A Randomized Concave Programming Method for Choice Network Revenue Management

Kalyan Talluri *

October 13, 2011

Abstract

Models incorporating more realistic models of customer behavior, as customers choosing from an offer set, have recently become popular in assortment optimization and revenue management. The dynamic program for these models is intractable and approximated by a deterministic linear program called the $CDLP$ which has an exponential number of columns. However, 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 we propose a new approach called $SDCP$ to solving $CDLP$ based on segments and their consideration sets. $SDCP$ is a relaxation of $CDLP$ and hence forms a looser upper bound on the dynamic program but coincides with $CDLP$ for the case of non-overlapping segments. If the number of elements in a consideration set for a segment is not very large ($SDCP$) can be applied to any discrete-choice model of consumer behavior. We tighten the $SDCP$ bound by (i) simulations, called the randomized concave programming (RCP) method, and (ii) by adding cuts to a recent compact formulation of the problem for a latent multinomial-choice model of demand ($SBLP+$). This latter approach turns out to be very effective, essentially obtaining $CDLP$ value, and excellent revenue performance in simulations, even for overlapping segments. By formulating the problem as a separation problem, we give insight into why $CDLP$ is easy for the MNL with non-overlapping considerations sets and why generalizations of MNL pose difficulties. We perform numerical simulations to determine the revenue performance of all the methods on reference data sets in the literature.

Key words. assortment optimization, randomized algorithms,network revenue management

1 Introduction and literature review

Revenue management 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. Sale is online, so the firm has to decide what products to offer (at a given price for each product), the tradeoff being selling too much at too low a price early and running out of capacity, or, rejecting too many low-valuation customers and ending up with excess unsold inventory.

*Kalyan Talluri, ICREA & Universitat Pompeu Fabra, Ramon Trias Fargas 25-27, 08005 Barcelona, Spain

In industries such as hotels and airlines the products consume bundles of different resources (multi-night stays, multi-leg itineraries) and the decision to accept or reject a particular product at a certain price depends on the future demands and revenues for all the resources used by the product and indirectly, on 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 [24] 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, initiated in Talluri and van Ryzin [23] for the single-resource problem. Many network RM extensions of such models (Gallego, Iyengar, Phillips, and Dubey [8], Liu and van Ryzin [14], Kunnumkal and Topaloglu [13], Zhang and Adelman [26], Meissner and Strauss [16], Bodea, Ferguson, and Garrow [4]) have recently been proposed. In many cases they are modifications of older methods proposed for network RM with the so-called independent class assumption.

The extension to the choice model of customer behavior however makes the approximations considerably more difficult to solve. The formulations have an exponential number of constraints and the solution strategy is to use column generation, but finding an entering column is computationally easy only in a limited number of cases.

In this paper we first give a segment-based deterministic concave-program ($SDCP$) upper bound to the underlying dynamic program (defined in §1.3), which is a relaxation that offers different offer sets to different customers, and that coincides with the $CDLP$ upper-bound for non-overlapping segments. We then tighten the bound in two different ways (i) By a simulation-based randomized concave programming (RCP) method, similar to the Randomized Linear Program (RLP) for the independent-class model ([22]) (ii) By adding valid inequalities to $SDCP$. Our cuts are a specialization of the ones developed in Meissner, Strauss, and Talluri [17] to the compact formulation of Gallego, Ratliff, and Shebalov [9] for the multinomial-logit choice model. The advantage of these cuts is that the space of the resulting program is exponential only in the number of products in the intersection of two segments' consideration sets, rather than the size of the consideration sets as in [17].

If the number of elements in a consideration set for a segment is not very large, both ($SDCP$) and (RCP) can be applied to any choice model whatsoever, expanding the models well beyond tractable-but-restrictive ones such as multinomial-logit. Small consideration sets can be justified in the airline setting where a segment's consideration set consists of choices (on one airline) for travel between an origin and destination, and typically there are only alternatives on a given date (Talluri [21]).

Another stream of literature that considers essentially the same mathematical problem in a different application context is assortment optimization for the retail industry (Kök, Fisher, and Vaidyanathan [12]). Network choice RM (the one considered in this paper) can be considered a dynamic assortment optimization problem with an additional network structure for the resources. For this reason many of the solution methodologies developed for network RM can be applied to the retail setting as well. Empirical studies in the marketing literature also motivate our assumption of small consideration sets; Hauser and Wernerfelt [11] report average consideration set sizes of 3 brands for deodorants, 4 brands for shampoos, 2.2 brands for air fresheners, 4 brands for laundry detergents and 4 brands for coffees.

To summarize, the contributions of this paper are as follows (i) We develop a new solution strategy for solving $CDLP$ based on segment consideration sets rather than column generation (ii) We tighten the formulation using randomization (RCP) and by adding cuts for the MNL choice model ($SBLP+$) to get close to the $CDLP$ value (iii) Give some insights as to why column-generation works for non-overlapping consideration sets and MNL and why it is difficult for any generalization of MNL (iv) Perform numerical experiments that show that $SBLP+$ runs extremely fast and should be scalable to industrial-size problems, giving the most robust revenues.

1.1 Notation

A product is a set of resources and a price. 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.

We assume that the booking horizon begins at time 0 and all the resources are consumed instantaneously at time T. Time is discrete and assumed to consist of T intervals, indexed by t. We make the standard assumption that the intervals are fine enough so that at most one customer arrives in each period.

The underlying network has I resources (indexed by i) and J products (indexed by j) of resources. Whenever it is clear from the context, we let J represent the set of products also (as in $j \in J$). 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 $\vec{c}_t$, 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).

1.2 Demand model

The demand model is a (latent, finite) segment-mixture model. We assume there are L underlying segments, each with distinct purchase behavior. Customers are assumed independent of each other, arrive randomly during a sale period and demand one unit of resource each. In each period, there is a customer arrival with probability λ, and 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 $\lambda = \sum_l \lambda_l$. Define $\vec{\lambda} = [\lambda_1, \ldots, \lambda_L]$. 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*, a subset of products $C_l \subseteq J$ that it considers for purchase. We assume this consideration set is known to the firm (by a previous process of estimation and analysis).

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. To simplify notation, we just assume that the null set, $\emptyset$, represents the no-purchase option, and it is always present in all offer sets. We clarify that when we write $j \in S$ in a summation or union, it does not include the null set; that is the indexing is over the products $1, \ldots, J$. The no-purchase option is indexed by 0 when necessary. We represent subsets of C_l by S_l. If the firm offers a set S

of products, the segment l customer would only consider the subset $S_l = C_l \cap S$.

The choice probabilities are given as follows: A segment-l customer purchases $j \in S$ with probability $P_{lj}(S)$. 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 model (§3 and §4).

The choice probabilities are assumed to satisfy, $P_{lj}(S) = P_{lj}(S \cap C_l)$, $\forall j \in S \cap C_l$ and $P_{lj}(S) = 0$, $\forall j \notin S \cap C_l$; i.e., a segment-$l$ customer is completely indifferent to a product outside his consideration set and his choice probabilities are not affected by products offered outside his consideration set. So whenever we specify probabilities for a segment l for a given offer-set S, we just write it with respect to S_l. Define the vector $\vec{P}_l(S) = [P_{l1}(S_l), \ldots, P_{ln}(S_l)]$.

Given a customer arrival, and an offer set S, the probability that the firm sells $j \in S$ is then given by $P_j(S) = \sum_l p_l P_{lj}(S_l)$. The probability of the no-purchase option is given by $P_0(S) = 1 - \sum_{j \in S} P_j(S)$. Define the vector $\vec{P}(S) = [P_1(S), \ldots, P_J(S)]$. Notice that $\vec{P}(S) = \sum_l p_l \vec{P}_l(S)$.

Define the m-vectors $\vec{Q}_l(S) = A\vec{P}_l(S)$ and $\vec{Q}(S) = A\vec{P}(S)$. Define the revenue functions $R_l(S) = \sum_{j \in S_l} r_j P_{lj}(S_l)$ and $R(S) = \sum_{j \in S} r_j P_j(S)$.

Define a segment-offer set 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.

In our notation and demand model we broadly follow Bront, Méndez-Díaz, and Vulcano [5] and Liu and van Ryzin [14]. We refer the reader to these papers for motivating examples behind the demand model.

1.2.1 Non-overlapping segments model

Liu and van Ryzin [14] show that their *CDLP* approximation is tractable for a model with MNL choice and non-overlapping segment consideration sets: for any two segments l and m, $C_l \cap C_m = \emptyset$.

The non-overlapping segment assumption can potentially be limiting in applications. For instance, in an airline context, Talluri [21] models the different itineraries between a city pair by a route-set. If say there are two types of customers, business and leisure, and we define a segment as type of customer and the origin-destination pair: A business customer might be considering just the shortest route, or the itinerary closest to his preferred time, whereas a leisure customer might consider both, looking for the cheapest flight. This would constitute overlapping consideration sets.

In the context of assortment optimization consideration sets are determined both by tastes as well as incomes and non-overlapping considerations sets would be a serious restriction.

As far as we know only Bront et al. [5] and Rusmevichientong, Shmoys, and Topaloglu [20] tackle the case of overlapping segments—they show that column-generation is NP-hard, and propose heuristics and a mixed-integer programming method for generating columns.

1.3 Dynamic program

The dynamic program (DP) to determine optimal controls is 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 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 $V_{T+1}(\vec{c}) = V_t(\vec{0}) = 0,\ \forall \vec{c}$. Let V^{DP} denote the optimal value of this dynamic program from 0 to T, for the given initial capacity vector $\vec{c}_0$.

2 Approximations and upper bounds

2.1 Choice Deterministic Linear Program ($CDLP$)

The choice deterministic linear program ($CDLP$) defined in Gallego et al. [8] and Liu and van Ryzin [14] is as follows:

$$\begin{aligned} V^{CDLP} = \quad & \max_{t_S} \quad T \sum_{S \subseteq J} \lambda R(S) t_S \\ (CDLP) \quad & \text{s.t.} \quad \sum_{S \subseteq J} \lambda \vec{Q}(S) t_S \leq \frac{1}{T} \vec{c}_0 \\ & \qquad \sum_{S \subseteq J} t_S \leq 1 \\ & \qquad 0 \leq t_S, \forall S \subseteq J \end{aligned} \tag{2}$$

The formulation has a $2^J - 1$ variables t_S, which represents the time each set is offered. Liu and van Ryzin [14] show that $CDLP$ is an upper bound on the DP given in (1). They also show that the problem can be solved efficiently, using column-generation, for the non-overlapping segments MNL model of customer choice.

2.2 Segment-based Deterministic Concave Program ($SDCP$)

In this section we give a formulation based on segment consideration sets. In general it is a looser formulation than $CDLP$ but we show that it coincides exactly with $CDLP$ for non-overlapping segments, is solvable for small consideration sets for more general choice probability functions, and can be tightened by randomization and valid inequalities bringing it closer to $CDLP$ for non-overlapping segments.

For segment l, define a capacity vector $\vec{y}_l \geq 0$ (even if we cannot identify that segment at the time of purchase). Given $\vec{y}_l$, let $G^*_l(\vec{y}_l, \lambda_l)$ represent the optimal revenue we can obtain offering some convex combination of sets to segment l. $G^*_l(\vec{y}_l, \lambda_l)$ can be obtained by solving the following linear program:

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

which, by performing a change of variables ($\lambda_l \tilde{w}_{S_l} = w_{S_l}$), we can write equivalently as as

$$
\begin{aligned}
G_l^*(\vec{y}_l, \lambda_l) = \quad & \max \quad \sum_{S_l \subseteq C_l} R_l(S_l) w_{S_l} \\
(Rgen) \quad & \text{s.t.} \quad \sum_{S_l \subseteq C_l} w_{S_l} \vec{Q}_l(S_l) \leq \vec{y}_l \\
& \qquad \sum_{S_l \subseteq C_l} w_{S_l} \leq \lambda_l \\
& \qquad w_{S_l} \geq 0, \ \forall S_l \subseteq C_l
\end{aligned} \tag{4}
$$

The columns of the linear program ($Rgen$) correspond to all subsets of the consideration set of a single segment at a time, and if the premise is that consideration sets are not large, one can even enumerate all the possible subsets.

Now, define the following concave programming problem over the capacity vectors:

$$
\begin{aligned}
V^{SDCP} = \quad & \max \quad T \sum_{l=1}^{L} G_l^*(\vec{y}_l, \lambda_l) \\
(SDCP) \quad & \text{s.t.} \quad \sum_{l=1}^{L} \vec{y}_l \leq \frac{1}{T} \vec{c}_0 \\
& \qquad \vec{y}_l \geq \vec{0}
\end{aligned} \tag{5}
$$

($SDCP$) is a compact formulation, 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 ($Rgen$).

For simplicity, in the formulation of $SDCP$ and RCP, we assumed a uniform arrival rate λ_l throughout the time horizon. If the arrival rates change over time, say according to a piece-wise linear function, we would need to have variables that correspond to each of the linear parts.

$SDCP$ can be formulated as a single mathematical program, but we chose a bi-level formulation, decomposing the capacity by segment and using subproblems $Rgen$ for each segment l to define the objective function. Our reasons for this modeling are as follows: (i) The bi-level formulation can accommodate slightly larger problems in memory. As $Rgen$ takes subsets of consideration sets, one can fit larger problems by solving it on the fly for each segment, one at a time (ii) As we shall see in §4, the bi-level formulation brings out the essential reason why MNL with non-overlapping segments is solvable and why generalizations are likely to be difficult—by reducing solvability to the ability to do a separation efficiently (iii) It becomes easier to present a randomized version of $SDCP$ in §2.4 and prove that it gives a tighter bound than $SDCP$.

Notice that the objective value of $(Rgen)$, $G_l^*(\vec{y}_l, \lambda_l)$ is a function of both $\vec{y}_l$ and λ_l. In §2.4 we randomize over λ_l and we need to use the following (which simply follows from that fact that both $\vec{y}_l$ and λ_l are on the right-hand side of the constraints of $G_l^*(\vec{y}_l, \lambda_l)$):

Lemma 1. *$G_l^*(\vec{y}_l, \lambda_l)$ is a concave function of $\vec{y}_l$ and λ_l.*

Lemma 2. *V^{SDCP} is a concave function of λ_l.*

The idea of decomposing the problem as in $SDCP$ is quite classical (§6.4.2 of Bertsekas [3]; Maglaras and Meissner [15] in a related context). We differ from the standard right-hand-side allocation as we reduce the total number of variables in the decomposed problems.

2.3 Relationship between $(SDCP)$ and $(CDLP)$

We show that $V^{SDCP} \geq V^{CDLP}$ in general and $V^{SDCP} = V^{CDLP}$ for the case of non-overlapping segments. $SDCP$ can be considered as a relaxation of $CDLP$ where we allow customization of offer sets by segment.

First formulate $CDLP$ as follows:

$$\begin{aligned} &\max && T \textstyle\sum_l \lambda_l \sum_{S_l \subseteq C_l} R_l(S_l) w_{S_l}^l && (6) \\ &(CDLP') && \textstyle\sum_l \lambda_l \sum_{S_l \subseteq C_l} \vec{Q}^l(S_l) w_{S_l}^l \leq \frac{1}{T}\vec{c}_0 && (7) \\ & && w_{S_l}^l \in \text{Proj}(\mathcal{W}), && (8) \end{aligned}$$

where $\mathcal{W}$ is a polytope representing probability distributions w over all subsets S and $\text{Proj}(\mathcal{W})$ is the projection of $\mathcal{W}$ onto the space of $w_{S_l}^l$'s. That is, $w_{S_l}^l \in \text{Proj}(\mathcal{W})$ if there exists a feasible solution to the following system (recall $B_{S_l S} = 1$ if subset $S_l = S \cap C_l$ and 0 otherwise):

$$\begin{aligned} & && \sum_{S \subseteq J} B_{S_l S} w_S = w_{S_l}^l \quad \forall l,\ \forall S_l \subseteq C_l && (9) \\ &(\mathcal{W}([w^l])) && \sum_{S \subseteq J} w_S = 1 && (10) \\ & && w_S \geq 0, \forall S \subseteq J \end{aligned}$$

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

Proposition 1. *$CDLP' = CDLP$.*

Proof
For a feasible $w_{S_l}^l$ of $(CDLP')$, $w_{S_l}^l \in \text{Proj}(\mathcal{W})$ implies, there exists a w_S satisfying (9). Now notice that

$$\sum_l \lambda_l \sum_{S_l \subseteq C_l} \vec{Q}_l(S_l) \sum_S B_{S_l S} w_S = \sum_{S \subseteq J} \lambda w_S \vec{Q}(S), \tag{11}$$

and therefore these w_S satisfy $(CDLP)$ with the same objective value (the objective value is the same by a calculation identical to that of (11)).

Likewise, equation (11) also shows that if w_S is a feasible solution to $(CDLP)$ we derive a feasible solution $w_{S_l}^l$ for $(CDLP')$ by $w_{S_l}^l = B_{S_l S} w_S$ and this has the same objective value.

$\square$

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

Theorem 1. $V^{SDCP} \geq V^{CDLP}$.

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 C_l. This implies that a feasible solution to $(CDLP')$ satisfies $\sum_{S_l} w^l_{S_l} \leq 1$ as $\sum w_S = 1$. Hence we add these redundant constraints and relax constraints (9) to obtain $SDCP$ as in the formulation (5) using (3).

$\square$

Theorem 2. *For the non-overlapping segments model,* $V^{SDCP} = V^{CDLP}$.

Proof

When we have non-overlapping segment consideration sets, the structure of B simplifies. Arrange the rows of B such that the segments are in order (that is all subsets of segment 1 precede those of 2, etc.). Arrange the columns of B so that the initial columns correspond to the subsets S_l representing the rows and in exactly the same order. When the segment consideration sets do not overlap, the matrix B then looks like $B = [I \,\dot{\cdot}\,.]$. Now if $w^l_{S_l}$ is feasible in $(SDCP)$, we can construct an equivalent feasible solution in $(CDLP')$ by setting $w_{S_l} = w^l_{S_l}$ for all subsets $S_l \subseteq C_l, \forall l$ and $w_S = 0$ otherwise. This is a feasible solution to $(CDLP')$ from the structure of B. This shows $V^{SDCP} \leq V^{CDLP}$ and from Theorem 1 we conclude that $V^{SDCP} = V^{CDLP}$ for non-overlapping segments.

$\square$

2.4 Randomized Concave Program (RCP)

We next tighten $(SDCP)$ by randomization, that we call Randomized Concave Program, RCP. Assume we draw a categorical random variable that takes value l with probability λ_l or no arrival (0) with probability $1-\sum_l \lambda_l$. Let the realization of segment l demand in period t for the k^{th} sample path be represented by the indicator function $\mathbb{1}^k_{[lt]}$ equal to 1 if there is a l segment arrival and 0 otherwise.

For the k^{th} instance, we define the concave program

$$
\begin{aligned}
V^{RCP^k} = \quad & \max \quad \sum_{t=1}^{T}\sum_{l=1}^{L} G^*_l(\vec{y}_{lt}, \mathbb{1}^k_{[lt]}\vec{1}) && (12)\\
(RCP^k) \quad & \text{s.t.} \quad \sum_{t=1}^{T}\sum_{l=1}^{L} \vec{y}_{lt} \leq \vec{c}_0 \\
& \qquad \vec{y}_{lt} \geq \vec{0}
\end{aligned}
$$

Next, we define the value of RCP as the average of the K concave programs:

$$V^{RCP(K)} = \frac{\sum_{k=1}^{K} V^{RCP^k}}{K}$$

As in the RLP method of Talluri and van Ryzin [22], we can take an average of the marginal values of (RCP^k) as the controlling bid-price.

2.5 DP, $SDCP$ and RCP

The dynamic program (1) maximizes the expected value over two sets of random variables: the (categorical) random variable of arrival types Λ_t which can take values 0 (the no-purchase option) and $1, \ldots, L$ representing the L segments; and conditioned on a l segment arrival, and for a given set S, the (categorical) purchase random variable $X_S|\Lambda_t = l$ which take the value $j = 0, \ldots, J$ with probability $P_{lj}(S)$ ($j = 0$ represents the no-purchase option). We represent $X_S|\Lambda_t = l$ as distributions over $J + 1$-dimensional unit vectors $\vec{e}_j$ (vector with 1 in the jth position and 0's elsewhere).

We define $V^{RCP}(\vec{c})$ as the expected value over $\{\Lambda_t\}$ of the function defined as below:

$$\begin{aligned} f(\{\Lambda_t\}, \vec{c}) = \quad \max \quad & \sum_{t=1}^{T} \sum_{l=1}^{L} G_l^*(\vec{y}_{lt}, \mathbb{1}_{[\Lambda_t = l]} \vec{1}) \\ \text{s.t.} \quad & \sum_{t=1}^{T} \sum_{l=1}^{L} \vec{y}_{lt} \leq \vec{c} \\ & \vec{y}_{lt} \geq \vec{0} \end{aligned} \tag{13}$$

So $V^{RCP}(\vec{c}) = E_{\{\Lambda_t\}}[f(\{\Lambda_t\}, \vec{c})]$. As $K \to \infty$, $V^{RCP(K)}(\vec{c}) \to V^{RCP}(\vec{c})$ by the Strong Law of Large Numbers as we are taking independent samples to estimate V^{RCP}. While $V^{RCP(K)}$ is an approximation to V^{RCP} we assume from now on that we take sufficient samples so the difference is negligible, and, heuristically, use V^{RCP} and $V^{RCP(K)}$ interchangeably. We show first the relation between RCP and $SDCP$.

Theorem 3. $V^{RCP} \leq V^{SDCP}$.

Proof
Notice that $f(\cdot)$ is a non-negative concave function, and $E[\mathbb{1}_{[\Lambda_t = l]} \vec{1}] = \lambda_l \vec{1}$. So by Jensen's inequality and Lemma 2, the result follows.

□

Recall V^{DP} is the optimal value of (1) for the initial capacity vector $\vec{c}_0$ at time $t = 0$.

Theorem 4. $V^{DP} \leq V^{RCP}$.

Proof
Note that at time t, $V_{t+1}(\cdot)$ is a constant independent of the period t random variables. Let $\vec{V}_{t+1}$ be a $J + 1$-dimension vector whose jth element is $(r_j + V_{t+1}(\vec{c}_t - A_j))$ ($r_0 = 0$ and $A_0 = \vec{0}$). The dynamic program (1) can be represented as

$$V_t(\vec{c}_t) = \max_{S \subseteq J} E_{\Lambda_t}[E_{X_S|\Lambda_t}[(X_S|\Lambda_t)^\top \vec{V}_{t+1}]] \tag{14}$$

Now maximum of expected value is always less than or equal to the expected value of the maximum, so

$$V_t(\vec{c}_t) \le R_t(\vec{c}_t) = E_{\Lambda_t}[\max_{S\subseteq J}[E_{X_S|\Lambda_t}[(X_S|\Lambda_t)^\top \vec{R}_{t+1}]] \tag{15}$$

where $\vec{R}_{t+1}$ is the recursively defined $J+1$-dimension vector whose jth element is $(r_j + R_{t+1}(\vec{c}_t - A_j))$. Now observe that the mathematical program of RCP, (13) can be written recursively as

$$E_{\{\Lambda_t\}}[f(\{\Lambda_t\}, \vec{c}_t)] = E_{\Lambda_t}[f_t(\Lambda_t, \vec{c}_t)]$$

where

$$f_t(\Lambda_t, \vec{c}_t) = \max_{\{\vec{y}_{lt}\ge 0\}} \{\sum_l G^*_l(\vec{y}_{lt}, \mathbb{1}_{[\Lambda_t=l]}) + E_{\Lambda_{t+1}}[f_{t+1}(\Lambda_{t+1}, [\vec{c}_t - \sum_l \vec{y}_{lt})]]\}$$

For any given realization of $\Lambda_t = l$, consider the optimal solution of the right hand side of (15), $S^*_{\Lambda_t=l}$, and the corresponding capacity that the solution occupies: $\vec{y}_{lt} = \sum_{j\in S^*_{\Lambda_t=l}} P_{lj}(S^*_{\Lambda_t=l})A_j$, that is the expected capacity with respect to the random choices $(X_S|\Lambda_t)$. $G^*_l(\vec{y}_{lt}, \mathbb{1}_{[\Lambda_t=l]}) \ge \sum_{j\in S^*_{\Lambda_t=l}} r_j P_{lj}(S^*_{\Lambda_t=l})$. So assume by induction that

$$E[f_{t+1}(\Lambda_{t+1}, \vec{c}_{t+1})] \ge V_{t+1}(\vec{c}_{t+1}),\ \forall \vec{c}_{t+1}$$

(which is trivially true for the last period T), and, from the concavity of $G^*_l(\cdot,\cdot)$ with respect to $\vec{y}_{lt}$ (from Lemma 1), we obtain $R_t(\vec{c}_t) \le E_{\Lambda_t}[f_t(\Lambda_t, \vec{c}_t)]$ and therefore, $V^{DP}(\vec{c}_0) \le V^{RCP}(\vec{c}_0)$.

□

From the results of this section and §2.3, V^{RCP} then gives a tighter upper bound to the DP than V^{CDLP} for non-overlapping segments, so we can write:

Corollary 1. *For the non-overlapping segments model,* $V^{DP} \le V^{RCP} \le V^{CDLP}$.

2.6 Solution Procedure

The concave programs $SDCP$ and (RCP^k) can be solved by subgradient optimization, but here we show that they in fact can be considered linear programs; this allows us to solve it with a general purpose linear program solver. It is a well known fact, that the duals of $(Rgen)$ are subgradients to $G^*(\cdot)$.

We can write $(SDCP)$ as follows:

$$\begin{aligned} V^{SDCP} = \quad & \max \quad \sum_{l=1}^{L} z_l && (16)\\ (SDCP') \quad & \text{s.t.} \quad \sum_{l=1}^{L} \vec{y}_l \le \vec{c}_0 \\ & \qquad\quad z_l - G^*_l(\vec{y}_l, \lambda_l) \le 0 \quad \forall l && (17)\\ & \qquad\quad \vec{y}_l \ge \vec{0} \end{aligned}$$

We replace the constraints (17) by linear constraints, adding them dynamically. If at the k^{th} iteration, $\vec{y}^k_l$ is the capacity vector assigned to segment l, and z^k_l the value of variable z_l, we solve $(Rgen)$ for this segment and obtain the dual vector corresponding to this $\vec{y}^k_l$, $\begin{bmatrix}\vec{\pi}^k_l & w^k\end{bmatrix}$, and the optimal value $G^*_l(\vec{y}^k_l, \lambda_l)$.

If $z_l^k > G_l^*(\vec{y}_l^k, \lambda_l)$, we have found a violated inequality, and we add the following *subgradient cut*

$$z_l - (\vec{\pi}_l^k)^\top \vec{y}_l \le \lambda_l T w^k$$

This procedures terminates after a finite number of steps as $(Rgen)$ is a piecewise linear concave function of y—indeed as the separation can be done in polynomial time, the algorithm actually runs in polynomial time. As a starting point, we solve $(SDCP)$ using the dual vectors generated for an assignment of capacities $\vec{y}_l = \min(\lambda_l T\vec{1}, \frac{\vec{c}_0}{L})$ with the minimum taken component-wise.

The same procedure can be applied to solve RCP substituting $\sum_{t=1}^{T} \mathbb{1}_{[lt]}^k$ for $\lambda_l T$ in both $(Rgen)$ for segment l as well as in (RCP).

3 Solution procedures for MNL

In this section we show how the $SDCP$ formulation can be strengthened for the case of MNL demand based on a recent compact formulation of $SDCP$ for the MNL model due to Gallego et al. [9].

3.1 Compact formulation for MNL

In a recent paper Gallego et al. [9] gave a compact formulation of $SDCP$ for the case of MNL model of demand. The formulation has at most LJ variables (number of products multiplied by number of segments) and just $I + L + LJ$ constraints (I is the number of resources). This is very appealing indeed, as it means, at least for MNL, we have a very fast procedure for solving $SDCP$. The formulation is given as below (following [9] we label it as sales-based linear program (SBLP) but simplify it for MNL rather than the slightly more general attraction model called GAM that they use):

$$\begin{aligned}
V^{SBLP} = \quad & \max \quad \sum_{l=1}^{L} \sum_{j \in C_l} r_j x_{lj} && (18)\\
(SBLP) \quad & \text{s.t.} \quad \sum_{l=1}^{L} \sum_{j \in C_l} A_j x_{lj} \le \vec{c}_0 \\
& \qquad x_{l0} + \sum_{j \in C_l} x_{lj} = \lambda_l T \quad \forall l \\
& \qquad \frac{x_{lj}}{v_{lj}} - \frac{x_{l0}}{v_{l0}} \le 0 \quad \forall l, \forall j \in C_l \\
& \qquad x_{lj} \ge 0
\end{aligned}$$

The constants v_{lj} is the weight of product j and v_{l0} the weight of the outside option in the MNL formula, $P_{lj}(S_l) = \frac{v_{lj}}{v_{l0} + \sum_{j \in S_l} v_{lj}}$. Gallego et al. [9] show that (18) it is equivalent to $SDCP$ when the segment consideration sets do not overlap. The connection between the two formulations is the interpretation

$$x_{lk} = \lambda_l T \sum_{\{S_l \subseteq C_l | k \in S_l\}} P_{lk} w_{S_l}^l = \lambda_l T \sum_{\{S_l \subseteq C_l | k \in S_l\}} \frac{v_{lk}}{v_{l0} + \sum_{j \in S_l} v_{lj}} w_{S_l}^l \qquad (19)$$

in (7) ([9]; see also Topaloglu [25]). Note that the formulation (18) is specific to the MNL model of choice and does not hold for any other choice model.

While ($SBLP$) is very appealing because of its compact size, it is equivalent to $CDLP$ only for the case of non-overlapping segments. In the next section we investigate methods for tightening the formulation when the consideration sets overlap.

First, call a set of constraints *valid* if adding them to an upper bound on the dynamic program (1) still results in an upper bound. Meissner et al. [17] develop a set of valid inequalities for $SDCP$ called product cuts (PC-cuts). They are of the following form:

$$\sum_{\{S_l \subseteq C_l | S_l \supseteq S_{lm}\}} w_{S_l}^l = \sum_{\{S_m \subseteq C_m | S_m \supseteq S_{lm}\}} w_{S_m}^m, \quad \forall S_{lm} \subseteq C_l \cap C_m. \tag{20}$$

which are added to $SDCP$ directly or through the generating linear program ($Rgen$). Now the limiting factor is that (20) can be applied only when the size of the consideration sets is small as it sums over all subsets of C_l that contain a given subset S_{lm}. This limits its applicability to situations where the considerations sets are at most of size 20 or so.

It would be very appealing indeed if one can tighten the formulation $SBLP$ by adding valid inequalities with at most LJ variables. Gallego et al. [9] mention that $SBLP$ *can* be tightened but do not give any hint about the nature of such cuts, and the problem is open. In this section we give valid inequalities in an expanded space that, while not as small as LJ, is of the order of the number of subsets in the *intersections* of consideration sets. So we remove the limitation on the size of consideration sets of (20) and replace it with the less restrictive limitation on the size of intersections of consideration sets. On the other hand, the cuts are limited to the MNL model. From our numerical studies (§5) the cuts appear to have the same power as (20) which almost always obtain the $CDLP$ value ([17]).

3.2 Valid inequalities

We restrict ourselves to the MNL model of choice, so $P_{lj} = \frac{v_{lj}}{v_{l0} + \sum_{j \in C_l} v_{lj}}$. Let $v_{S_l}^l = \sum_{j \in S_l} v_{lj}$. The algebra is significantly reduced if we first make a change of variables as follows:

$$\bar{w}_{S_l}^l = \frac{w_{S_l}^l}{v_{l0} + v_{S_l}^l} \tag{21}$$

So the variables x_{lk} in (19) become

$$\frac{x_{lk}}{v_{lk}} = \lambda_l T \sum_{\{S_l \subseteq C_l | S_l \ni k\}} \bar{w}_{S_l}^l \tag{22}$$

The cuts (20) then become, $\forall S_{lm} \subseteq C_l \cap C_m$

$$\sum_{\{S_l \subseteq C_l | S_l \supseteq S_{lm}\}} v_{l0} \bar{w}_{S_l}^l + \sum_{\{S_l \subseteq C_l | S_l \supseteq S_{lm}\}} v_{S_l}^l \bar{w}_{S_l}^l = \sum_{\{S_m \subseteq C_m | S_m \supseteq S_{lm}\}} v_{m0} \bar{w}_{S_m}^m + \sum_{\{S_m \subseteq C_m | S_m \supseteq S_{lm}\}} v_{S_m}^m \bar{w}_{S_m}^m \tag{23}$$

These are valid inequalities for $SDCP$ as shown in [17]. We will just reduce the number of variables by replacing appropriate summations by new variables as done in (22)—so validity of the resulting inequalities follows from [17] and a simple feasibility check.

For every $S_{lm} \subseteq C_l \cap C_m$ and each product $k \in C_l \setminus C_m$ (i.e., $k \in C_l, k \notin C_m$) define the variable

$$x_{S_{lm},k}^{lm} = \sum_{\{S_l \subseteq C_l | S_l \ni k, S_l \cap (C_l \cap C_m) = S_{lm}\}} \bar{w}_{S_l}^l \tag{24}$$

and for every $S_{lm} \subseteq C_l \cap C_m$, let

$$x^{lm}_{S_{lm}} = \sum_{\{S_l \subseteq C_l | S_l \cap (C_l \cap C_m) = S_{lm}\}} \bar{w}^l_{S_l} \tag{25}$$

Notice that the total number of new variables we are defining is proportional to the number of subsets in the intersections of the consideration sets. Observe now that, (we are adding $0 = (v^l_{S_{lm}} - v^l_{S_{lm}})\bar{w}^l_{S_{lm}}$ to the right hand side)

$$\sum_{k \in C_l \setminus C_m} v_{lk} x^{lm}_{S_{lm},k} = \sum_{\{S_l \subseteq C_l | S_l \cap (C_l \cap C_m) = S_{lm}\}} (v^l_{S_l} - v^l_{S_{lm}})\bar{w}^l_{S_l} \tag{26}$$

obtaining

$$\sum_{\{S_l \subseteq C_l | S_l \cap (C_l \cap C_m) = S_{lm}\}} v^l_{S_l} \bar{w}^l_{S_l} = \sum_{k \in C_l \setminus C_m} v_{lk} x^{lm}_{S_{lm},k} + v^l_{S_{lm}} x^{lm}_{S_{lm}} \tag{27}$$

So, the PC-cuts (23) of [17], written in terms of the new variables are, $\forall S_{lm} \subseteq C_l \cap C_m$

$$\sum_{\{S_l \subseteq (C_l \cap C_m) | S_l \supseteq S_{lm}\}} \left\{ \sum_{k \in C_l \setminus C_m} v_{lk} x^{lm}_{S_l,k} + (v_{l0} + v^l_{S_l}) x^{lm}_{S_l} \right\} =$$
$$\sum_{\{S_m \subseteq (C_l \cap C_m) | S_m \supseteq S_{lm}\}} \left\{ \sum_{k \in C_m \setminus C_l} v_{mk} x^{ml}_{S_m,k} + (v_{m0} + v^m_{S_m}) x^{ml}_{S_m} \right\} \tag{28}$$

The relationship with the variables x_{lk} is given by

$$\frac{x_{lk}}{\lambda_l T v_{lk}} = \sum_{\{S_{lm} \subseteq (C_l \cap C_m) | S_{lm} \ni k\}} x^{lm}_{S_{lm}}, \quad \forall k \in C_l \cap C_m \tag{29}$$

because of the fact that $\forall k \in C_l \cap C_m$ all $S_l \ni k$ intersect with $C_l \cap C_m$.

Finally we have the relationship between $x^{lm}_{S_{lm},k}$ and $x^{lm}_{S_{lm}}$: $x^{lm}_{S_{lm},k} \leq x^{lm}_{S_{lm}}$.

Putting it all together, the tightened formulation for MNL when segment consideration sets overlap can be written as (l, m index the segments):

$$
\begin{aligned}
V^{SBLP+} = \quad & \max \sum_{l=1}^{L} \sum_{j \in C_l} r_j x_{lj} && (30) \\
\text{s.t.} \quad & \\
(SBLP+) \quad & \sum_{l=1}^{L} \sum_{j \in C_l} A_j x_{lj} \le \vec{c}_0 \\
& x_{l0} + \sum_{j \in C_l} x_{lj} = \lambda_l T \quad \forall l \\
& \frac{x_{lj}}{v_{lj}} - \frac{x_{l0}}{v_{l0}} \le 0 \quad \forall l, \forall j \in C_l \\
& \frac{x_{lk}}{\lambda_l T v_{lk}} = \sum_{\{S_{lm} \subseteq (C_l \cap C_m) | S_{lm} \ni k\}} x^{lm}_{S_{lm}}, \quad \forall k \in C_l \cap C_m \\
& x^{lm}_{S_{lm},k} \le x^{lm}_{S_{lm}}, \quad \forall S_{lm} \subseteq C_l \cap C_m, k \in C_l \setminus C_m \\
& \sum_{\{S_l \subseteq (C_l \cap C_m) | S_l \supseteq S_{lm}\}} \left\{ \sum_{k \in C_l \setminus C_m} v_{lk} x^{lm}_{S_l,k} + (v_{l0} + v^l_{S_l}) x^{lm}_{S_l} \right\} = \\
& \quad \sum_{\{S_m \subseteq (C_l \cap C_m) | S_m \supseteq S_{lm}\}} \left\{ \sum_{k \in C_m \setminus C_l} v_{mk} x^{ml}_{S_m,k} + (v_{m0} + v^m_{S_m}) x^{ml}_{S_m} \right\}, \ \forall S_{lm} \subseteq C_l \cap C_m \\
& x_{lj}, x^{lm}_{S_{lm},k}, x^{lm}_{S_{lm}} \ge 0
\end{aligned}
$$

Proposition 2. *Inequalities (28) are valid for* $(SBLP)$*; or in other words,* $V^{DP} \le V^{SBLP+}$ *for the MNL model of choice.*

Proof
We show $V^{CDLP} \le V^{SBLP+}$. Consider $SDCP$ with equations (20). From Meissner et al. [17] this is a relaxation of $CDLP$, and therefore has an objective value $\ge V^{CDLP}$. Consider therefore a solution that satisfies the $SDCP$ constraints as well as equations (20). Based on this solution, define the variables $\bar{w}^l_{S_l}, x_{lk}, x^{lm}_{S_{lm},k}, x^{lm}_{S_{lm}}$ as in (21), (22), (24), (25) respectively. Feasibility of x_{lk} in $(SBLP+)$ in the first three constraint classes of (30) follows from the proof in Gallego et al. [9] of the equivalence of $SDCP$ and $SBLP$ for MNL; that $x^{lm}_{S_{lm},k}, x^{lm}_{S_{lm}}$ satisfy the last three constraint classes of (30) follows from the derivation of (26), (27), (28), (29).

□

3.3 Complexity

The valid inequalities (28) and (29) were defined in an expanded space, and the resulting formulation $(SBLP+)$ can no longer be considered compact. However, we argue that the size of the problem is still reasonable for most applications in assortment optimization and network revenue management. Define

$$\kappa = \max_{\{l,m | l \neq m\}} |C_l \cap C_m|.$$

Then the number of new variables is at most $L^2(J+1)2^\kappa$. Likewise the number of PC-cuts are at most $L^2 2^\kappa$. So the deciding factor for solvability is κ; the Barrier method for linear programming in commercial solvers such as Gurobi or CPLEX is parallelized and extremely powerful, and can easily solve models with κ up to 20 on any modern workstation. We believe this covers most industrial applications. Note that solving $CDLP$ is NP-hard in general, so one cannot expect a polynomial-time solution. Section §5.2.1 shows computational times for a mid-size network and $SBLP+$ runs under a second.

4 *Rgen* oracle for different choice models

The mathematical programs $SDCP$ and RCP are compact (non-differentiable) concave programs, and one can use any standard algorithm to solve them. The complexity however rests on the function evaluation done by ($Rgen$).

If the number of elements in a consideration set is not very large, then one can generate all the subsets by brute force. For instance with 10 elements, one needs to generate only $2^{10} - 1$, or 1027, columns. It is rather unlikely that a customer evaluates more than 10 or 15 alternatives so this is quite plausible. The great advantage of generating all subsets is that the solution methodology can be applied to any choice model whatsoever, expanding the models well beyond tractable-but-restrictive ones such as multinomial-logit.

Notice that generating all subsets is not feasible in $CDLP$ in general—when segments overlap, we need to generate subsets over the full ground set J rather than subsets of segment consideration sets C_l. We believe this and the ability to deal with general choice models is the most attractive aspect of $SDCP$ and RCP.

If for some reason we cannot generate all subsets of the consideration sets, say because the consideration sets are large, then we need to rely on column generation, and we have to assure ourselves that this generation can be done efficiently for at least some choice probability systems. This is in general difficult (NP-hard) as shown in Bront et al. [5] and Rusmevichientong et al. [20].

In the following we intend to throw some light on the column generation procedures and argue that perhaps one cannot really generate the columns efficiently for any but the multinomial-logit model of customer choice.

4.1 Column generation or separation

Let $\vec{\pi} \geq 0$ and w be the dual variables for ($Rgen$). Polynomial-time solvability of ($Rgen$) comes down to the solvability of the separation problem of the dual (Grötschel, Lovász, and Schrijver [10]): Given a $\vec{\pi} \geq 0$ and w, is there a set $S_l \subseteq C_l$ such that

$$w + \lambda_l \vec{Q}_l(S_l)^\top \vec{\pi} < \lambda_l R_l(S_l)$$

or alternately, find $S_l \subseteq C_l$ such that

$$R_l(S_l) - \vec{Q}_l(S_l)^\top \vec{\pi} > \frac{w}{\lambda_l} \tag{31}$$

that we call the separation problem. Letting $w' = \frac{w}{\lambda_l}$ and expanding $\vec{Q}_l(S_l), R_l(S_l)$,

$$\sum_{j \in S_l} [r_j P_{lj}(S_l) - P_{lj}(S_l)(\sum_{i=1}^{I} a_{ij}\pi_i)] > w' \tag{32}$$

Using the fact that $\sum_{j \in S_l} P_{lj}(S_l) + P_{l0}(S_l) = 1$,

$$\sum_{j \in S_l} [r_j P_{lj}(S_l) - P_{lj}(S_l)(\sum_{i=1}^{I} a_{ij}\pi_i)] > w'(\sum_{j \in S_l} P_{lj}(S_l) + P_{l0}(S_l)) \tag{33}$$

which can be rewritten as

$$\sum_{j \in S_l} \frac{P_{lj}(S_l)}{P_{l0}(S_l)}[r_j - w' - \sum_{i=1}^{I} a_{ij}\pi_i] > w' \tag{34}$$

4.2 MNL

The separation problem (34) provides an alternate explanation why $(CDLP)$ for the MNL with disjoint segments model, as well as (RCP) for segment choice probabilities given by MNL (possibly with overlapping segments) can be solved efficiently. The ratio $\frac{P_{lj}(S_l)}{P_{l0}(S_l)}$ is independent of S_l for the MNL model, being the weight of product j divided by the weight of the no-purchase option, and therefore the separation problem (34) is trivial—pick all the products with positive values of $[r_j - w' - \sum_{i=1}^{I} a_{ij}\pi_i]$ and check whether the weighted sum is greater than w'. In short, the greedy algorithm solves the separation problem.

4.3 Supermodular

Can we expand the scope of choice models of consumer behavior, while still maintaining tractability? From the form of (34) it should be clear that we are trying to find a subset that maximizes a weighted cost function. If one looks for set functions that are somewhat tractable, what immediately comes to mind is the class of submodular and supermodular functions (Grötschel et al. [10]). Indeed, this is the only class that we are aware of that can be solved efficiently.

The functions $R_l(S_l)$ and $\vec{Q}_l(S_l)$ in fact have a special form: they are weighted sums of $P_{lj}(S_l)$ with non-negative weights. A set function $\phi : 2^N \to \Re$ is *supermodular* if

$$\phi(S \cup T) + \phi(S \cap T) \geq \phi(S) + \phi(T). \tag{35}$$

and called *submodular* if the inequality is reversed.

The function ϕ is *intersecting* supermodular if the inequality (35) holds whenever $S \cap T \neq \emptyset, S \setminus T \neq \emptyset, T \setminus S \neq \emptyset$. Consider a J-dimensional vector function $\vec{\phi}(S) : 2^J \to \Re^J_+$ that maps subsets of J to a real vector, with the j^{th} component $\vec{\phi}_j(S) = 0$ if $j \notin S$. Call the function *(vector)-supermodular* if it is component-wise supermodular.

Consider the class of choice probability models for which $\frac{P(S)}{P_0(S)}$ is (vector)-supermodular. Clearly the MNL model is one such[1].

We wish to find a set S_l that maximizes the left-hand side of (34). The main difficulty now is if for some of the j's, $r_j - w' - \sum_{i=1}^{I} a_{ij}\pi_i \leq 0$. The problem of *minimizing* supermodular functions is NP-hard again, so we would really like $\frac{P_{lj}(S_l)}{P_{l0}(S_l)}$ to be submodular functions for all such j's with

[1] We were able to uncover only a handful of articles (Fujishige [7] Benati [1], Berman and Krass [2]) that link choice systems and submodularity, despite both concepts being used in an immense variety of applications.

negative coefficients. As the coefficients can be positive or negative, it could well be that only modular set functions (i.e., both super and sub modular, such as the MNL of §4.2) can be separated efficiently.

We believe that this is some evidence that there are very few choice functions for which the separation can be done efficiently.

4.4 Nested MNL and Generalized Extreme Value (GEV) models

Generalized Extreme Value (GEV) models generalize the MNL choice function and the nested logit model, a generalization of MNL that avoids some of the consistency problems of MNL.

In GEV models, the probabilities for an offer set S are given by

$$P_j(S) = \frac{e^{V_j + \ln G_j(S)}}{\sum_{i \in S} e^{V_i + \ln G_i(S)}} \tag{36}$$

where the functions $G_j(S) = \frac{\partial G(S)}{\partial x_j}$ and the function $G(S) = G(\{x_j, j \in S\}), x_j \geq 0$ is a non-negative differentiable function satisfying some additional properties (see Daly and Bierlaire [6]). Consider now the ground set J. When $G(x_1, \dots, x_J) = \sum_{j \in J} x_j^{\mu}, \mu \geq 0$, we get the MNL (that has the form described in §4.2), and when $G(x_1, \dots, x_J) = \sum_{k=1}^{K} \left(\sum_{j \in N_k} x_j^{\mu_1} \right)^{\frac{\mu}{\mu_1}}, \mu, \mu_1 \geq 0$, where $N_k, k = 1, \dots, K$ are mutually exclusive exhaustive subsets of S ("nests" of alternatives) we get the so-called nested MNL model (with a tree of depth two). If the offer set is S, we restrict the nests to be $N_k \cap S$. We investigate tractability of separation of (31) this nested MNL model—the problem appears to be intractable even for this specialization, but suggests one can use standard approximation algorithms for maximization of submodular functions to do the separation approximately.

For a subset $S \subseteq J$, we define $G(S) = \sum_{k=1}^{K} \left(\sum_{j \in N_k \cap S} x_j^{\mu_1} \right)^{\frac{\mu}{\mu_1}}$. For this case, letting $k(j)$ be the index k such that $N_k \ni j$,

$$G_j(S) = \frac{\partial G(S)}{\partial x_j} = \mu x_j^{\mu_1 - 1} \left(\sum_{i \in N_{k(j)} \cap S} x_i^{\mu_1} \right)^{(\frac{\mu}{\mu_1} - 1)}$$

Recall that all the attributes and parameters are fixed and we are only interested in finding a subset S that satisfies (34) for a segment l. We assume $x_i > 0$. The form of (36) then implies that $\frac{P_{lj}(S_l)}{P_{l0}(S_l)}$ in (34) can be expressed as $c_j a_{k(j)}(S)$, where, to simplify the algebra, we use some terms $c_j \geq 0, a_i = x_i^{\mu_1}, a_k(S) = \left(\sum_{i \in N_k \cap S} a_i \right)^{(\frac{\mu}{\mu_1} - 1)}$. By our assumption $x_i > 0$, $a_i > 0, \forall i$. Likewise, to simplify notation, let $b_j = c_j(r_j - w' - \sum_{i=1}^{I} a_{ij} \pi_i)$. Note that b_j can be negative.

The separation problem then is to find a subset S_l that maximizes $\sum_{j \in S_l} a_{k(j)}(S_l) b_j$. One can observe that this breaks up by nest; i.e., for each k, we find the subset

$$S_k^* = \arg \max_{S \subseteq N_k} \sum_{j \in S} a_k(S) b_j \tag{37}$$

and compose $S_l = \cup_{k=1}^{K} S_k^*$.

So we fix a nest k and consider subsets $S \subseteq N_k$ from now on. The objective function in (37) can be rewritten as $a_k(S) \sum_{j \in S} b_j$. Now notice that if $b_j \geq 0$, then j belongs to the optimal set—if

j is excluded, we can add it to the optimal set and it increases the value of both $a_k(S)$ as well as $\sum_{j\in S} b_j$ contradicting optimality. So let $J^+ = \{i|b_i \geq 0\}$ and $J^- = \{i|b_i < 0\}$, $A = \sum_{i\in J^+} a_i$, $B = \sum_{i\in J^+} b_i$, and notice that $A, B \geq 0$. The objective function then can be written as

$$\max_{S\subseteq J^-} \left(A + \sum_{i\in N_k\cap S} a_i\right)^{(\frac{\mu}{\mu_1}-1)} \left(B + \sum_{i\in S} b_i\right) \tag{38}$$

We change the objective function by taking logarithms

$$\max_{S\subseteq J^-} \left(\frac{\mu}{\mu_1} - 1\right) \log\left(A + \sum_{i\in N_k\cap S} a_i\right) + \log\left(B + \sum_{i\in S} b_i\right) \tag{39}$$

defining $\log(x) = -\infty$ whenever $x \leq 0$.

Now notice that functions of the form $\log\left(B + \sum_{i\in S} b_i\right)$ in (39) are intersecting submodular whenever either $b_i > 0, \forall i$ or $b_i < 0, \forall i$: From the definition (35), we need to show

$$\left(B + \sum_{i\in S\cup T} b_i\right)\left(B + \sum_{i\in S\cap T} b_i\right) \leq \left(B + \sum_{i\in S} b_i\right)\left(B + \sum_{i\in T} b_i\right)$$

which after canceling common terms on both sides, becomes

$$\left(\sum_{i\in S\cup T} b_i\right)\left(\sum_{i\in S\cap T} b_i\right) \leq \left(\sum_{i\in S} b_i\right)\left(\sum_{i\in T} b_i\right)$$

which holds whenever $b_i > 0, \forall i$ or $b_i < 0, \forall i$, as $b_i b_{i'} > 0$ for all pairs i, i' in the product expansions, and every such pair in the left-hand side is present in the right-hand side.

If $(\frac{\mu}{\mu_1} - 1) < 0$, we then have a problem of maximizing the difference of two supermodular functions (NP-hard) and if $(\frac{\mu}{\mu_1} - 1) > 0$, maximizing a submodular function (again NP-hard). However, both problems are quite well studied and one can use approximation algorithms and heuristics to approximately separate the inequalities (Narasimhan and Bilmes [18], Nemhauser and Wolsey [19]).

5 Numerical Results

In the following we solve the compact formulations of $SDCP$ and RCP and $SBLP$ and $SBLP+$ assuming uniform arrival rates over all the time periods. We use the examples in the literature and compare against $CDLP$ which we solve exactly generating all the columns by enumeration. We first compare the upper bounds generated by the methods and their run-times and then report results of simulations that test their revenue performance.

5.1 Test Networks

We wish to compare against past computational studies, so we take the exact same networks as used in Liu and van Ryzin [14] and in Bront et al. [5] as we are able to reconstruct the data from the papers. We perform revenue simulations on two benchmark networks, calling them as in their original papers:

1. Parallel Flights/Overlapping (Bront et al. [5]): 2 flights, 6 products, 2 overlapping segments
2. Small Network (overlapping) (Bront et al. [5]): 7 flights, 22 products, 2 overlapping segments

and use another larger example called the Hub-and-Spoke Network (overlapping) (Bront et al. [5]) with 8 flights, 80 products, 40 overlapping segments for testing computational running time.

5.1.1 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.

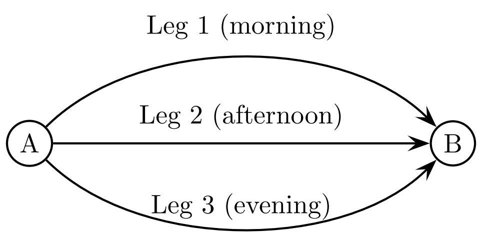

Figure 1: Parallel Flights Example.

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.

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.

5.1.2 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 3 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 4, 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.

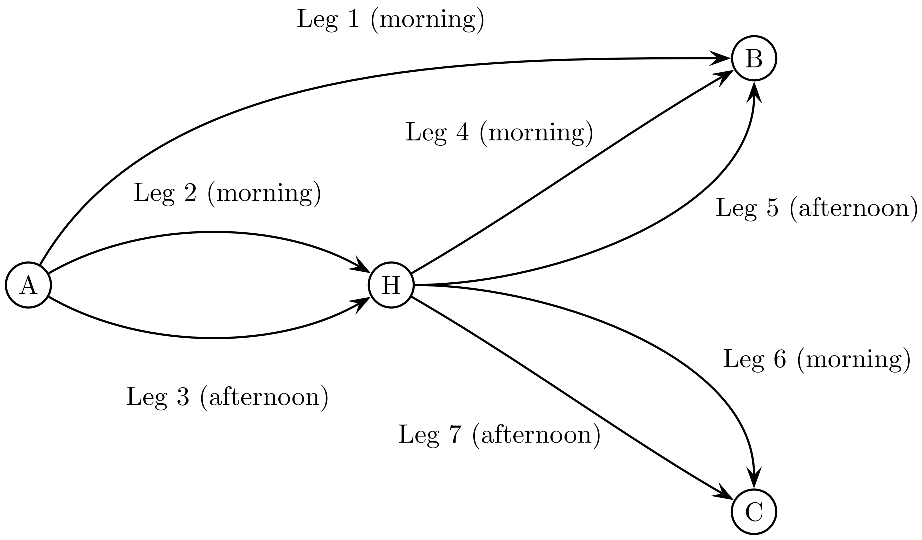

Figure 2: Small Network example.

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 3: 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 4: Segment definitions for Small Network Example

5.2 Value functions

We scale the capacities as in Liu and van Ryzin [14] and Bront et al. [5], multiplying the capacities by a factor α. We also test with different no-purchase weights, using the same choices as in Liu and van Ryzin [14] and Bront et al. [5]. $SDCP$ is quite close to $CDLP$ for this example with overlapping

α	v_0	CDLP	SDCP	RCP	SBLP	SBLP+
0.6	[1,5,5,1]	56,884	58,755	58,313	58,755	56,912
	[1,10,5,1]	56,848	58,755	58,313	58,755	56,884
	[5,20,10,5]	53,819	54,684	54,523	54,684	53,842
0.8	[1,5,5,1]	71,936	73,870	73,720	73,870	72,031
	[1,10,5,1]	71,794	73,870	73,672	73,870	71,936
	[5,20,10,5]	61,868	63,439	63,401	63,439	61,996
1	[1,5,5,1]	79,155	85,424	84,978	85,424	80,078
	[1,10,5,1]	76,866	83,376	83,071	83,376	77,605
	[5,20,10,5]	63,255	65,847	65,794	65,847	63,274
1.2	[1,5,5,1]	80,371	88,331	88,110	88,331	81,003
	[1,10,5,1]	78,045	86,332	86,054	86,332	78,385
	[5,20,10,5]	63,296	66,647	66,647	66,647	63,321

Table 5: Upper bounds for Parallel Flights/overlapping segments example (Bront et al. [5]).

segments at high load factors (low α), but loses out by a large margin at low load-factors. RCP improves over $SDCP$ but perhaps not by as much as one expects (say, compared to the improvement of RLP compared to DLP for the independent demand model). The upper bound given by $SBLP+$ is quite close to $CDLP$ value in all the configurations.

The computational times for all of the above problems were negligible ($SDCP$ for instance runs under one CPU second). We believe $SDCP$ scales to industrial-size problems; moreover, as we mentioned earlier, if the size of the consideration sets are reasonable (10 to 15), can be applied to any choice model whatsoever.

α	v_0	CDLP	SDCP	RCP	SBLP	SBLP+
0.6	[1,5]	215,793	216,672	216,347	216,672	215,793
	[5,10]	200,515	206,457	205,628	206,457	200,515
	[10,20]	170,137	173,959	173,958	173,959	170,137
0.8	[1,5]	266,934	272,670	272,393	272,670	266,934
	[5,10]	223,173	230,500	230,417	230,500	223,173
	[10,20]	188,574	193,629	193,501	193,629	188,574
1	[1,5]	281,967	296,523	296,301	296,523	281,967
	[5,10]	235,284	245,402	245,225	245,402	235,284
	[10,20]	192,038	198,872	198,746	198,872	192,038
1.2	[1,5]	284,772	301,477	301,475	301,477	284,772
	[5,10]	238,562	248,816	248,810	248,816	235,862
	[10,20]	192,373	198,994	198,994	198,994	192,373

Table 6: Upper bounds for Small Network example (Bront et al. [5]). *SBLP+* achieves the *CDLP* value in all cases.

5.2.1 Computational Time

We report computational times on the Hub-and-Spoke Network (overlapping) of Bront et al. [5]) (8 flights, 80 products, 40 overlapping segments). We use CPLEX 12.2 and the machine has a Core i7 980 processor. The CPU time reported for *RCP* includes the time for the generation of the sample paths (300). As the consideration sets for each segment are relatively small (up to 4 in each set), we generate all possible subsets of the consideration set. The problem is too large (80 products) for solving *CDLP* by subset generation so we do not report its running times. The running times reported for CDLP in Liu and van Ryzin [14] and Bront et al. [5] are on entirely different machines with a different version of CPLEX and using column-generation techniques so they are rather hard to recreate or compare. But to get an idea, the computational times reported for this network in [5], using column generation, is as high as 3000 seconds.

α	v_0	SDCP	RCP	SBLP	SBLP+
0.6	[1,5]	0.5760	39.7800	0.0160	0.0620
	[5,10]	0.6870	52.165	0.0010	0.0620
	[10,20]	0.6879	59.7280	0.0140	0.0620
0.8	[1,5]	0.5462	29.4800	0.0010	0.0010
	[5,10]	0.6400	36.0200	0.0010	0.0010
	[10,20]	0.7219	41.2500	0.0030	0.0010
1	[1,5]	0.1870	51.4700	0.0150	0.0149
	[5,10]	0.2650	68.8580	0.0010	0.0010
	[10,20]	0.2300	66.2500	0.0030	0.0160
1.2	[1,5]	0.4220	90.4800	0.0010	0.0010
	[5,10]	0.5150	101.4600	0.0010	0.0010
	[10,20]	0.4680	94.2400	0.0020	0.0010

Table 7: CPU time (in Seconds) for the different methods on a large hub-and-spoke network with capacity of 180 for all legs of the network.

α	v_0	SDCP	RCP	SBLP	SBLP+
0.6	[1,5]	167,569	167,458	167,569	158,040
	[5,10]	136,498	136,455	136,498	126,720
	[10,20]	116,307	116,294	116,307	106,399
0.8	[1,5]	188,551	187,938	188,551	171,121
	[5,10]	154,324	153,910	154,324	132,976
	[10,20]	131,270	130,550	131,270	117,327
1	[1,5]	206,432	206,409	206,432	181,559
	[5,10]	170,500	170,109	170,500	149,955
	[10,20]	136,203	136,153	136,203	122,064
1.2	[1,5]	223,637	223,040	223,637	190,548
	[5,10]	178,213	177,908	178,213	154,624
	[10,20]	136,203	136,222	136,203	122,401

Table 8: The value functions of the different approximations for this large example. $SBLP+$ gives a 5 to 10% tighter bound than the other methods with a negligible running time (Table 7).

5.3 Revenue simulations

In this section we perform revenue simulations to test the revenue performance of the various methods.

5.3.1 Description of the simulations and policy

Our simulation procedure generates arrival streams with each arrival stream representing the booking requests for one instance of demand for the network. For the parallel flights examples we generate 2000 streams and for the small network 250 streams (we reduce the number of instances due to $CDLP$ solution times—we solve $CDLP$ by generating all the subsets).

The simulations in [5], [14], [16] and [17] use the dual solution of $CDLP$ and a decomposition procedure to obtain a control policy. In contrast we follow a simple randomization procedure: We interpret the variables w_S as giving the parameter of a Bernoulli random variable for offering set S with probability w_S. For the segment-level formulations we randomize over the offer sets for each segment $w^l_{S_l}$ and compose the offer set as the union of the segment-level offer sets. For RCP we averaged the values across all the randomized solutions and then took the union.

For the formulations $SBLP$ and $SBLP+$ we follow a similar policy calculating the probability of offering product k for segment l as follows. Following our interpretation of the variables x_{lk} as

$$x_{lk} = \lambda_l T \sum_{\{S_l \subseteq C_l | k \in S_l\}} \frac{v_{lk}}{v_{l0} + \sum_{j \in S_l} v_{lj}} w^l_{S_l}$$

we independently draw a Bernoulli random variable with probability p_{lk} for including k in the offer set, where

$$p_{lk} = \frac{x_{lk} v_{l0}}{x_{l0} v_{lk}}.$$

The offer set is then composed of all the products that are drawn (that is union of the offer sets for the segments as for the other methods). While distinct from the bid-price/decomposition approaches, we find that this policy gives good revenues for all the methods (except perhaps RCP). For instance, the revenue results we report for the methods are comparable to the results in Table A1 reported in the electronic companion of [5].

We also report the standard deviations of the observed revenue to give an idea of the level of confidence.

5.3.2 Simulation results for the Parallel-Flights example

We report in Table 9 the average revenues obtained in our simulations experiments at various capacity scalings and parameter choices for the Parallel-Flights example of §5.1.1. Somewhat surprisingly

α	v_0	CDLP	SDCP	RCP	SBLP	SBLP+
0.6	[1,5,5,1]	54,435	56,220	55,891	56,068	55,540
	[1,10,5,1]	54,502	56,162	55,673	56,000	55,460
	[5,20,10,5]	50,737	52,355	51,806	52,169	52,037
0.8	[1,5,5,1]	68,993	70,120	69,520	69,899	69,654
	[1,10,5,1]	68,624	69,707	68,804	69,470	69,230
	[5,20,10,5]	59,720	59,719	58,873	59,593	59,997
1	[1,5,5,1]	76,883	76,973	76,125	76,884	76,829
	[1,10,5,1]	75,173	74,669	73,801	74,694	75,195
	[5,20,10,5]	62,366	60,861	60,512	60,843	62,185
1.2	[1,5,5,1]	79,588	77,772	77,163	77,841	79,390
	[1,10,5,1]	77,309	75,413	74,478	75,452	77,519
	[5,20,10,5]	62,677	61,543	61,325	61,573	62,700

Table 9: Average revenue results for the overlapping segment Parallel Flights example [5] with 2000 demand sample paths.

$SDCP$, RCP and $SBLP$ give much better revenue results than $CDLP$ when capacity is highly constrained, but all three do badly at higher capacities. As one would expect, $SDCP$ and $SBLP$ show similar characteristics, as they coincide for MNL. $SBLP+$ is the most robust, beating $CDLP$ at low capacities and equaling $CDLP$ at the higher capacities. Table 10 gives the percentage comparison with $CDLP$ and Table 11 the standard deviations of the revenues to determine confidence levels.

5.3.3 Simulation results for the Small-Network example

Table 12 gives the average revenues obtained in our simulations experiments at various capacity scalings and parameter choices for the Small-Network example of §5.1.2. Here, the performance of all the methods is nearly identical to that of $CDLP$ at the low capacity points, but at higher capacity only $SBLP+$ holds its own against $CDLP$, while all the other methods show markedly poor revenue with the first configuration. So, once again $SBLP+$ is the most robust, with good performance at all capacity scalings. Table 13 shows the percentage comparison with respect to $CDLP$. The standard deviations of the revenues are given in Table 14.

α	v_0	CDLP	SDCP	RCP	SBLP	SBLP+
0.6	[1,5,5,1]	0.00	3.28	2.68	3.00	2.03
	[1,10,5,1]	0.00	3.04	2.15	2.75	1.76
	[5,20,10,5]	0.00	3.19	2.11	2.82	2.56
0.8	[1,5,5,1]	0.00	1.63	0.76	1.31	0.96
	[1,10,5,1]	0.00	1.58	0.26	1.23	0.88
	[5,20,10,5]	0.00	0.00	-1.42	-0.21	0.46
1	[1,5,5,1]	0.00	0.12	-0.99	0.00	-0.07
	[1,10,5,1]	0.00	-0.67	-1.83	-0.64	0.03
	[5,20,10,5]	0.00	-2.41	-2.97	-2.44	-0.29
1.2	[1,5,5,1]	0.00	-2.28	-3.05	-2.19	-0.25
	[1,10,5,1]	0.00	-2.45	-3.66	-2.40	0.27
	[5,20,10,5]	0.00	-1.81	-2.16	-1.76	0.04

Table 10: Percentage average revenue improvement over $CDLP$ for the overlapping segment Parallel Flights example.

α	v_0	CDLP	SDCP	RCP	SBLP	SBLP+
0.6	[1,5,5,1]	1,990	1,422	1,500	1,583	1,831
	[1,10,5,1]	2,003	1,459	1,651	1,563	1,569
	[5,20,10,5]	3,657	2,253	2,362	2,369	2,268
0.8	[1,5,5,1]	3,076	3,141	3,310	3,267	3,054
	[1,10,5,1]	3,331	3,422	3,712	3,454	3,277
	[5,20,10,5]	4,615	4,613	4,532	4,689	4,529
1	[1,5,5,1]	5,295	5,861	5,846	5,815	5,217
	[1,10,5,1]	5,698	5,906	6,085	5,871	5,650
	[5,20,10,5]	6,019	5,888	5,840	5,825	6,176
1.2	[1,5,5,1]	6,934	6,950	6,916	6,941	6,841
	[1,10,5,1]	6,981	6,954	7,005	6,939	6,932
	[5,20,10,5]	6,301	6,153	6,051	6,169	6,406

Table 11: Standard deviations of revenue simulations with 2000 sample paths for the Parallel Flights Example.

α	v_0	CDLP	SDCP	RCP	SBLP	SBLP+
0.6	[1,5]	212,816	212,979	211,447	212,058	212,730
	[5,10]	195,559	196,226	194,330	196,607	195,496
	[10,20]	167,553	167,744	166,038	167,885	167,269
0.8	[1,5]	262,579	260,059	258,855	259,868	261,233
	[5,10]	220,665	219,116	216,640	218,925	220,773
	[10,20]	186,807	187,408	185,429	186,758	186,784
1	[1,5]	280,882	272,733	272,192	273,357	278,520
	[5,10]	233,607	233,234	231,847	233,668	234,156
	[10,20]	192,286	192,201	190,105	192,216	191,131
1.2	[1,5]	285,251	277,366	276,580	277,004	283,220
	[5,10]	238,858	239,049	236,394	238,578	239,665
	[10,20]	193,298	193,244	190,992	193,177	192,809

Table 12: Average revenue results for the Small Network example [5] with 250 demand sample paths.

α	v_0	CDLP	SDCP	RCP	SBLP	SBLP+
0.6	[1,5]	0.00	0.08	-0.64	-0.36	-0.04
	[5,10]	0.00	0.34	-0.63	0.54	-0.03
	[10,20]	0.00	0.11	-0.90	0.20	-0.17
0.8	[1,5]	0.00	-0.96	-1.42	-1.03	-0.51
	[5,10]	0.00	-0.70	-1.82	-0.79	0.05
	[10,20]	0.00	0.32	-0.74	-0.03	-0.01
1	[1,5]	0.00	-2.90	-3.09	-2.68	-0.84
	[5,10]	0.00	-0.16	-0.75	0.03	0.24
	[10,20]	0.00	-0.04	-1.13	-0.04	-0.60
1.2	[1,5]	0.00	-2.76	-3.04	-2.89	-0.71
	[5,10]	0.00	0.08	-1.03	-0.12	0.34
	[10,20]	0.00	-0.03	-1.19	-0.06	-0.25

Table 13: Percentage average revenue improvement over $CDLP$ for the Small Network example [5].

α	v_0	CDLP	SDCP	RCP	SBLP	SBLP+
0.6	[1,5]	4,129	3,284	3,845	3,645	4,094
	[5,10]	5,261	5,594	6,140	5,889	5,560
	[10,20]	5,151	5,719	5,707	5,491	5,058
0.8	[1,5]	6,612	7,500	7,251	7,714	7,335
	[5,10]	7,454	6,621	6,552	6,909	6,080
	[10,20]	6,494	6,595	7,022	6,786	6,913
1	[1,5]	8,995	9,542	9,511	9,448	9,715
	[5,10]	9,219	8,044	7,816	8,321	7,911
	[10,20]	9,117	8,498	8,931	8,672	8,623
1.2	[1,5]	10,360	10,483	9,859	10,658	10,896
	[5,10]	9,524	9,781	9,259	10,238	10,102
	[10,20]	8,738	8,733	8,315	9,296	8,790

Table 14: Standard deviations of revenue simulations with 250 sample paths for the Small Network example.

6 Conclusions and further research

We gave a new segment-based deterministic concave-program ($SDCP$) upper bound to the choice network RM dynamic program, that coincides with the $CDLP$ upper-bound of Gallego et al. [8] and Liu and van Ryzin [14] for non-overlapping segments. We then showed how this can be tightened in the randomized concave programming (RCP) method, similar to the RLP for the independent-class model, and by adding valid inequalities. Our cuts are a specialization of the ones developed in [17] to the compact formulation of [9] for the MNL choice model. The advantage of these cuts is that the space of the resulting program is exponential only in the number of products in the intersection of two segments' consideration sets, rather than the size of the consideration sets as in [17].

If the number of elements in a consideration set for a segment is not very large, both ($SDCP$) and (RCP) can be applied to any choice model whatsoever, expanding the set of models well beyond the multinomial-logit. We have given some evidence to show that the assortment optimization appears to be difficult for almost all choice models except the MNL, so this approach defining segments to have small consideration sets (and justified by applications and empirical research as in Talluri [21], Hauser and Wernerfelt [11]) is a tractable way to approach the problem for general discrete-choice models.

References

[1] Benati, S. 1997. Submodularity in competitive location problems. *Ricerca Operativa* **26** 3–34.

[2] Berman, O., D. Krass. 1998. Flow intercepting spatial interaction model: a new approach to optimal location of competitive facilities. *Location Science* **6** 41–65.

[3] Bertsekas, D. P. 1999. *Nonlinear Programming.* 2nd ed. Athena Scientific, Belmont, MA.

[4] 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.

[5] 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.

[6] Daly, A., M. Bierlaire. 2006. A general and operational representation of generalised extreme value models. *Transportation Research Part B* **40** 285–305.

[7] Fujishige, S. 1978. Polymatroidal dependence structure of a set of random variables. *Information and Control* **39**(1) 55–72.

[8] 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.

[9] 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/`.

[10] Grötschel, M., L. Lovász, A. Schrijver. 1988. *Geometric Algorithms and Combinatorial Optimization*, vol. 2. Springer.

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

[12] 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.

[13] Kunnumkal, S., H. Topaloglu. 2008. A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. Tech. rep., School of IEOR, Cornell University, Ithaca, NY.

[14] 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.

[15] Maglaras, C., J. Meissner. 2006. Dynamic pricing strategies for multi-product revenue management problems. *Manufacturing and Service Operations Management* **8**(2) 136–148.

[16] Meissner, J., A. K. Strauss. 2008. Network revenue management with inventory-sensitive bid prices and customer choice. Tech. rep., Lancaster University, Department of Management Science.

[17] Meissner, J., A. K. Strauss, K. T. Talluri. 2011. An enhanced concave programming method for choice network revenue management. Tech. rep., Working Paper 1259, Department of Economics, Universitat Pompeu Fabra, available at `http://www.econ.upf.edu/docs/papers/downloads/1259.pdf`.

[18] Narasimhan, M, J. A. Bilmes. 2005. A submodular-supermodular procedure with applications to discriminative structure learning. *UAI'05*. 404–412.

[19] Nemhauser, G. L., L. A. Wolsey. 1978. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research* **3** 177–188.

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

[21] 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.

[22] Talluri, K. T., G. J. van Ryzin. 1999. A randomized linear programming method for computing network bid prices. *Transportation Science* **33** 207–216.

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

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

[25] Topaloglu, H. 2010. Joint stocking and product offer decisions under the multinomial logit model. Tech. rep., School of IEOR, Cornell University, Ithaca, NY, available at `http://people.orie.cornell.edu/~huseyin/publications/retail_assort.pdf`.

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