S_2} w^2_{S_2} \leq 10 \\
& \sum_{S_1 \subseteq C_1} w^1_{S_1} \leq 1 \\
& \sum_{S_2 \subseteq C_2} w^2_{S_2} \leq 1 \\
& w^1_{\{b\}} + w^1_{\{b,c\}} + w^1_{\{a,b\}} + w^1_{\{a,b,c\}} = \\
& \qquad w^2_{\{b\}} + w^2_{\{b,c\}} + w^2_{\{b,d\}} + w^2_{\{b,e\}} + w^2_{\{b,c,d\}} + w^2_{\{b,d,e\}} + w^2_{\{b,c,e\}} + w^2_{\{b,c,d,e\}} \\
& w^1_{\{c\}} + w^1_{\{b,c\}} + w^1_{\{a,c\}} + w^1_{\{a,b,c\}} = \\
& \qquad w^2_{\{c\}} + w^2_{\{b,c\}} + w^2_{\{c,d\}} + w^2_{\{c,e\}} + w^2_{\{b,c,d\}} + w^2_{\{b,c,e\}} + w^2_{\{d,c,e\}} + w^2_{\{b,c,d,e\}} \\
& w^1_{\{b,c\}} + w^1_{\{a,b,c\}} = w^2_{\{b,c\}} + w^2_{\{b,c,d\}} + w^2_{\{b,c,e\}} + w^2_{\{b,c,d,e\}} \\
& w^1_{S_1}, w^2_{S_2} \geq 0.
\end{aligned}
\tag{13}
$$

4.1.2 Size of the problem

We show that the size of the problem with all the product constraints added is polynomial for a fixed size of the consideration sets. Assume homogenous arrival rates so we can aggregate the time periods into one. Let $\sigma_{\max} = \max_{l \in L} |C_l|$, i.e., the size of the largest segment consideration set. The size of $SDCP$, when written out as a single linear program (i.e. folding in ($Rgen$) directly in the $SDCP$ formulation) has at most $L2^{\sigma_{\max}}$ columns and $(m + L)$ rows corresponding to the m resource capacities and the L time constraints. The maximum number of product constraints that we can add are then at most $\binom{L}{2} 2^{\sigma_{\max}}$ corresponding to the intersection of consideration sets for all pairs of segments, and for every pair the fact that there are at most $2^{\sigma_{\max}}$ subsets in the intersection—still polynomial for a fixed size of the consideration sets. So if the maximum consideration set size is ~ 10, we are adding *at most* $1000 \times \binom{L}{2}$.

In practice it is quite unlikely that every pair of segments have overlapping consideration sets—for instance, in the airline context, segments are defined for each origin-destination pair, and their consideration sets are the routes they consider, so the overlap is rather limited. If memory and computational resources permit, given the power of modern linear programming solvers such as CPLEX or GUROBI (in a 64-bit operating system), we can even solve the entire problem ($SDCP$ and the product constraints) as a single linear program for a few hundred segments.

For larger problems or for limited computational resources, we can resort to keeping σ small or generating the constraints on the fly. For instance, if we are taking only $2PC$ constraints, i.e., $\sigma = 2$, then we are adding at most $\binom{\sigma_{\max}}{2} \times \binom{L}{2}$ constraints. Finally, if we have non-homogenous arrival rates, say represented by a piece-wise linear curve, all the above problem sizes are multiplied by the number of break-points in the piece-wise linear curve.

4.2 Enhanced κ-segment deterministic concave program ($ESDCP_\kappa$)

In this section we describe our second method that is a natural tightening of ($SDCP$). Consider $SDCP_\kappa$ as given in (11) in which, for a fixed value of κ and $\mathcal{L}$ with $|\mathcal{L}| = \kappa$, and a vector of capacities assigned to segment l, $\vec{y}_{lt}$, we had defined:

$$
\begin{aligned}
R^*_{\mathcal{L}}(\vec{y}_{lt}) = \quad & \max \quad \sum_{l \in \mathcal{L}} \sum_{S_l \subseteq C_l} \lambda_l R^l(S_l) w^l_{S_l} \\
& \text{s.t.} \quad \sum_{S_l \subseteq C_l} \lambda_l \vec{Q}^l(S_l) w^l_{S_l} \leq \vec{y}_{lt} \quad \forall l \in \mathcal{L}, \\
(Rgen_{\mathcal{L}}) \quad & \qquad \sum_{S_l \subseteq C_l} w^l_{S_l} \leq 1 \quad \forall l \in \mathcal{L}, \\
& \qquad w^l_{S_l} \geq 0, \ \forall l \in \mathcal{L}, \forall S_l \subseteq C_l.
\end{aligned}
$$

In the above generating mathematical program, we allow different offer sets S_l to be offered to the different segments. Our idea now is to tighten $SDCP_\kappa$ by using a different generating mathematical program that forces the use of the same offer set for all the segments. To reduce notation, we describe this new generating mathematical program for segment pairs, i.e., $\kappa = 2$; the general case should be transparent from the description. For every pair of segments (k_1, k_2) where $k_1 < k_2$ and a set of vectors of assigned capacities $\vec{y}_{k_1}, \vec{y}_{k_2}$, define

$$
\begin{aligned}
R^*_{k_1,k_2}(\vec{y}_{k_1 t}, \vec{y}_{k_2 t}) = \quad & \max \quad \sum_{S \subseteq C_{k_1} \cup C_{k_2}} (\lambda_{k_1} R_{k_1}(S_{k_1}) + \lambda_{k_2} R_{k_2}(S_{k_2})) w_S \\
& \text{s.t.} \quad \sum_{S \subseteq C_{k_1} \cup C_{k_2}} \left[\lambda_{k_1} \vec{Q}_{k_1}(S_{k_1}) + \lambda_{k_2} \vec{Q}_{k_2}(S_{k_2})\right] w_S \leq \vec{y}_{k_1 t} + \vec{y}_{k_2 t} \\
(Rgen_{(k_1,k_2)}) \quad & \qquad \sum_{S \subseteq C_{k_1} \cup C_{k_2}} w_S \leq 1 \\
& \qquad w_S \geq 0, \ \forall S \subseteq C_{k_1} \cup C_{k_2}
\end{aligned}
\tag{14}
$$

We call this level-κ formulation the enhanced κ-segment deterministic concave program ($ESDCP_\kappa$). Thus we can fine-tune the formulation for different values of κ, and we always maintain an upper bound on the dynamic program, in fact on the $CDLP$, and can choose the level to suit the network and computational resources.

Notice that $R^*_{k,l}(\vec{y}_{kt}, \vec{y}_{lt})$ is a concave function of the variables $\vec{y}_{kt}, \vec{y}_{lt}$. We add the following constraint to (11) for all pairs (k_1, k_2) of segments:

$$z_{k_1t} + z_{k_2t} - R^*_{k_1,k_2}(\vec{y}_{k_1t}, \vec{y}_{k_2t}) \leq 0 \tag{15}$$

and call the resulting formulation $ESDCP_\kappa$. We solve the generating concave program $(Rgen_{(k_1,k_2)})$ on the fly (and in parallel) and replace the constraints (15) by linear subgradient constraints (the dual solution of $R^*_{k_1,k_2}$ is a subgradient from linear programming theory)

$$z_{k_1t} + z_{k_2t} - (\vec{\pi}^k_{tk_1k_2})^\top(\vec{y}_{k_1} + \vec{y}_{k_2}) \leq \mu^k_{tk_1k_2}, \tag{16}$$

where $(\vec{\pi}^k_{tk_1k_2}, \mu^k_{tk_1k_2})$ is the dual solution to $R^*_{k_1,k_2}(\vec{y}_{k_1t}, \vec{y}_{k_2t})$.

To motivate $ESDCP_\kappa$, consider a simple situation where there are exactly two segments ($L = 2$) with consideration sets C_1 and C_2. $ESDCP_\kappa$ is then equivalent to $CDLP$, just written slightly differently. Now if the network naturally has a partition of the segments so that the consideration sets of segments in two different elements of the partition do not overlap (or have scarce overlap), then our formulation would exploit it as follows: We assign a capacity vector to each element of the partition and then, for a fixed set of capacity vectors, try to determine the optimal revenue from the assignment. For instance, if each element of the partition has exactly two segments, then we do recover the $CDLP$ as pointed out earlier.

For the general case, where each element of the partition has multiple segments, we can still solve the $ESDCP_2$ as an approximation. Of course, we can strengthen the formulation by defining generating concave programs for triplets of segments and so on. For a κ-tuple of segments $\{k_1, \ldots, k_\kappa\} \subseteq \{1, \ldots, L\}$ we define generating concave programs $R^*_{\{k_1,\ldots,k_\kappa\}}(\vec{y}_{k_1t}, \ldots, \vec{y}_{k_\kappa t})$ analogous to (14) and incorporate the corresponding constraints as in (15) for the κ-tuple of variables. If we do not have an idea of the partition, we just solve it for all pairs of segments or all κ-tuples in general. No matter to what depth we solve the problem, at every stage, we are assured of an upper bound on the dynamic program. In general this upper bound is weaker than the $CDLP$ bound.

Proposition 3. *$ESDCP_\kappa$ has an objective value greater than or equal to $CDLP$.*

Proof

We prove for $\kappa = 2$; the general case follows identically. Let w_S be a solution to $(CDLP)$. For every segment

l, define

$$\vec{y}_{lt} = \lambda_l \sum_S \vec{Q}_l(S_l) \frac{w_S}{T}.$$

We verify $ESDCP_\kappa$ with vectors $\vec{y}_{lt}$ has an objective value same as $(CDLP)$. The vectors $\vec{y}_{lt}$ satisfy $\sum_{t=1}^{T} \sum_{l=1}^{L} \vec{y}_{lt} \le \vec{c}_0$ as these are the same constraints as those of $(CDLP)$. Next, notice that by construction $\bar{w}_S := \sum_{\{S' | S' \cap (C_{k_1} \cup C_{k_2}) = S\}} w_{S'}$, for all $S \subseteq C_{k_1} \cup C_{k_2}$ is a feasible solution to $(Rgen_{(k_1,k_2)})$, so we conclude $ESDCP_\kappa \ge CDLP$.

□

5 Numerical results

The $(CDLP)$ is usually implemented to produce an estimate of the marginal value of capacity for each resource, and subsequently to decompose the network problem into a collection of single-resource problems. There are numerous studies that analyze the revenue performance of this decomposition process, see for example Zhang and Adelman (2009). Our objective is to solve the $(CDLP)$ for overlapping segments (or approximate it closely), so the revenue performance will be identical to $(CDLP)$ if we achieve the same value, and if we approximate it closely, be comparable. So our computations are to show how well our product constraints tighten $SDCP$ in the case of overlapping segment consideration sets, and in the computational times to achieve this tightening compared to alternate approaches. The experiments in §5.2.1–5.2.3 were programmed in Matlab R2009b with Tomlab R7 /CPLEX 11.2 on a desktop PC running MS Windows XP with a Pentium 4 2GHz CPU and 1GB RAM. The large network experiments of §5.2.4 used Matlab R2011a with CPLEX 12.2 on a machine with Intel Xeon 2.67 GHz CPU and 6G RAM. The data sets and the Matlab code can be obtained at `http://go.warwick.ac.uk/astrauss/research/code`.

5.1 Overview of the tested methods

We conduct a numerical study on various test networks where we compare the values resulting from the following (time-aggregated) approaches:

- $CDLP$: Defined in §3.1. As proposed by Bront et al. (2009), we use their pricing heuristic to identify new columns; if it does not find any more columns, then we use their mixed integer programming formulation until optimality is reached.

- $SDCP$: Segment-based deterministic concave program as defined in §3.2.
- σPC: $SDCP$ with product constraints as defined in §4.1. In method σPC we add product constraints of the form (12) only for subsets $|S_{lk}| \leq \sigma$. We generally use $\sigma \in \{1, 2\}$ and try $\sigma = 3$ only when column generation does not solve $CDLP$ or the gap between $2PC$ and $CDLP$ is significant.
- $ESDCP_\kappa$: The procedure described in §4.2.

We add product constraints for just pairs of products in the intersections of the considerations sets ($2PC$) as in all but one case (where $3PC$ gets $CDLP$ value) we obtain the $CDLP$ value and need not consider larger subsets. Likewise we test $ESDCP_\kappa$ with κ at most 2.

5.2 Test networks

We use the same test networks as in Liu and van Ryzin (2008) and Bront et al. (2009), where different scenarios were obtained by scaling the capacities by a factor $\alpha \in \{0.4, 0.6, 1, 1.2, 1.4\}$. For each of these scenarios, different no-purchase weights v_0 are applied to vary demand. The probabilities are derived from the weights by using the MNL model for each of the segments exactly as in Bront et al. (2009).

5.2.1 Parallel-Flights example

The first network example consists of three parallel flight legs as depicted in Figure 1 with initial leg capacity 30, 50 and 40, respectively. On each flight there is a low and a high fare class L and H, respectively, with fares as specified in Table 1. We define four customer segments in Table 2; note that we do not give the preference values for the no-purchase option at this point. This is because we consider various scenarios of this network by varying both the vector of no-purchase preferences and the network capacity. The sales horizon consists of 300 time periods. In Table 3 we report upper bounds on the optimal expected revenue

Product	Leg	Class	Fare
1	1	L	400
2	1	H	800
3	2	L	500
4	2	H	1,000
5	3	L	300
6	3	H	600

Table 1: Product definitions for Parallel-Flights Example.

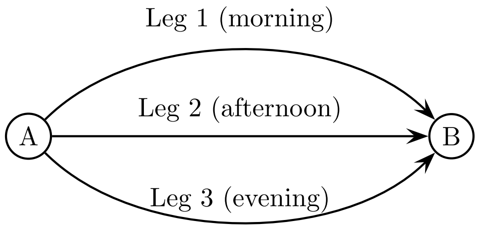

Figure 1: Parallel-Flights Example.

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 2: Segment definitions for Parallel-Flights Example.

obtained from our various approaches. The product constraints are very successful; 2PC obtains the $CDLP$ objective value in all instances, and even 1PC is already close to $CDLP$.

α	v_0	$CDLP$	$2PC$	$1PC$	$ESDCP_2$	$SDCP$
0.6	[1,5,5,1]	56,884	56,884	57,338	57,556	58,755
	[1,10,5,1]	56,848	56,848	57,316	57,546	58,755
	[5,20,10,5]	53,820	53,820	53,839	54,047	54,684
0.8	[1,5,5,1]	71,936	71,936	72,025	72,650	73,870
	[1,10,5,1]	71,795	71,795	71,865	72,608	73,870
	[5,20,10,5]	61,868	61,868	61,898	62,302	63,440
1.0	[1,5,5,1]	79,156	79,156	79,373	82,188	85,424
	[1,10,5,1]	76,866	76,866	77,069	79,938	83,377
	[5,20,10,5]	63,256	63,256	63,256	64,036	65,848
1.2	[1,5,5,1]	80,371	80,371	80,371	83,130	88,332
	[1,10,5,1]	78,045	78,045	78,045	80,880	86,333
	[5,20,10,5]	63,296	63,296	63,296	64,339	66,648
1.4	[1,5,5,1]	81,067	81,067	81,067	83,130	88,621
	[1,10,5,1]	78,817	78,817	78,817	80,880	86,355
	[5,20,10,5]	63,337	63,337	63,337	64,642	66,841

Table 3: Upper bounds for Parallel-Flights Example.

5.2.2 Small-Network example

Next, we test the policies on a network with seven flight legs as depicted in Figure 2. In total, 22 products are defined in Table 4 and the network capacity is $\vec{c}_0 = [100, 150, 150, 150, 150, 80, 80]$, where c_{0i} is the initial seat capacity of flight leg i. In Table 5, we summarize the segment definitions according to desired origin-destination (O-D), price sensitivity and preference for earlier flights. The booking horizon has $\tau = 1000$ time periods.

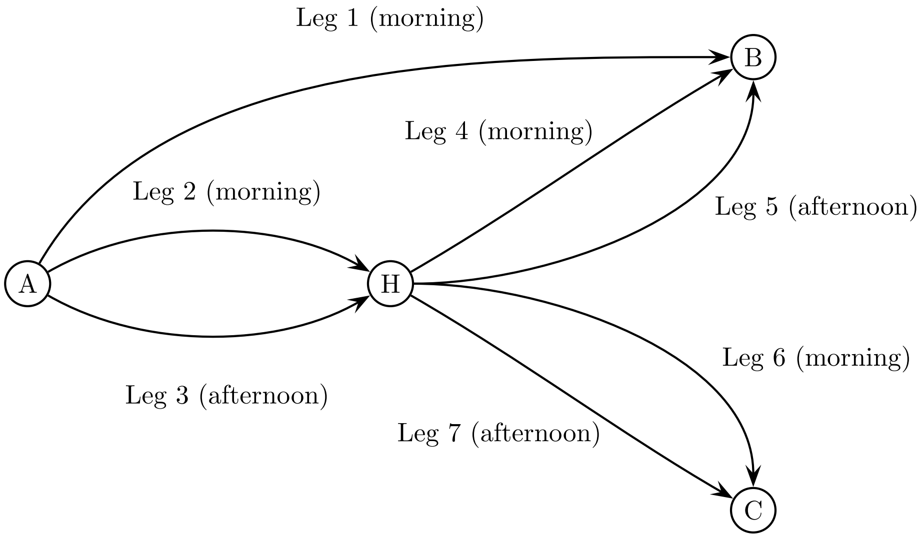

Figure 2: Small-Network example.

Class = H			Class = L		
Product	Legs	Fare	Product	Legs	Fare
1	1	1,000	12	1	500
2	2	400	13	2	200
3	3	400	14	3	200
4	4	300	15	4	150
5	5	300	16	5	150
6	6	500	17	6	250
7	7	500	18	7	250
8	2,4	600	19	2,4	300
9	3,5	600	20	3,5	300
10	2,6	700	21	2,6	350
11	3,7	700	22	3,7	350

Table 4: Product definitions for Small-Network Example

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

Table 5: Segment definitions for Small-Network Example

The upper bound results for the Small-Network example in Table 6 look a bit different from the Parallel-Flights case in that $ESDCP_2$ achieves the $CDLP$ value in all instances. This is due to the fact that each product is being considered by exactly two customer segments, and from the definition of $ESDCP_2$ it follows that this approach is equivalent to $CDLP$ in this situation. The product constraints perform again quite well: $2PC$ equals $CDLP$ in all except one instance. This instance $\alpha = 0.8$, $v_0 = [1, 5 \ldots]$ is the only one where $2PC$ does not equal $CDLP$; however, $3PC$ does deliver the $CDLP$ solution 266,934.

α	v_0	$CDLP$	$2PC$	$1PC$	$ESDCP_2$	$SDCP$
	[1,5]	215,793	215,793	215,793	215,793	216,649
0.6	[5,10]	200,515	200,515	201,294	200,515	206,392
	[10,20]	170,137	170,137	170,265	170,137	173,948
	[1,5]	266,934	266,949	268,842	266,934	272,719
0.8	[5,10]	223,173	223,173	223,536	223,173	230,393
	[10,20]	188,574	188,574	188,657	188,574	193,464
	[1,5]	281,967	281,967	282,078	281,967	296,513
1.0	[5,10]	235,284	235,284	235,446	235,284	245,226
	[10,20]	192,038	192,038	192,094	192,038	198,636
	[1,5]	284,772	284,772	285,052	284,772	301,773
1.2	[5,10]	238,562	238,562	238,562	238,562	248,728
	[10,20]	192,373	192,373	192,373	192,373	198,914
	[1,5]	287,076	287,076	287,357	287,076	305,329
1.4	[5,10]	238,562	238,562	238,562	238,562	249,372
	[10,20]	192,373	192,373	192,373	192,373	198,914

Table 6: Upper bounds for Small-Network example

5.2.3 Hub & spoke example

Consider the Hub & Spoke network in Figure 3. It has eight flight legs, one hub and four spokes. Each flight i has initial capacity $c_i = 200$ and the booking horizon is divided into $\tau = 2000$ time periods. There are 80 products in total which we define in Table 7 in the following way: product 1 corresponds to the trip ATL-BOS using leg 3 in class Y, product 4 is ATL-BOS in class Q, product 5 is BOS-ATL using leg 4 in class Y and so on. Definitions of the 40 customer segments for this example can be found in Table 8.

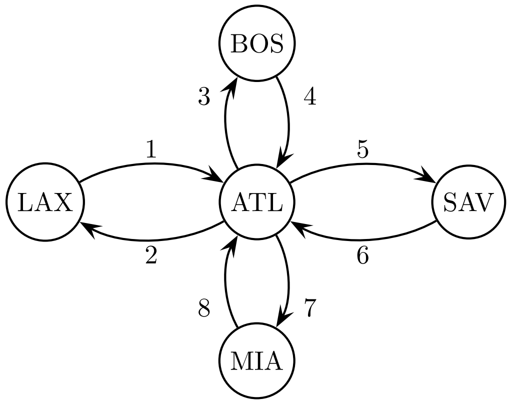

Figure 3: Hub & Spoke Network example.

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

Table 7: Product definitions for hub-and-spoke Example.

We report upper bounds on the optimal expected revenue in Table 9. The product constraints $2PC$ obtain $CDLP$ value in all instances.

5.2.4 Two-hub network

We consider a hub and spoke network of the type shown in Figure 4. There are two hubs H1 and H2 connected with two flights at 11am and 3pm in each direction, and each hub is connected to B spokes each. From each spoke leave two flights to the adjacent hub at 9am and 1pm, and two flights return at 11am and 3pm. The spokes around hub H1 (H2) are labeled from 1 to B (from $B+1$ to $2B$).

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/AT 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 example.

α	v_0	$CDLP$	$2PC$	$1PC$	$ESDCP_2$	$SDCP$
	[1,5]	163,897	163,897	163,952	163,897	176,808
0.6	[5,10]	132,674	132,674	132,674	132,674	144,249
	[10,20]	111,897	111,897	111,897	111,897	122,932
	[1,5]	177,384	177,384	177,978	177,384	199,682
0.8	[5,10]	146,338	146,338	146,641	146,338	164,037
	[10,20]	122,464	122,464	122,575	122,464	138,752
	[1,5]	187,270	187,270	189,294	187,270	219,671
1.0	[5,10]	156,243	156,243	157,082	156,243	180,880
	[10,20]	128,386	128,386	128,389	128,386	143,723
	[1,5]	195,269	195,269	198,923	195,269	236,739
1.2	[5,10]	160,206	160,206	160,674	160,206	189,955
	[10,20]	128,448	128,448	128,448	128,448	143,723
	[1,5]	197,113	197,113	201,894	197,113	246,768
1.4	[5,10]	160,453	160,453	160,818	160,453	189,955
	[10,20]	128,448	128,448	128,448	128,448	143,723

Table 9: Upper bounds for the hub-and-spoke example

All direct flights between a spoke and a hub are short-haul flights and those between hubs are long-haul. Depending on the number of spokes per hub, B, the network consists of $8B + 4$ flight legs. There are

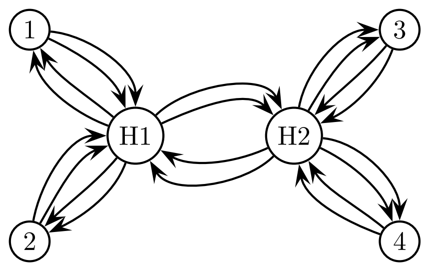

Figure 4: Two-hub network example with two hubs and $B = 2$ spokes each.

$4B^2 + 6B + 2$ origin-destination pairs ($4B$ between spoke and hub around one hub, 2 between hubs, $2B^2$ spoke to spoke via 2 hubs, $2B(B-1)$ spoke to spoke via one hub, $2B$ hub to hub to spoke, and $2B$ spoke to hub via another hub).

There are $8B^2 + 10B + 4$ possible itineraries ($8B$ between spoke and hub around one hub, 4 between hubs, $2B^2$ between spoke and spoke via 2 hubs, $6B(B-1)$ between spoke and spoke via 1 hub, $2B$ hub to hub to spoke, $6B$ spoke to hub to hub). For example, the only itinerary between spoke 1 and spoke $(B+1)$ is the 9am flight 1→ H1, the 11am flight H1→ H2, and the 3pm flight H2→ $(B+1)$. Other origin-destination pairs can have up to three possible itineraries, for example going from spoke 1 to H2, or to B.

For each itinerary there are five booking classes Y, M, Q, G and T; hence we have $40B^2 + 50B + 20$ products in total. The fares are sampled from a Poisson distribution with mean depending on the type of itinerary as reported in Table 10. If the fares are not in the order $Y > M > Q > G > T$, then we re-sample until that order is obtained. For each OD pair, there are four customer segments. Customers

Itinerary Type	Y	M	Q	G	T
short-haul with 1 leg	100	90	60	40	30
short-haul with 2 legs	200	180	120	80	60
long-haul with 1 leg	300	270	180	120	90
long-haul with 2 legs	400	360	240	160	120
long-haul with 3 legs	500	450	300	200	150

Table 10: Mean fares for different itinerary types and booking classes.

from each segment consider purchasing all possible itineraries for the desired OD pair, but each segment only considers a subset of booking classes. Each segment considers either $s = 2$, $s = 3$ or $s = 4$ booking classes (s is sampled from a uniform distribution). Which classes these are depends on the type of segment: the two business segments always consider Y and M, and additional classes Q and G depending on the drawn sample. For example, if $s = 3$ for such a segment, then it considers {Y,M,Q} on all itineraries for the desired OD pair. The first leisure segment considers always Q and G class, and additionally T, and M depending on the drawn sample s. Likewise, the second leisure segment always considers G and T class, and additionally Q and M depending on the drawn sample s. The MNL preference values for each considered booking class on any itinerary are sampled from a Poisson distribution with a mean for each product j given by "round($\gamma \exp(\beta r_j)) + 1$", with ($\gamma$,$\beta$) defined in Table 11. The preference values v_0 for the non-purchase option, denoted by 0, are defined for all four segments for each OD pair and stated with the results. The

Type	Description	β	γ
1	Business, insensitive	-0.001	15
2	Business, insensitive	-0.003	20
3	Leisure, sensitive	-0.006	20
4	Leisure, very sensitive	-0.01	20

Table 11: Types of customer segments for every OD pair. Parameters β and γ define the mean of preference value distribution.

arrival rates for each segment is constructed by defining a vector $b = [1, 2, 4, 5, \ldots, 1, 2, 4, 5] \in \mathbb{Z}^L$ and setting $\lambda = (0.7/\vec{1}^T b). * b$. This means, for example, that an arrival of a customer of segment 4 is five times as likely than an arrival of a segment 1 customer for any OD pair. There are 4,000 time periods.

For the network with $B = 4$ ($B = 8$) spokes per hub, all short-haul flight legs have a capacity of 70 (40) seats, and all long-haul flight legs have capacity of 120 (70) seats. These capacities are jointly scaled up or down via a factor $\alpha \in \{0.6, 0.8, 1.0, 1.2, 1.4\}$ in order to observe the effect of varied network load.

$CDLP$ was solved with column generation using the heuristic of Bront et al. (2009). We use their mixed integer programming formulation of the column generation subproblem if the heuristic cannot identify any additional columns any more. The column generation process uses the following stopping criterion: stop if reduced cost is less or equal to $10^{-8} *$ (current restricted objective + reduced cost).

B	Legs	OD pairs	Itineraries	Products	Segments
4	36	90	172	860	360
8	68	306	596	2980	1224

Table 12: Two-hub network specification.

α	v_0	$CDLP^*$	$3PC$	$2PC$	$1PC$	$SDCP$
	[2,5,10,15]	190,760	190,804	190,804	190,874	196,123
0.6	[5,10,15,20]	183,291	183,314	183,314	183,349	187,725
	[10,15 20,20]	177,454	177,457	177,457	177,472	181,062
	[2,5,10,15]	231,498	231,508	231,508	231,604	239,532
0.8	[5,10,15,20]	219,458	219,465	219,465	219,479	225,766
	[10,15 20,20]	210,502	210,506	210,506	210,519	214,820
	[2,5,10,15]	261,334	261,399	261,399	261,457	271,337
1	[5,10,15,20]	243,324	243,331	243,331	243,342	250,288
	[10,15 20,20]	229,808	229,810	229,810	229,813	235,642
	[2,5,10,15]	281,628	281,733	281,733	281,749	292,363
1.2	[5,10,15,20]	259,528	259,538	259,538	259,545	268,458
	[10,15 20,20]	245,308	245,310	245,310	245,315	252,505
	[2,5,10,15]	296,527	296,546	296,546	296,586	309,249
1.4	[5,10,15,20]	272,615	272,618	272,618	272,628	283,250
	[10,15 20,20]	256,902	256,905	256,905	256,906	264,364

Table 13: Upper bounds for two-hub network example with $B = 4$ spokes per hub. $CDLP^*$: Restricted objective of $CDLP$ master problem once stopping criterion was met.

α	v_0	$\Delta 3PC$ (%)	$\Delta 2PC$ (%)	$\Delta 1PC$ (%)	$\Delta SDCP$ (%)
	[2,5,10,15]	0.02	0.02	0.06	2.81
0.6	[5,10,15,20]	0.01	0.01	0.03	2.42
	[10,15 20,20]	0.00	0.00	0.01	2.03
	[2,5,10,15]	0.00	0.00	0.05	3.47
0.8	[5,10,15,20]	0.00	0.00	0.01	2.87
	[10,15 20,20]	0.00	0.00	0.01	2.05
	[2,5,10,15]	0.02	0.02	0.05	3.83
1	[5,10,15,20]	0.00	0.00	0.01	2.86
	[10,15 20,20]	0.00	0.00	0.00	2.54
	[2,5,10,15]	0.04	0.04	0.04	3.81
1.2	[5,10,15,20]	0.00	0.00	0.01	3.44
	[10,15 20,20]	0.00	0.00	0.00	2.93
	[2,5,10,15]	0.01	0.01	0.02	4.29
1.4	[5,10,15,20]	0.00	0.00	0.00	3.90
	[10,15 20,20]	0.00	0.00	0.00	2.90

Table 14: ΔX stands for $(X - CDLP^*)/CDLP^*$. Two-hub network example with $B = 4$ spokes per hub. $CDLP^*$: Restricted objective of $CDLP$ master problem once stopping criterion was met.

α	v_0	$CDLP$	$3PC$	$2PC$	$1PC$	$SDCP$
0.6	[2,5,10,15]	165,209⋄	169,973	169,973	170,040	175,384
	[5,10,15,20]	156,642⋄	162,371	162,371	162,404	166,885
	[10,15 20,20]	151,043⋄	157,308	157,308	157,314	160,963
0.8	[2,5,10,15]	196,151⋄	199,730	199,730	199,824	208,380
	[5,10,15,20]	185,342⋄	188,992	188,992	189,039	195,480
	[10,15 20,20]	178,338⋄	182,596	182,596	182,604	187,369
1	[2,5,10,15]	216,972⋄	219,921	219,922	220,037	229,451
	[5,10,15,20]	204,982⋄	208,221	208,221	208,263	215,396
	[10,15 20,20]	197,756⋄	200,904	200,904	200,930	206,411
1.2	[2,5,10,15]	235,190⋄	236,060	236,062	236,217	246,050
	[5,10,15,20]	222,498⋄	222,644	222,644	222,714	230,367
	[10,15 20,20]	214,136⋄	214,184	214,184	214,215	220,027
1.4	[2,5,10,15]	249,299*	249,324	249,324	249,474	259,483
	[5,10,15,20]	234,342*	234,348	234,348	234,413	242,346
	[10,15 20,20]	225,072*	225,080	225,080	225,108	230,700

Table 15: Upper bounds for two-hub network example with $B = 8$ spokes per hub. $CDLP$: restricted objective of $CDLP$ master problem either (*) once stopping criterion was met, or (⋄) once run time exceeded 10 hours. Hence, optimal $CDLP$ value is unknown for cases marked ⋄ and comparison with other methods is not possible.

Tables 13 and 14 show the upper bound values and percentage comparison with $CDLP$ for the two-hub network with 4 spokes per hub. As for the other networks, even $1PC$ achieves near $CDLP$ value. Table 15 gives the upper bound values for the 8-spoke configuration, but comparison with $CDLP$ is difficult as column generation had to be stopped after 10 hours for many of the instances. The values of $1PC$ is very close for the $\alpha = 1.4$ configuration where we have $CDLP$ values. $3PC$ values are in general identical to that of $2PC$. One possible explanation is that there is not much room for improvement between $2PC$ and $CDLP$ (0% to 0.04%) and further minuscule improvements can be achieved, if at all (as $CDLP$ is a NP-hard problem), only by adding all cuts of the form σPC.

5.3 Run times

The main motivation for the methods discussed in this article is to overcome the numerical difficulties inherent to the ($CDLP$) formulation for overlapping segments. When the sets C_l are small, there are only few subsets $S_l \subset C_l$ for each segment, we can even solve the ($SDCP$) (with or without product constraints) as a single linear program. In Tables 16, 17, 18, we compare the run times in CPU seconds of the different approaches on the networks for the small network and the large networks. We emphasize that the run times depend on the programming language, code efficiency, hardware, and the version of the linear programming package, but we believe the run times are indicative of their *relative* performance. There is considerable overlap across the segments, as can be seen in the gap between the $SDCP$ and $CDLP$ values for the networks.

When the considerations sets are small, as current linear programming packages can handle millions of variables and computer memory has become relatively cheap, we might even be able to solve (σPC) or ($ESDCP_\kappa$) for small σ, κ as a single linear program. In our computational tests, we run ($2PC$), ($1PC$) and ($SDCP$) as a single linear program, while $ESDCP_\kappa$ uses the dynamic generation of constraints as described above. The run run-times of ($SDCP$) with and without product constraints are significantly shorter than ($CDLP$) using column-generation. ($ESDCP_\kappa$) is slower but still significantly faster than column generation. The advantage of ($ESDCP_\kappa$) is that it may (as in one case in our computations) give a tighter bound than (σPC) for $\sigma = \kappa$ for small values of κ, as happens for the case of $\alpha = 0.8, v_0 = [1,5]$ in Table 6.

α	v_0	$CDLP$	$2PC$	$1PC$	$ESDCP_2$	$SDCP$
	[1,5]	15.92	0.28	0.28	2.43	0.16
0.6	[5,10]	11.79	0.28	0.27	2.80	0.16
	[10,20]	16.14	0.28	0.27	2.85	0.15
	[1,5]	23.72	0.29	0.28	2.44	0.16
0.8	[5,10]	18.95	0.36	0.31	2.95	0.19
	[10,20]	17.90	0.28	0.27	3.15	0.15
	[1,5]	24.85	0.29	0.29	2.76	0.15
1.0	[5,10]	7.27	0.33	0.27	3.12	0.16
	[10,20]	4.32	0.28	0.27	2.87	0.15
	[1,5]	9.40	0.34	0.27	2.18	0.15
1.2	[5,10]	4.55	0.28	0.27	2.84	0.15
	[10,20]	1.52	0.28	0.27	2.81	0.16
	[1,5]	1.71	0.28	0.27	2.19	0.15
1.4	[5,10]	1.69	0.28	0.27	2.90	0.16
	[10,20]	1.53	0.28	0.27	2.81	0.15

Table 16: Run times in CPU seconds for the hub-and-spoke network of §5.2.3. $CDLP$ run times are for solution by column-generation. $2PC$ and $ESDCP_2$ obtain the same values as $CDLP$ for all cases except one where we need to go to $3PC$.

5.4 Revenue simulations

In our numerical results, we have concentrated so far on comparing how well the various methods ($SDCP$, $2PC$, $ESDCP_2$ etc.) achieve the $CDLP$ objective value, even though they are all relaxations of the $CDLP$. In this section we perform a small simulation study to evaluate revenue performance of the various methods. Note that how one uses the optimization results (primal solution, dual solution, post-processing etc.) affect the results. For reference purposes, we use the optimization output the same way as it was used in Bront et al. (2009).

α	v_0	$CDLP^*$	$3PC$	$2PC$	$1PC$	$SDCP$
	[2,5,10,15]	3,199	71	64	58	47
0.6	[5,10,15,20]	3,747	71	63	59	48
	[10,15 20,20]	4,263	71	63	58	48
	[2,5,10,15]	3,353	71	63	58	47
0.8	[5,10,15,20]	4,696	71	63	58	47
	[10,15 20,20]	5,428	71	63	58	47
	[2,5,10,15]	4,054	71	63	58	47
1	[5,10,15,20]	3,682	71	61	58	48
	[10,15 20,20]	4,301	71	62	57	48
	[2,5,10,15]	3,445	71	62	57	48
1.2	[5,10,15,20]	3,411	71	62	57	47
	[10,15 20,20]	4,104	71	62	57	48
	[2,5,10,15]	3,016	71	61	57	47
1.4	[5,10,15,20]	3,218	71	61	57	47
	[10,15 20,20]	3,381	71	62	57	48

Table 17: Run times in seconds for the two-hub network of §5.2.4 with $B = 4$ spokes per hub. $CDLP^*$: run time to solve $CDLP$ by column generation until stopping criterion is met.

α	v_0	$CDLP$	$3PC$	$2PC$	$1PC$	$SDCP$
	[2,5,10,15]	36,044◇	644	518	483	255
0.6	[5,10,15,20]	36,016◇	653	512	481	254
	[10,15 20,20]	36,101◇	609	511	482	252
	[2,5,10,15]	36,000◇	584	508	480	253
0.8	[5,10,15,20]	36,108◇	745	496	474	252
	[10,15 20,20]	36,114◇	783	521	494	274
	[2,5,10,15]	36,170◇	633	498	473	251
1	[5,10,15,20]	36,054◇	646	494	471	251
	[10,15 20,20]	36,194◇	609	497	473	255
	[2,5,10,15]	36,134◇	720	494	469	252
1.2	[5,10,15,20]	36,233◇	651	491	466	251
	[10,15 20,20]	36,232◇	779	510	475	254
	[2,5,10,15]	9,897*	643	502	476	257
1.4	[5,10,15,20]	7,168*	576	484	465	251
	[10,15 20,20]	9,138*	650	499	484	274

Table 18: Run times in seconds for the two-hub network of §5.2.4 with $B = 8$ spokes per hub. $CDLP$: run time to solve $CDLP$ by column generation until either (*) stopping criterion is met, or (◇) run time exceeded 10 hours.

$CDLP$ by column generation takes considerable amount of time so we can run simulations only on small networks and we restrict ourselves to the Parallel-Flights example of §5.2.1 and the Small-Network example of §5.2.2. In each case we generated 2,000 demand sample paths (the sample paths are the same for all the methods tested) and used the dual values of the capacity constraints of the respective policy in a dynamic programming decomposition. At each point in time, the greedy heuristic presented in Bront et al. (2009) is used to obtain the offer set (based on an opportunity cost estimate using the value function approximation from the DP decomposition). For both examples, $SDCP$ occasionally outperforms all the methods, but is also more erratic, while revenue from $CDLP$, and $2PC$ that approximates it very closely, is more robust.

5.4.1 Parallel-Flights example

Table 20 reports the percentage average revenue improvement of policies $2PC$, $ESDCP_2$ and $SDCP$ over $CDLP$. The standard deviation of the revenue samples for each simulation can be found in Table 21.

We observe that 2PC achieves in all scenarios the same average revenue as $CDLP$; as outlined above,

		$CDLP$		$2PC$		$ESDCP_2$		$SDCP$	
α	v_0	Rev	LF	Rev	LF	Rev	LF	Rev	LF
0.6	[1,5,5,1]	55,967	0.98	55,967	0.98	54,708	0.96	54,708	0.96
	[1,10,5,1]	55,841	0.98	55,841	0.98	54,215	0.94	54,203	0.94
	[5,20,10,5]	51,360	0.95	51,360	0.95	51,551	0.96	51,943	0.97
0.8	[1,5,5,1]	69,358	0.96	69,358	0.96	66,493	0.91	69,828	0.97
	[1,10,5,1]	69,094	0.95	69,003	0.95	65,311	0.89	69,433	0.97
	[5,20,10,5]	60,046	0.90	60,046	0.90	59,995	0.91	59,943	0.91
1.0	[1,5,5,1]	76,936	0.95	76,936	0.95	77,331	0.95	76,782	0.94
	[1,10,5,1]	75,710	0.91	75,710	0.91	75,823	0.90	75440	0.89
	[5,20,10,5]	62,601	0.77	62,601	0.77	62,606	0.77	62,602	0.77
1.2	[1,5,5,1]	79,761	0.83	79,761	0.83	79,780	0.83	79,799	0.83
	[1,10,5,1]	77,499	0.78	77,499	0.78	77,505	0.78	77,505	0.78
	[5,20,10,5]	63,010	0.66	63,010	0.66	63,011	0.66	63,011	0.66
1.4	[1,5,5,1]	80,494	0.70	80,494	0.70	80,494	0.70	80,494	0.70
	[1,10,5,1]	78,223	0.66	78,223	0.66	78,223	0.66	78,223	0.66
	[5,20,10,5]	63,097	0.58	63,097	0.58	63,097	0.58	63,097	0.58

Table 19: Average revenue (Rev) and load factor (LF) results for Parallel-Flights example with 2000 demand sample paths.

this is due to the fact that ($2PC$) returns almost the same objective values as ($CDLP$), hence the value function approximation resulting from the dynamic programming decomposition is almost identical. The policy $ESDCP_2$ appears to be not as successful; in four out of 18 cases it performs between 2–5.5% worse than $CDLP$. Finally, the policy $SDCP$ underperforms in some of the crucial scenarios of medium capacity tightness (i.e., α around 0.6–1) by up to 3% with respect to $CDLP$. However, in two scenarios it did significantly better by improving 0.68% and 1.1% over $CDLP$. The results indicate that ($2PC$) can indeed be used to obtain policies with the similar revenue performance as ($CDLP$).

α	v_0	$\Delta\frac{2PC}{CDLP}(\%)$	$\Delta\frac{ESDCP_2}{CDLP}(\%)$	$\Delta\frac{SDCP}{CDLP}(\%)$
	[1,5,5,1]	0.00	-2.25	-2.25
0.6	[1,10,5,1]	0.00	-2.91	-2.93
	[5,20,10,5]	0.00	0.37	1.13
	[1,5,5,1]	0.00	-4.13	0.68
0.8	[1,10,5,1]	-0.13	-5.48	0.49
	[5,20,10,5]	0.00	-0.08	-0.17
	[1,5,5,1]	0.00	0.51	-0.20
1.0	[1,10,5,1]	0.00	0.15	-0.36
	[5,20,10,5]	0.00	0.01	0.00
	[1,5,5,1]	0.00	0.02	0.05
1.2	[1,10,5,1]	0.00	0.01	0.01
	[5,20,10,5]	0.00	0.00	0.00
	[1,5,5,1]	0.00	0.00	0.00
1.4	[1,10,5,1]	0.00	0.00	0.00
	[5,20,10,5]	0.00	0.00	0.00

Table 20: Percentage average revenue improvement over $CDLP$ for Parallel-Flights example.

5.4.2 Small-Network example

As we noted earlier, $CDLP$ and $ESDCP_2$ are equivalent for this example, hence both policies are identical. 2PC produces in all except for one scenario ($\alpha = 0.8$, [1,5]) the same upper bound as $CDLP$, and since the dual solution is also identical, the resulting policies arising from the dynamic programming decomposition deliver the same revenues (Table 22). The percentage average revenue improvement of each tested policy with respect to $CDLP$ is given in Table 23. The revenue performance of $SDCP$ relative to $CDLP$ is similar to our observations for the Parallel-Flights example. The standard deviations of the revenues are reported in Table 24.

6 Conclusions

In this paper, we have developed computationally attractive methods for approximating ($CDLP$) for the choice network RM problem with overlapping segments and small consideration sets; as the general problem is difficult even for the MNL model with few segments, this represents a promising line of attack for industries,

α	v_0	$CDLP$	2PC	$ESDCP_2$	$SDCP$
	[1,5,5,1]	1,799	1,799	5,956	3,586
0.6	[1,10,5,1]	1,925	1,925	3,853	3,904
	[5,20,10,5]	3,408	3,408	3,302	2,910
	[1,5,5,1]	3,877	3,877	5,219	3,182
0.8	[1,10,5,1]	3,917	3,982	5,358	3,414
	[5,20,10,5]	4,677	4,677	4,230	4,213
	[1,5,5,1]	4,329	4,329	4,814	4,879
1.0	[1,10,5,1]	5,235	5,235	5,412	5,682
	[5,20,10,5]	5,713	5,713	5,690	5,693
	[1,5,5,1]	6,071	6,071	6,066	6,127
1.2	[1,10,5,1]	6,140	6,140	6,138	6,138
	[5,20,10,5]	5,955	5,955	5,959	5,956
	[1,5,5,1]	6,458	6,458	6,458	6,458
1.4	[1,10,5,1]	6,478	6,478	6,478	6,478
	[5,20,10,5]	5,899	5,899	5,899	5,899

Table 21: Standard deviations of revenue simulations with 2000 sample paths for the Parallel-Flights Example.

		$CDLP$		$2PC$		$ESDCP_2$		$SDCP$	
α	v_0	Rev	LF	Rev	LF	Rev	LF	Rev	LF
	[1,5]	149,715	0.99	149,715	0.99	149,715	0.99	149,715	0.99
0.4	[5,10]	144,216	0.98	144,216	0.98	144,216	0.98	144,449	0.98
	[10,20]	134,395	0.96	134,395	0.96	134,395	0.96	133,424	0.95
	[1,5]	209,167	0.94	209,167	0.94	209,167	0.94	212,280	0.95
0.6	[5,10]	193,291	0.95	193,291	0.95	193,291	0.95	190,031	0.91
	[10,20]	167,545	0.95	167,545	0.95	167,545	0.95	167,290	0.93
	[1,5]	262,312	0.90	262,271	0.90	262,312	0.90	254,493	0.86
0.8	[5,10]	220,525	0.93	220,525	0.93	220,525	0.93	220,468	0.92
	[10,20]	185,754	0.89	185,754	0.89	185,754	0.89	185,274	0.89
	[1,5]	279,023	0.86	279,023	0.86	279,023	0.86	280,444	0.85
1.0	[5,10]	233,451	0.87	233,451	0.87	233,451	0.87	229,853	0.83
	[10,20]	191,345	0.81	191,345	0.81	191,345	0.81	191,148	0.79
	[1,5]	284,077	0.75	284,077	0.75	284,077	0.75	284,092	0.75
1.2	[5,10]	237,974	0.75	237,974	0.75	237,974	0.75	237,974	0.75
	[10,20]	192,107	0.71	192,107	0.71	192,107	0.71	192,107	0.71
	[1,5]	286,165	0.64	286,165	0.64	286,165	0.64	286,165	0.64
1.4	[5,10]	238,270	0.64	238,270	0.64	238,270	0.64	238,270	0.64
	[10,20]	192,120	0.61	192,120	0.61	192,120	0.61	192,120	0.61

Table 22: Average revenue (Rev) and load factor (LF) results for Small-Network example with 2000 demand sample paths.

α	v_0	$\Delta\frac{2PC}{CDLP}(\%)$	$\Delta\frac{ESDCP_2}{CDLP}(\%)$	$\Delta\frac{SDCP}{CDLP}(\%)$
	[1,5]	0.00	0.00	0.00
0.4	[5,10]	0.00	0.00	0.16
	[10,20]	0.00	0.00	-0.72
	[1,5]	0.00	0.00	1.49
0.6	[5,10]	0.00	0.00	-1.69
	[10,20]	0.00	0.00	-0.15
	[1,5]	-0.02	0.00	-2.98
0.8	[5,10]	0.00	0.00	-0.03
	[10,20]	0.00	0.00	-0.26
	[1,5]	0.00	0.00	0.51
1.0	[5,10]	0.00	0.00	-1.54
	[10,20]	0.00	0.00	-0.10
	[1,5]	0.00	0.00	0.01
1.2	[5,10]	0.00	0.00	-0.00
	[10,20]	0.00	0.00	0.00
	[1,5]	0.00	0.00	0.00
1.4	[5,10]	0.00	0.00	0.00
	[10,20]	0.00	0.00	0.00

Table 23: Percentage average revenue improvement over $CDLP$ for Small-Network example.

α	v_0	$CDLP$	$2PC$	$ESDCP_2$	$SDCP$
	[1,5,5,1]	2,040	2,040	2,040	2,040
0.4	[1,10,5,1]	2,232	2,232	2,232	2,478
	[5,20,10,5]	3,720	3,720	3,720	4,039
	[1,5,5,1]	5,020	5,020	5,020	4,284
0.6	[1,10,5,1]	5,446	5,446	5,446	6,872
	[5,20,10,5]	4,908	4,908	4,908	5,370
	[1,5,5,1]	6,266	6,314	6,266	7,705
0.8	[1,10,5,1]	5,666	5,666	5,666	6,391
	[5,20,10,5]	5,980	5,980	5,980	5,322
	[1,5,5,1]	6,848	6,848	6,848	7,738
1.0	[1,10,5,1]	6,505	6,505	6,505	6,973
	[5,20,10,5]	6,951	6,951	6,951	7,123
	[1,5,5,1]	7,932	7,932	7,932	7,947
1.2	[1,10,5,1]	7,545	7,545	7,545	7,545
	[5,20,10,5]	6,987	6,987	6,987	6,987
	[1,5,5,1]	8,077	8,077	8,077	8,077
1.4	[1,10,5,1]	7,816	7,816	7,816	7,816
	[5,20,10,5]	6,991	6,991	6,991	6,991

Table 24: Standard deviations of revenue simulations with 2,000 sample paths for the Small-Network example.

such as airline and retail, where the conditions apply. Using a formulation based on segments and their consideration sets, we add constraints that are easy to generate and highly effective—the methods obtain the same value as $CDLP$ in all the benchmark test instances, usually in a fraction of CPU time (Table 16) required for alternate approaches. Moreover, the formulation and the constraints operate at a high level of generality being applicable to a general discrete-choice model of demand, and of course for overlapping customer segments. Finally, we perform extensive numerical simulations to test the methods. Our results indicate that ($SDCP$) with product constraints can be very effective when segment consideration sets are small, as is often the case in many applications, and this strategy of starting with a looser relaxation than $CDLP$ ($SDCP$) and gradually adding constraints to tighten the formulation is a viable solution method.

References

Bodea, T., M. Ferguson, L. Garrow. 2009. Choice-based revenue management: Data from a major hotel chain. *Manufacturing and Service Operations Management* **11** 356–361.

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

Gallego, G., G. Iyengar, R. Phillips, A. Dubey. 2004. Managing flexible products on a network. Tech. Rep. TR-2004-01, Dept of Industrial Engineering, Columbia University, NY, NY.

Gallego, G., R. Ratliff, S. Shebalov. 2010. A general attraction model and an efficient formulation for the network revenue management problem. Tech. rep., Department of IEOR, Columbia University, available at `http://www.columbia.edu/~gmg2/`.

Hauser, J.R., B. Wernerfelt. 1990. An evaluation cost model of consideration sets. *Journal of Consumer Research* **16** 393–408.

Kök, G., M. L. Fisher, R. Vaidyanathan. 2009. *Retail supply chain management: Quantitative models and empirical studies*, chap. Assortment planning: Review of literature and industry practice. Springer, NY,NY, USA, 99–154.

Kunnumkal, S. 2011. Randomization approaches for the network revenue management with customer choice behavior. Tech. rep., Indian School of Business, Hyderabad, India.

Kunnumkal, S., H. Topaloglu. 2010. A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. *Production and Operations Management* **19**(5) 575–590.

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

Lussier, D. A., R. W. Olshavsky. 1979. Task complexity and contingent processing in brand choice. *Journal of Consumer Research* **6**(2) 154–165.

Meissner, J., A. K. Strauss. 2012. Network revenue management with inventory-sensitive bid prices and customer choice. *European Journal of Operational Research* **216**(2) 459–468.

Méndez-Díaz, I., J. Miranda Bront, G. Vulcano, P. Zabala. 2012. A branch-and-cut algorithm for the latent-class logit assortment problem. *Discrete Applied Mathematics* (forthcoming).

Payne, J. W. 1976. Task complexity and contingent processing in decision-making: An information search and protocol analysis. *Organizational Behavior and Human Performance* **16**(2) 366–387.

Rusmevichientong, P., D. Shmoys, H. Topaloglu. 2010. Assortment optimization with mixtures of logits. Tech. rep., School of IEOR, Cornell University.

Talluri, K. T. 2001. Airline revenue management with passenger routing control: A new model with solution approaches. *International Journal of Services Technology and Management* **2** 102–115.

Talluri, K. T. 2010. A randomized concave programming method for choice network revenue management. Tech. rep., Working Paper 1215, Department of Economics, Universitat Pompeu Fabra, Barcelona, Spain.

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

Talluri, K. T., G. J. van Ryzin. 2004b. *The Theory and Practice of Revenue Management*. Kluwer, New York, NY.

Wright, P., F. Barbour. 1977. *Multiple Criteria Decision Making*, chap. Phased decision strategies: Sequels to intial screening. TIMS: Studies in the Management Sciences, North-Holland, Amsterdam, Holland, 91–109.

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