A Tractable Consideration Set Structure for Network Revenue Management

Arne K. Strauss
Kalyan Talluri

This version: October 2012
(February 2012)

Barcelona GSE Working Paper Series

Working Paper nº 606

A tractable consideration set structure for network revenue management

Arne K. Strauss[*], Kalyan Talluri[†]

26 October 2012

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. When there are products that are being considered for purchase by more than one customer segment, $CDLP$ is difficult to solve since column generation is known to be NP-hard. However, recent research indicates that a formulation based on segments with cuts imposing consistency ($SDCP+$) is tractable and approximates the $CDLP$ value very closely. In this paper we investigate the structure of the consideration sets that make the two formulations exactly equal. We show that if the segment consideration sets follow a tree structure, $CDLP = SDCP+$. We give a counterexample to show that cycles can induce a gap between the $CDLP$ and the $SDCP+$ relaxation. We derive two classes of valid inequalities called flow and synchronization inequalities to further improve ($SDCP+$), based on cycles in the consideration set structure. We give a numeric study showing the performance of these cycle-based cuts.

Keywords: discrete-choice models, network revenue management, consideration sets

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's 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

[*]Warwick Business School, University of Warwick, Coventry CV4 7AL, United Kingdom. email:arne.strauss@wbs.ac.uk
[†]ICREA & Universitat Pompeu Fabra, Ramon Trias Fargas 25-27, 08005 Barcelona, Spain. email:kalyan.talluri@upf.edu

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 hence 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 [16] 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 [15], Gallego, Iyengar, Phillips, and Dubey [5], Liu and van Ryzin [9], Kunnumkal and Topaloglu [8], Zhang and Adelman [18], Meissner and Strauss [10], Bodea, Ferguson, and Garrow [2], Bront, Méndez-Díaz, and Vulcano [3], Méndez-Díaz, Bront, Vulcano, and Zabala [12], Kunnumkal [7]).

Gallego et al. [5] and Liu and van Ryzin [9] propose a linear program called Choice Deterministic Linear Program ($CDLP$), while Talluri [17] developed a formulation called Segment-based Deterministic Concave Program ($SDCP$) that is weaker than the upper bound resulting from the $CDLP$ but coincides for non-overlapping segment consideration sets. Gallego, Ratcliff, and Shebalov [6] taking a similar approach, but specialized to the multinomial logit (MNL) model of choice, propose a compact linear program called the Sales Based Linear Program ($SBLP$) that also coincides with $CDLP$ for non-overlapping segment consideration sets.

The size of the $CDLP$ formulation grows exponentially in the number of products, and the solution strategy is to use column generation; but finding an entering column is NP-hard even in restrictive cases such as the MNL model of choice with just two segments and overlapping consideration sets (Bront et al. [3], Rusmevichientong, Shmoys, and Topaloglu [13]).

$SDCP$ is tractable when consideration sets are small, but its performance is poor when segment consideration sets overlap (i.e., the bound is significantly looser than $CDLP$). Meissner, Strauss, and Talluri [11] extend the $SDCP$ formulation, that we call $SDCP$+ here, to obtain progressively tighter relaxations of $CDLP$ for the case of overlapping consideration sets by adding *product cuts* that interpret the linear programming decision variables as randomization rules. These constraints are easy to generate and work for general discrete-choice models. Talluri [17] specializes the $SBLP$ formulation of Gallego et al. [6] to obtain a more compact formulation (that we call $SBLP$+) that is exponential only in the size of the intersections of the consideration sets. In numerical testing $SDCP$+ and $SBLP$+ achieve the $CDLP$ value in many instances despite being an order of magnitude more compact.

This paper is an attempt to understand when the optimal value of the relatively compact $SDCP$+ relaxation (and $SBLP$+ for the MNL model) achieves the intractable $CDLP$ formulation value. We show that if the segment consideration sets follow a certain tree structure, the two problems are equivalent, and give a counterexample to show that cycles can induce a gap between $CDLP$ and the $SDCP$+ relaxation.

We then derive valid inequalities for the dynamic program based on cycles in the consideration set structure. We give a numeric study showing the performance of these cycle-based cuts.

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 two approximations of the dynamic program, namely the *CDLP* and the *SDCP+*, followed by the main structural results that we prove in this paper. In §4 we illustrate the approximations on a numerical example from the literature. In §5 we show two applications from sports RM and buy-up modeling where the segment consideration set structure is a tree as in our structural result. In §6 we describe two classes of cuts based on cycles in the consideration set graph, and test their performance in §7. In §8 we summarize our conclusions.

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 the sale of a seat for an itinerary consumes one seat each on a set of flight legs (the resources); likewise, 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. The resources used by j are represented by $a_{ij} = 1$ if product j uses resource i, and $a_{ij} = 0$ otherwise, or alternately with the 0-1 incidence vector $\vec{A}_j$ of product j. Let A denote the resource-product incidence matrix; columns of A are then $\vec{A}_j$. We denote the vector of capacities at time t 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).

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 $\sum_{l \in \mathcal{L}} p_l = 1$, so $\lambda = \sum_{l \in \mathcal{L}} \lambda_l$. 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 (see [14] for a survey on consideration set literature). 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 the availability of products $j \in J \setminus C_l$. 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 as in the MNL model. Whenever we specify probabilities for a segment l for a given offer set S, we just write it with respect to $S_l|S := C_l \cap S$ (note that $P_j^l(S) = P_j^l(S_l|S)$). We define the vector $\vec{P}^l(S) = [P_1^l(S_l|S), \dots, P_n^l(S_l|S)]$ (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 firm sells $j \in S$ with probability $P_j(S) = \sum_{l \in \mathcal{L}} p_l P_j^l(S_l|S)$ and makes no sale with probability $P_0(S) = 1 - \sum_{j \in S} P_j(S)$. We define the vector $\vec{P}(S) = [P_1(S), \dots, P_n(S)]$. Notice that $\vec{P}(S) = \sum_{l \in \mathcal{L}} p_l \vec{P}^l(S)$. 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 $R^l(S) = \sum_{j \in S_l} r_j P_j^l(S_l|S)$ and $R(S) = \sum_{j \in S} r_j P_j(S)$.

Let $V_t(\vec{c}_t)$ denote the maximum expected revenue that can be earned over the remaining time horizon $[t, T]$, given remaining capacity $\vec{c}_t$ in period t. Then $V_t(\vec{c}_t)$ must satisfy the Bellman equation

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

with the boundary condition $V_T(\vec{c}_T) = 0$ for all $\vec{c}_T$. The set $J(\vec{c}_t)$ contains all products that can be feasibly offered given remaining available capacity: $J(\vec{c}_t) := \{j \in J | \vec{A}_j \leq \vec{c}_t\}$. 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 $\vec{c}_0$.

In our notation and demand model we broadly follow Bront et al. [3] and Liu and van Ryzin [9].

3 Approximations

We give the two formulations that we consider in this paper. Both can be seen as relaxations of (1).

3.1 Choice Deterministic Linear Program ($CDLP$)

The choice-based deterministic linear program ($CDLP$) defined in Gallego et al. [5] and Liu and van Ryzin [9] 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 fraction of time set S is offered (including $w_{\{\emptyset\}}$). Liu and van Ryzin [9] 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. [3] and Rusmevichientong et al. [13] 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 Enhanced Segment-Based Deterministic Concave Program ($SDCP+$)

Talluri [17] proposed an upper bound on $CDLP$ called the segment-based deterministic concave program ($SDCP$) that coincides with the $CDLP$ when the segments do not overlap. The $SDCP$ corresponds to the formulation ($SDCP+$) given below without the equations (3). In applications, 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. Meissner et al. [11] develop a set of valid inequalities for $SDCP$ called product cuts as defined in (3) that tighten $SDCP$, and we call the combined formulation ($SDCP+$):

$$
(SDCP+) \quad
\begin{aligned}
\max \quad & \sum_{l=1}^{L} \sum_{S_l \subseteq C_l} \lambda_l R^l(S_l) w^l_{S_l} \\
\text{s.t.} \quad & \\
& \sum_{l=1}^{L} \vec{y_l} \le \vec{c}_0 \\
& \sum_{S_l \subseteq C_l} \lambda_l \vec{Q}^l(S_l) w^l_{S_l} \le \vec{y_l}, \quad \forall\, l \in \mathcal{L} \\
& \sum_{S_l \subseteq C_l} w^l_{S_l} = T, \quad \forall\, l \in \mathcal{L} \\
& \vec{y_l} \le \lambda_l T \vec{1}, \quad \forall\, l \in \mathcal{L} \\
& \sum_{\{S_l \subseteq C_l | S_l \supseteq S_{lm}\}} w^l_{S_l} - \sum_{\{S_m \subseteq C_m | S_m \supseteq S_{lm}\}} w^m_{S_m} = 0, \\
& \qquad \forall S_{lm} \subseteq C_l \cap C_m, \forall \{l, m\} \subset \mathcal{L} : C_l \cap C_m \ne \emptyset \qquad (3) \\
& w^l_{S_l} \ge 0, \qquad \forall S_l \subseteq C_l, \forall\, l \in \mathcal{L}, \\
& \vec{y_l} \ge \vec{0}, \qquad \forall\, l \in \mathcal{L}.
\end{aligned}
$$

The product cuts are *valid* in the sense that adding them to $(SDCP)$ still results in an upper bound on the stochastic dynamic program (1). The number of these cuts depends on the size of the overlap of consideration sets, and can be enumerated when overlaps are small. Even otherwise, one can add the cuts selectively, say restricting to the case where $|S_{lm}|$ is 2 or 3. The number of variables depends on the size of the consideration sets themselves, and can likewise be enumerated if consideration sets are small.

We state the following proposition whose proof is trivial:

Proposition 1. *If two segments have identical consideration sets, we can merge the two segments into one, changing $R^l(S_l)$ and $Q^l(S_l)$ as the expected revenue and resource consumption for the two segments combined, without altering the solution of $(SDCP+)$.*

Therefore, whenever two segments have identical consideration sets, we merge them and consider a merged segment.

3.3 Example showing gap between $CDLP$ and $SDCP+$

Figure 1 shows an example with five products and three segments and their respective consideration sets. For this example we show that there is a gap between the optimal objective values of $SDCP+$ and $CDLP$. Assume $T = 1$, capacity $c = 1$, revenue $r_j = 1$ for all products $j \in J = \{1, 2, 3, 4, 5\}$, $\lambda_l = 1/3$ for all segments $l \in \{A, B, C\}$, and the purchase probabilities defined as follows: $P^A_j(\{1, 2\}) := 0.5$ for $j = 1, 2$, $P^A_j(\{2, 5\}) := 0.5$ for $j = 2, 5$, $P^B_2(\{2\}) := 1$, $P^B_j(\{1, 2, 3\}) := 1/3$ for $j = 1, 2, 3$, $P^C_4(\{4\}) := 1$ and

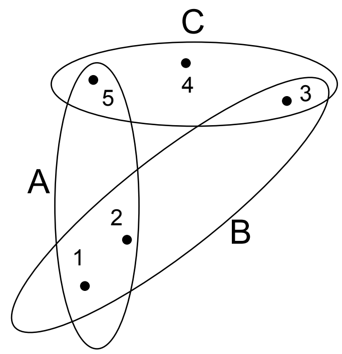

Figure 1: Consideration sets for 3 segments, 5 products

$P_j^C(\{3,5\}) := 0.5$ for $j = 3, 5$, and 0 for all other sets.

Claim: There is a feasible solution to $SDCP+$ for this network with objective value 1.

Proof

For this data, an optimal solution to $SDCP+$ is given by $y_l = 1/3$ for all segments l, and $w^A_{\{1,2\}} = w^A_{\{2,5\}} = 0.5$, $w^B_{\{2\}} = w^B_{\{1,2,3\}} = 0.5$, $w^C_{\{4\}} = w^C_{\{3,5\}} = 0.5$, and $w^l_{S_l} = 0$ otherwise for all $l \in \{A, B, C\}$, $S_l \subset C_l$. This solution is feasible to $(SDCP+)$ since $\lambda_l \sum_{S_l \subseteq C_l} Q^l(S_l) w^l_{S_l} = 1/3 = y_l$ for all segments l, the product cuts (3) for all pairs of segments $\{l, m\}$ and sets $S_{lm} \subset C_l \cap C_m$, $S_{lm} \neq \emptyset$, are satisfied as reported in Table 1, and they also hold for $S_{lm} = \emptyset$ since the solution satisfies $\sum_{S_l \subseteq C_l} w^l_{S_l} = 1$ for all segments l. The other constraints are likewise satisfied as can easily be checked. The objective value is 1 since $\lambda_l \sum_{S_l \subseteq C_l} R^l(S_l) w^l_{S_l} = 1/3$ for each $l \in \{A, B, C\}$.

□

$\{l,m\}$	S_{lm}	$S_l \supseteq S_{lm}$	$\sum_{S_l \supseteq S_{lm}} w^l_{S_l}$	$S_m \supseteq S_{lm}$	$\sum_{S_m \supseteq S_{lm}} w^m_{S_m}$
{A,B}	{1}	{1},{1,2},{1,5},{1,2,5}	0.5	{1},{1,2},{1,3},{1,2,3}	0.5
{A,B}	{2}	{2}, {1,2}, {2,5}, {1,2,5}	1	{2}, {1,2}, {2,3}, {1,2,3}	1
{A,B}	{1,2}	{1,2},{1,2,5}	0.5	{1,2},{1,2,3}	0.5
{A,C}	{5}	{5},{1,5},{2,5},{1,2,5}	0.5	{5},{3,5},{4,5},{3,4,5}	0.5
{B,C}	{3}	{3},{1,3},{2,3},{1,2,3}	0.5	{3},{3,4},{3,5},{3,4,5}	0.5

Table 1: Evaluation of all product cuts (3) for the example in §3.3.

Claim: There is no corresponding solution to $CDLP$ with the same objective value.

Proof

$CDLP$ has $2^5 = 32$ variables corresponding to subsets $S \subset J = \{1, 2, 3, 4, 5\}$. Under the single-leg example described above, we can enumerate all 32 subsets S and calculate the corresponding objective coefficient $\lambda R(S)$. We find that $\lambda R(S) \leq 2/3$ for all $S \subset J$, with equality reached for the sets $\{2, 5\}, \{1, 2, 3\}, \{1, 2, 4\}, \{2, 3, 5\}, \{1, 2, 3, 4\}$ and $\{1, 2, 3, 5\}$. It follows that there can be no feasible solution to $CDLP$ that has objective value greater than 2/3 since the objective is a convex combination of these coefficients (note that

$T = 1$). This proves the claim.

□

This simple example illustrates that there can be a strict gap between $CDLP$ and $SDCP+$ even if all product cuts are satisfied. The underlying reason for that gap is that there is no global offer set that can be projected onto the segment consideration sets so as to coincide with the segment-level optimal solution. These clashes cannot be removed by the product cuts because there is a cycle in the dependency structure of the consideration sets. We elaborate on this observation in the next section.

3.4 A tree structure for consideration sets

We seek to obtain a structural result on when $CDLP$ and $SDCP+$ are equivalent. To that end, we define a bipartite *intersection graph of the consideration sets* as follows: There are two types of nodes, one type called segment node, the other is called intersection node. Each node of the former type corresponds to a segment, each of the latter represents a set of the form $C_k \cap C_l$ for some segment pair (k, l). If a set S is the intersection of two distinct pairs $S = C_m \cap C_n = C_k \cap C_l$, then S is represented by a single node. Edges from segment k node connect to all the sets of the form $C_k \cap C_l \neq \emptyset$.

The example of §3.3 has a cycle as can be seen from Figure 2. This turns out to be the critical feature: If the segment consideration sets do not have a cycle, arranged say in the form of a tree (or, in general, a forest), then the product cuts are sufficient to ensure equivalence between $CDLP$ and $SDCP+$.

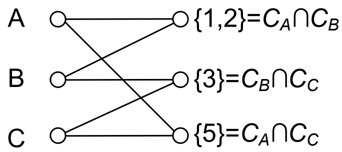

Figure 2: Intersection graph for the example in §3.3.

3.5 Equivalence for tree overlapping consideration sets

We show in this section that if the intersection graph is a forest then $SDCP+ = CDLP$. To provide some intuition behind this result, note that the intersection tree tells us which segments are directly or indirectly connected to each other, in the sense that a solution w^l for some segment l has implications on the solution w^k for all segment nodes k that are reachable from segment node l. The product cuts synchronize the segment-level sub-solutions with respect to their weights, so that for a tree-structured intersection graph it is possible for a given w^l to arrange the segment-level solutions w^k for all segments k reachable from l, in a way such that they represent the projections of a feasible global solution onto the respective segments. However, if there is a cycle then the offer set implications along that cycle can lead to a contradiction since the product cuts do not synchronize segment-level solutions with respect to which sets can be offered in parallel so as to guarantee the existence of a global offer set solution.

This was illustrated in Example 3.3, where the sets $S_1^A = \{1,2\}$, $S_1^B = \{1,2,3\}$, and $S_1^C = \{3,5\}$ are offered in parallel, but S_1^C would require the product 5 to be offered on segment A as well, which creates a contradiction and thus indicates that these segment-level solutions do not have a corresponding *CDLP* solution.

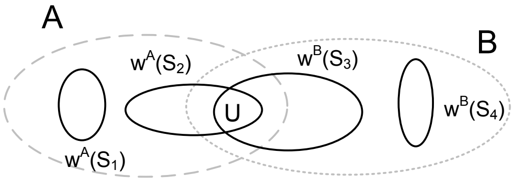

Figure 3: Merging procedure used in the proof of Proposition 2.

Proposition 2. *If the intersection graph is a forest, then $CDLP = SDCP+$.*

Proof

We use a merging procedure in the proof that for clarity we explain with a simple example shown in Figure 3 with two segments A, B and their consideration sets. Figure 3 shows a solution to $SDCP+$ and we wish to construct a solution to $CDLP$ from this solution. Note that the product cuts imply the restriction $w^A_{S_2} = w^B_{S_3}$, corresponding to the set U in the intersection of the consideration set. Moreover, $w^A_{S_1} + w^A_{S_2} = T$ and $w^B_{S_3} + w^B_{S_4} = T$. This implies that $w^A_{S_1} = w^B_{S_4}$. So we construct weights for the $CDLP$ formulation as $w^{\text{CDLP}}_{S_1 \cup S_4} = w^A_{S_1} = w^B_{S_4}$ and $w^{\text{CDLP}}_{S_2 \cup S_3} = w^A_{S_2} = w^A_{S_3}$. This $CDLP$ solution satisfies $w^{\text{CDLP}}_{S_1 \cup S_4} + w^{\text{CDLP}}_{S_2 \cup S_3} = T$, as well as the capacity constraints, and has the same objective value as the $SDCP+$ solution. In the following proof of Proposition 2, we essentially repeat this argument for the much more complicated case with L segments and arbitrary consideration sets using an induction argument (made possible by the tree structure).

Any solution to $CDLP$ is a solution to $SDCP+$ as shown in [11], hence $CDLP \leq SDCP+$. It remains to show $CDLP \geq SDCP+$. We construct a feasible solution to $CDLP$ from a feasible solution to $SDCP+$ by induction on the number of segments L.

Consider the case of a single segment $L = 1$, and a given feasible solution $(w^L_{S_L}, y_L)_{S_L}$ to $SDCP+$. Then $w_S^{CDLP} := w^L_S$ for all $S \subset J$ is a feasible solution to $CDLP$ with the same objective value.

Next, we consider $L > 1$. Without loss of generality, the discussion will focus on an intersection graph that is a finite tree rather than a forest since the same arguments can be made for each tree that makes up the forest. Assuming that it is a tree, there must be at least two leaves, i.e. nodes with degree 1. By definition, intersection nodes have at least degree 2, so there exists a segment node that is a leaf. Without loss of generality, let this node correspond to the consideration set of segment L, and let $\overline{SDCP+}$ represent the problem $SDCP+$ with the segment L removed. Consider a feasible solution (w, y) to $SDCP+$, where (w, y) is shorthand notation for $w^l_{S_l}$ for $S_l \subseteq C_l$, for all segments $l \in \mathcal{L} := \{1, 2, \ldots, L\}$, and y_{il} for all resources i and $l \in \mathcal{L}$.

This solution induces a feasible solution $(\bar{w}, \bar{y})$ to $\overline{SDCP+}$ by defining $\bar{w}^l_{S_l} := w^l_{S_l}$ for all $S_l \subseteq C_l$, for all $l \in \bar{\mathcal{L}} := \mathcal{L} \setminus \{L\}$, and $\bar{y}_l := y_l$ for all $l \in \bar{\mathcal{L}}$. The solution $(\bar{w}, \bar{y})$ produces an objective value equal to that of

$SDCP+$ less $\sum_{S_L \subseteq C_L} \lambda_L R^L(S_L) w^L_{S_L}$. By the induction assumption, there exists a feasible solution $\bar{w}^{CDLP}_S$ for all $S \subseteq \bar{J} := \cup_{l=1}^{L-1} C_l$ to $CDLP$ with the same objective value, and $\bar{w}^{CDLP}$ induces $(\bar{w}, \bar{y})$ meaning that $\bar{w}^l_{S_l} = \sum_{S \subseteq \bar{J} | S \cap C_l = S_l} \bar{w}^{CDLP}_S$ for all $l \in \bar{\mathcal{L}}$, $S_l \subseteq C_l$ for $l \in \bar{\mathcal{L}}$, and $\bar{y}_l = \sum_{S_l \subseteq C_l} \lambda_l \vec{Q}^l(S_l) \bar{w}^l_{S_l}$ for all $l \in \bar{\mathcal{L}}$.

Now we construct a feasible solution w^{CDLP}_S for all $S \subset J$ to $CDLP$ that induces (w, y) for $SDCP+$ with same objective value. Since L is a leaf of the intersection tree, all interactions with other segments are via a set S^{int} that is associated with the intersection node to which L is connected. Let us denote all segments that are connected to this intersection node by $\mathcal{L}^{\text{int}}$.

Consider a set $U \subseteq S^{\text{int}}$ that is *maximal* for segment L with respect to S^{int}, that is there is no set $S_L \subseteq C_L$ such that $U \subsetneq S_L \cap S^{\text{int}}$ and positive support $w^L_{S_L} > 0$. Note that for a feasible solution to $SDCP+$, the product cuts ensure that if a set is maximal for L with respect to S^{int}, it is maximal for all segments $l \in \mathcal{L}^{\text{int}}$ with respect to S^{int}. Moreover, from the definition of maximal

$$\sum_{\{S_l \subseteq C_l | S_l \cap S^{\text{int}} \supseteq U\}} w^l_{S_l} = \sum_{\{S_l \subseteq C_l | S_l \cap S^{\text{int}} = U\}} w^l_{S_l}, \forall l \in \mathcal{L}^{\text{int}}.$$

We select an arbitrary *maximal* set $U \subseteq S^{\text{int}}$ and segment $l \in \mathcal{L}^{\text{int}}$. The following argument shows that the total weight $\tau(U)$ that we offer sets that intersect with S^{int} exactly in U is the same in solutions w^L and $\bar{w}^{CDLP}$:

$$\begin{aligned}
\tau(U) &= \sum_{S_L \subseteq C_L | S_L \cap S^{\text{int}} = U} w^L_{S_L} && (4)\\
&= \sum_{S_l \subseteq C_l | S_l \cap S^{\text{int}} = U} w^l_{S_l} && (5)\\
&= \sum_{S_l \subseteq C_l | S_l \cap S^{\text{int}} = U} \bar{w}^l_{S_l} && (6)\\
&= \sum_{S_l \subseteq C_l | S_l \cap S^{\text{int}} = U} \sum_{S \subseteq \bar{J} | S \cap C_l = S_l} \bar{w}^{CDLP}_S && (7)\\
&= \sum_{S \subseteq \bar{J} | S \cap S^{\text{int}} = U} \bar{w}^{CDLP}_S.
\end{aligned}$$

The first equality holds by definition, the second due to maximality and the product cuts being satisfied by the solution w to $SDCP+$, the third since $w^l_{S_l} = \bar{w}^l_{S_l}$, the fourth because $\bar{w}^{CDLP}$ induces $\bar{w}$, and the final one as a result of a reformulation.

As a consequence, we can merge the solution $w^L_{S_l}$ with $\bar{w}^{CDLP}$ over total weight $\tau(U)$ to obtain w^{CDLP} for all sets that intersect with S^{int} only in the fixed set U. We illustrate the process in Figure 4 by drawing two parallel bars of equal length representing the weight $\tau(U)$, each bar with intervals corresponding to the support of the solutions w^L and $\bar{w}^{CDLP}$ (the order of the sets does not matter). Merging the sets as depicted ensures that the constructed solution w^{CDLP} induces w^L as well as w^l for $l \in \bar{\mathcal{L}}$ (the latter due to the induction assumption on $\bar{w}^{CDLP}$).

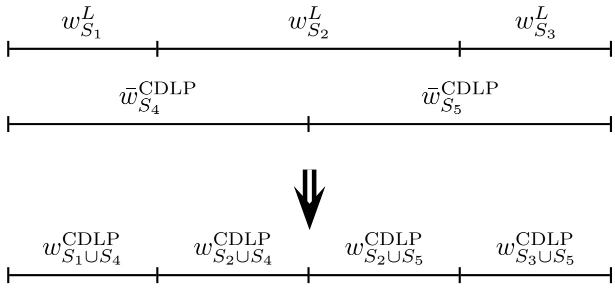

Figure 4: Illustration of the procedure to merge solutions in the support of w^L and $\bar{w}^{CDLP}$ to obtain w^{CDLP} for a fixed set $U \subseteq S^{\text{int}}$.

Now remove all the solution components $w^L_{S_L}$ and $w^l_{S_l}$ with positive support for all $l \in \mathcal{L}^{\text{int}}$ with $S_l \cap S^{\text{int}} = U$ and $S_L \cap S^{\text{int}} = U$. After the removal, the product cut equations for the remaining solution remain valid because of the equalities (4–5). We repeat this merging process by taking a maximal $U \subseteq S^{\text{int}}$ at each stage till we conclude with $U = \emptyset$. At every stage, as U is a maximal set, all the sets that contained U, namely sets of the form $U \subsetneq S_l \cap S^{\text{int}}, l \in \mathcal{L}^{\text{int}}$ were maximal sets in previous stages and therefore accounted for by equalities (4–5) for the set S_l; now combining it with the product cuts for the set U, we again obtain equalities (4–5).

The solution w^{CDLP} that emerges from this process is feasible to $CDLP$: it holds that $\sum_{U \subseteq S^{\text{int}}} \tau(U) = T$ (note that $U = \emptyset \subset S^{\text{int}}$), and therefore, by construction, $\sum_S w^{CDLP}_S = T$. That the capacity constraint of $CDLP$ is satisfied follows from the induction assumption that $\bar{w}^{CDLP}$ induces $(\bar{w}, \bar{y})$, with $\bar{w} := w$ and $\bar{y} := y$, combined with the fact that (w, y) is feasible to $SDCP+$ and that we constructed w^{CDLP} in a way such that we only added capacity consumption equal to that of segment L under solution w^L. So the combined solution also satisfies the induction for L segments.

The objective value of $CDLP$ equals that of $SDCP+$ because in the merging process we only add products of $C_L \setminus S^{\text{int}}$ to the solution $\bar{w}^{CDLP}$, and since these products do not influence other segments as they are only in the consideration set of segment L, we only add the contribution of segment L to the objective without a change of the contribution of other segments.

□

4 A numerical example in the literature

In this section we examine a test network used first in Liu and van Ryzin [9] and Bront et al. [3], and later in Talluri [17], Meissner et al. [11] and Meissner and Strauss [10]. For brevity, we just give the bare details of the network, specifically the consideration sets and the segments. We also give the value of a formulation called $SBLP+$ that was derived by Talluri [17]. $SBLP+$ applies the product-cuts to a compact formulation called sales-based linear program ($SBLP$) due to Gallego et al. [6], and it is interesting to observe that this formulation does not give $CDLP$ value even with tree intersection structures, and we explain why at the end of this section.

The example consists of three parallel flight legs with 4 segments and 6 products, and the consideration sets and preference weights under the MNL choice model are given in Table 2. The intersection graph of

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.

the parallel flights example is shown in Figure 5 and it can be seen that it has a cycle. However, note that segments 3 and 4 have identical consideration sets, and by Proposition 1 we merge them and obtain a tree intersection graph. This explains why the $SDCP+$ values are identical to that of $CDLP$ in Table 3.

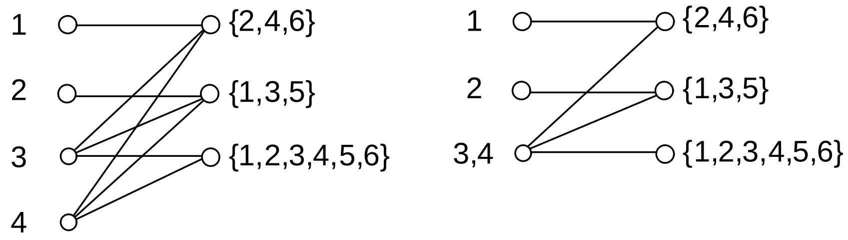

Figure 5: Intersection graph for the parallel-flights example before (left) and after (right) merging segments 3 and 4.

We now explain the reason for the gap between the objective values of the $SDCP+$ and the $SBLP+$ formulations. Since $SDCP+$ applies to any general discrete-choice model, we are able to use Proposition 1 and consider a merged segment with a resulting tree intersection structure. One cannot do that for the $SBLP+$ formulation as it is specific to the MNL model and merging the segments would give a mixture model, so the cycle in the intersection graph persists for the $SBLP+$ formulation. It follows that $SDCP+$ is in general not equivalent to $SBLP+$ and the more complex formulation of $SDCP+$ can result in improved bounds under the MNL choice model.

5 Applications With Tree-Structures Intersection Graphs

In this section we describe two applications where customer choice behavior is modeled by a tree consideration-set structure.

α	v_0	$CDLP$	$SDCP+$	$SDCP$	$SBLP+$
0.6	[1,5,5,1]	56,884	56,884	58,755	56,912
	[1,10,5,1]	56,848	56,848	58,755	56,884
	[5,20,10,5]	53,820	53,820	54,684	53,842
0.8	[1,5,5,1]	71,936	71,936	73,870	72,031
	[1,10,5,1]	71,795	71,795	73,870	71,936
	[5,20,10,5]	61,868	61,868	63,439	61,996
1	[1,5,5,1]	79,156	79,156	85,424	80,078
	[1,10,5,1]	76,866	76,866	83,376	77,605
	[5,20,10,5]	63,256	63,256	65,847	63,274
1.2	[1,5,5,1]	80,371	80,371	88,331	81,003
	[1,10,5,1]	78,045	78,045	86,332	78,385
	[5,20,10,5]	63,296	63,296	66,647	63,321

Table 3: Upper bounds for Parallel Flights/overlapping segments example (Bront et al. [3]). Capacities are scaled by multiplying the capacities by a factor α. Different no-purchase weights are given in v_0, using the same choices as in Liu and van Ryzin [9] and Bront et al. [3]. $SDCP+$ was obtained using product cuts corresponding to subsets $|S_{lm}| \leq 2$

.

5.1 Sports RM

In Balseiro, Gocmen, Gallego, and Phillips [1], the authors consider an application of revenue management to sport events ticket options. The products consist of an advance ticket A and a product O^l for each one of the segments l. A customer in segment l has a consideration set $\{A, O^l\}$. Thus we obtain a star consideration structure (Figure 6), and the $SDCP+$ formulation is sufficient to obtain the $CDLP$ value.

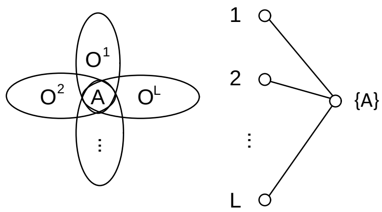

Figure 6: A star consideration structure (left) and the corresponding intersection graph (right) in sports RM.

5.2 Buy-up/buy-down chain models

In buy-up models, the products are arranged linearly based on price and some nested restrictions, with the less restrictive product being priced higher. A customer is modeled as considering a contiguous range of products and purchase the products with decreasing probabilities; the early paper [4] for instance models customer behavior this way. If the consideration sets form a chain as in Figure 7, then the consideration set graph can be represented by a tree.

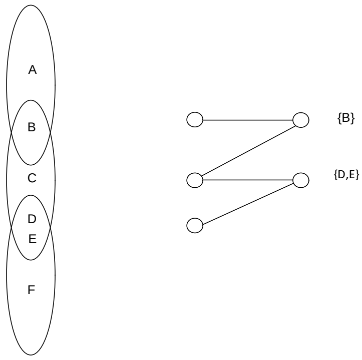

Figure 7: Contiguous-range consideration sets (left) and the corresponding intersection graph (right) in a linearly arranged set of products.

6 Cycle Cuts

The cycles that cause the gap between $CDLP$ and $SDCP+$ lead us to a set of inequalities that can further strengthen $SDCP+$. In this section we describe two classes of cuts that are based on cycles. The first, called cycle-flow inequalities, works in the $SDCP$ space of variables while the second, called synchronization cuts, adds new variables. Neither class is polynomial-time separable, but rely on identifying cycles in the intersection graph and the sets that form this cycle; on the other hand they provide a systematic way of tightening the formulation and can be useful to determine if we can improve the solution further, as calculating the $CDLP$ value is prohibitive except for small examples. Moreover, there may be applications where the consideration structure has few cycles, in which case it might be possible to enumerate the cycles in a reasonable amount of time. Both the classes of cuts are applicable to any discrete-choice model of demand.

6.1 Cycle-flow Inequalities

We first explain these inequalities with respect to a small example shown in Figure 8. A cut is valid if weights induced by a set $S \subset J$ for each of the segment-level problems satisfy the cut, as the $CDLP$ solution is a convex combination of such sets. To start off with the simplest possible scenario, suppose a set S induces R_1, R_2, R_3, R_4 in the above picture (i.e., $S \cap C_1 = R_1$, and so on), and has a weight w_S in the $CDLP$ solution. Then the projection of the weight w_S in $SDCP$ onto each of the subsets (alternately, the solution induced by w_S in $SDCP$ by $w^i_{S \cap C_i} = w_S$) should satisfy $w^1_{R_1} = w^2_{R_2} = w^3_{R_3} = w^4_{R_4}$.

The product-cut equalities (3) are really flow-balance equalities, that say $w^1_{R_1} = w^2_{R_2}$, $w^2_{R_2} = w^3_{R_3}$ etc., but do not encompass a cycle. To capture the cycle flow, we pick a segment in the cycle (say segment 2 in

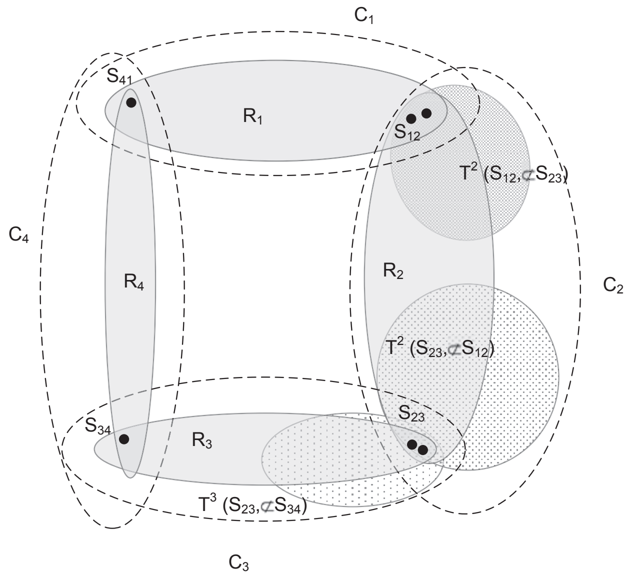

Figure 8: A cycle that induces S_{12} in $C_1 \cap C_2$, S_{23} in $C_2 \cap C_3$, S_{34} in $C_3 \cap C_4$, S_{41} in $C_4 \cap C_1$; set $T^2(S_{12}, \not\subset S_{23}) \subseteq C_2$ where $T^2(S_{12}, \not\subset S_{23}) \cap C_1 \supseteq S_{12}$ and $T^2(S_{12}, \not\subset S_{23}) \cap C_3 \not\supseteq S_{23}$; set $T^2(S_{23}, \not\subset S_{12}) \subseteq C_2$ such that $T^2(S_{23}, \not\subset S_{12}) \cap C_3 \supseteq S_{23}$ and $T^2(S_{23}, \not\subset S_{12}) \cap C_1 \not\supseteq S_{12}$; set $R_2 \subseteq C_2$ such that $R_2 \cap (C_1 \cap C_2) \supseteq S_{12}$ and $R_2 \cap (C_2 \cap C_3) \supseteq S_{23}$, etc.

our running example) and derive some conditions that the flow in the cycle should satisfy.

There are two complications that come up in capturing the flow on the cycle. First, we have to sum over all the sets of the form R_i that *contain* sets of the form $S_{i,i+1} \subset C_i \cap C_{i+1}$. For this purpose, define $\mathcal{R}_i(S_{i-1,i}, S_{i,i+1}) = \{R \subseteq C_i | R \cap C_{i-1} \supseteq S_{i-1,i}, R \cap C_{i+1} \supseteq S_{i,i+1}\}$ (if we index the segments in the cycle $1, \ldots, k$ (so k is the length of the cycle), it should be understood that 0 corresponds to k and $k+1$ corresponds to 1).

The second, and the more complicated part, comes up when $S \cap C_i$ leads to sets of the form $T^2(S_{12}, \not\subset S_{23})$ where $T^2(S_{12}, \not\subset S_{23}) \cap C_1 \supseteq S_{12}$ and $T^2(S_{12}, \not\subset S_{23}) \cap C_3 \not\supseteq S_{23}$ in Figure 8. A set S can presumably induce R_1, R_3, R_4 in segments $1, 3, 4$ respectively and sets $T^2(S_{12}, \not\subset S_{23})$ and $T^2(S_{23}, \not\subset S_{12})$ in segment 2. So the "flow" can go out of the cycle through sets such as these.

We account for this next. Define

$$\mathcal{T}^i(S_{i-1,i}, \not\subset S_{i,i+1}) = \{T \subseteq C_i | T \cap C_{i-1} \supseteq S_{i-1,i}, T \cap C_{i+1} \not\supseteq S_{i,i+1}\}.$$

We define similarly $\mathcal{T}^i(S_{i+1,i}, \not\subset S_{i-1,i})$.

For a given set of subsets in the intersections of the consideration sets of the cycle $\{S_{i,i+1} \neq \emptyset\}$, we draw a flow-graph with a node for each $S_{i,i+1}$ and a set of arcs from $S_{i-1,i}$ to $S_{i,i+1}$ representing the sum of weights of the type R_i. In addition at each node there is a certain flow entering $S_{i,i+1}$ made up of weights of sets of the form $T^{i+1}(S_{i,i+1}, \not\subset S_{i+1,i+2})$ and a flow leaving from sets of the form $T^i(S_{i,i+1}, \not\subset S_{i-1,i})$. Figure 9 represents the flow graph corresponding to Figure 8. The product-cut equalities (3) say that flow in should

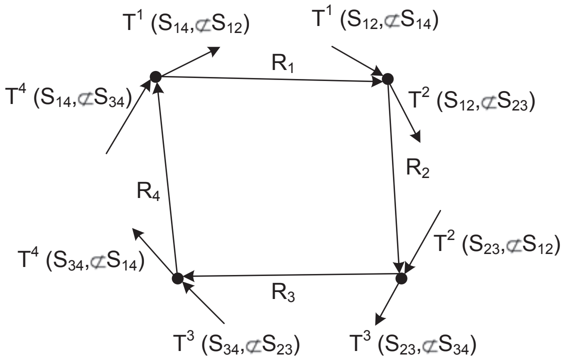

Figure 9: The flow graph corresponding to the example in Figure 8

equal to the flow out at each node in this graph.

Now our *cycle-flow* inequalities are the following for each segment i on the cycle and for a fixed set of subsets $\{S_{i,i+1}\}$, going in the clockwise direction:

$$\sum_{j=i+1}^{i+k-1} \sum_{T \in \mathcal{T}^j(S_{j-1,j}, \not\subset S_{j,j+1})} w_T^j \geq \sum_{T \in \mathcal{T}^i(S_{i,i+1}, \not\subset S_{i-1,i})} w_T^i \tag{8}$$

Recall that k is the length of the cycle and we take $(j \mod k)$ in the indexing. This is the flow entering the rest of the cycle from C_i via sets in $\mathcal{T}^i(S_{i,i+1}, \not\subset S_{i-1,i})$ minus the total flow leaving the cycle from outside C_i. Note that sets in $\mathcal{T}^i(S_{i-1,i}, \not\subset S_{i,i+1})$ do not figure in the above inequality.

Before we prove the validity of this class of inequalities, we apply them to our counter-example in Figure 1. Consider segment A, and the sequence of sets $S_{AB} = \{1,2\}$, $S_{BC} = \{3\}$, $S_{CA} = \{5\}$ in the intersections of the three consideration sets.

The cycle-flow equality going in the direction A, B, C is

$$w^B_{\{1,2\}} + w^C_{\{3\}} + w^C_{\{3,4\}} \geq w^A_{\{1,2\}}$$

which is violated by the given solution as

$$0 + 0 + 0 \ngeq 0.5.$$

The cycle-flow equality going in the direction A, C, B is

$$w^C_{\{5\}} + w^C_{\{4,5\}} + w^B_{\{3\}} + w^B_{\{1,3\}} + w^B_{\{2,3\}} \geq w^A_{\{1,5\}} + w^A_{\{2,5\}} + w^A_{\{5\}}$$

which is also violated by the given solution as

$$0 + 0 + 0 + 0 + 0 \not\geq 0 + 0.5 + 0.$$

Proposition 3. *The cycle-flow inequalities (8) are valid.*

Proof

We show that the weights induced by any feasible solution to $CDLP$ satisfies these inequalities, thereby proving validity for the dynamic program as per our definition.

Since any solution to $CDLP$ is a convex combination of weights given to sets S, it is sufficient to show that if w_S is the weight of a set S, the weights it induces in each consideration set in $SDCP$, namely $w^i_{S\cap C_i} = w_S$ satisfies the inequality. Our proof is based on the flow graph defined earlier (Figure 9). The

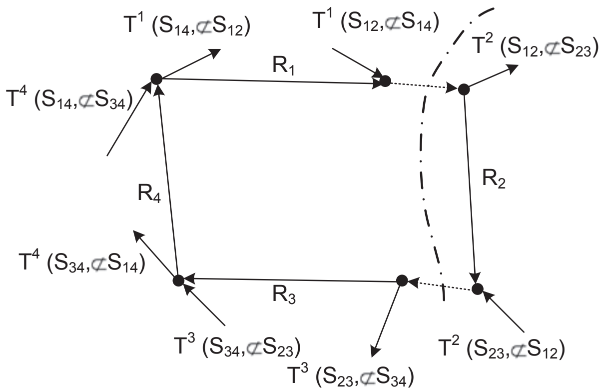

Figure 10: The cut for segment 2 for the flow graph of Figure 8

flow-balance at each node follow from the validity of the cuts (3).

To keep the notation easy to follow, we describe the proof in terms of this Figure 9 and looking at segment 2 in isolation as in Figure 10. This avoids the messy summations over $\mathcal{T}$'s and $\mathcal{R}$'s and the indexing, without sacrificing argument clarity.

Suppose the weights induced by the set S violate the cycle-flow inequalities (8). This implies the right-hand side of (8) is > 0 which can only happen if $w^2_{T^2(S_{23},\not\subset S_{12})} > 0$.

Starting from S_{23} and going clockwise, if we add all the flow-balance equalities at each node in Figure 9, we obtain:

$$\begin{aligned} 0 = \; & w^2_{T^2(S_{23},\not\subset S_{12})} - w^2_{T^2(S_{12},\not\subset S_{23})} + [w^3_{T^3(S_{34},\not\subset S_{23})} - w^3_{T^3(S_{23},\not\subset S_{34})}] + [w^4_{T^4(S_{14},\not\subset S_{34})} - w^4_{T^4(S_{34},\not\subset S_{14})}] \\ & + [w^1_{T^1(S_{12},\not\subset S_{14})} - w^1_{T^1(S_{14},\not\subset S_{12})}]. \end{aligned} \tag{9}$$

Going counter-clockwise just reverses the signs of all the terms. From (9),

$$\begin{aligned}
w^2_{T^2(S_{12},\not\subset S_{23})} = \quad & w^2_{T^2(S_{23},\not\subset S_{12})} + [w^3_{T^3(S_{34},\not\subset S_{23})} - w^3_{T^3(S_{23},\not\subset S_{34})}] + [w^4_{T^4(S_{14},\not\subset S_{34})} - w^4_{T^4(S_{34},\not\subset S_{14})}] \\
& +[w^1_{T^1(S_{12},\not\subset S_{14})} - w^1_{T^1(S_{14},\not\subset S_{12})}] \\
\geq \quad & w^2_{T^2(S_{23},\not\subset S_{12})} - w^3_{T^3(S_{23},\not\subset S_{34})} - w^4_{T^4(S_{34},\not\subset S_{14})} - w^1_{T^1(S_{14},\not\subset S_{12})} \\
> \quad & 0,
\end{aligned}$$

where the strict inequality is due to our assumption that the right-hand side of (8) is > 0. Our basic observation is the following: For any set of weights of subsets in segment 2's consideration set *induced by a set* S, at most one of these $w^2_{T^2(S_{23},\not\subset S_{12})}, w^2_{T^2(S_{12},\not\subset S_{23})}, w^2_{R_2}$ can have a positive value, simply because a set S cannot intersect C_2 in two different ways.

So we cannot have $w^2_{T^2(S_{12},\not\subset S_{23})} > 0$ and we derive a contradiction.

□

6.2 Cycle-synchronization Inequalities

In this section we propose another class of valid inequalities called the synchronization cuts that tighten the $SDCP+$ bound when cycles are present in the intersection graph.

We illustrate the idea underlying the cuts on the example in §3.3 with the $SDCP+$ solution values: $w^A_{\{1,2\}} = w^A_{\{2,5\}} = 0.5$, $w^B_{\{2\}} = w^B_{\{1,2,3\}} = 0.5$, $w^C_{\{4\}} = w^C_{\{3,5\}} = 0.5$, and $w^l_{S_l} = 0$ otherwise for all $l \in \{A, B, C\}$, $S_l \subset C_l$. We pick an arbitrary segment on the cycle in the support of this solution, say segment A. For an element $\bar{S} \subset C_A$ in the support of w^A, we consider the corresponding synchronization tree, by which we mean a hierarchical enumeration of all offer sets for each segment along the cycle that may be offered (with some unknown weight) synchronously with $\bar{S}$. Each level of this tree corresponds to a segment on the cycle, and on each level, we enumerate all compatible sets.

For example, Figures 11 and 12 depict the synchronization tree for $\bar{S} := \{1, 2\}$ and $\bar{S} := \{2, 5\}$, respectively. The second level (corresponding to segment B) comprises all sets $\{S^B | S^B \cap C_A = \bar{S} \cap C_B\}$, the third level (corresponding to segment C), corresponds to all subsets of C_C in $\{S^C | S^C \cap C_B = S^B \cap C_C, S^C \cap C_A = \bar{S} \cap C_C\}$ for all S_B on the second level. Note that since segment C is the last in the cycle when starting from segment A, the additional condition $S^C \cap C_A = \bar{S} \cap C_C$ ensures that the sets are "compatible" with $\bar{S}$ in the solution for segment A. We denote the trees in Figure 11 and 12 by T_1 and T_2 respectively, and paths p in the trees by $T_1(p)$ and $T_2(p)$ respectively.

Given a synchronization tree, say $T_1(\{1, 2\})$, we define valid inequalities as follows: given the weight of a current solution corresponding to the root node, we know that the same weight must be present along each level of the tree. E.g., for $\bar{S} := \{1, 2\}$ of C_A, the first level has the constraint

$$w^A_{\{1,2\}} = w^{B,T_1(\{1,2\})}_{\{1,2\}} + w^{B,T_1(\{1,2\})}_{\{1,2,3\}}$$

associated with it. Note that we need to introduce new tree-specific variables $w_{\{1,2\}}^{B,T_1(\{1,2\})}$ and $w_{\{1,2,3\}}^{B,T_1(\{1,2\})}$ since the sets {1,2} and {1,2,3} for segment B might only be synchronous with {1,2} on segment A for a fraction of the overall offer weight. However, we do know that for the current given *SDCP+* solution, it must hold that these two new variables are bounded from above via $w_{\{1,2\}}^{B,T_1(\{1,2\})} \leq w_{\{1,2\}}^{B}$ and $w_{\{1,2,3\}}^{B,T_1(\{1,2\})} \leq w_{\{1,2,3\}}^{B}$. The same logic applies to the remaining branches, overall giving the following valid inequalities associated with the synchronization tree in Figure 11:

$$
\begin{aligned}
w_{\{1,2\}}^{A} &= w_{\{1,2\}}^{B,T_1(\{1,2\})} + w_{\{1,2,3\}}^{B,T_1(\{1,2\})} && (10)\\
w_{\{1,2\}}^{B,T_1(\{1,2\})} &\leq w_{\{1,2\}}^{B} && (11)\\
w_{\{1,2,3\}}^{B,T_1(\{1,2\})} &\leq w_{\{1,2,3\}}^{B} && (12)\\
w_{\{1,2,3\}}^{B,T_1(\{1,2\})} &= w_{\{3\}}^{C,T_1(\{1,2\},\{1,2,3\})} + w_{\{3,4\}}^{C,T_1(\{1,2\},\{1,2,3\})} && (13)\\
w_{\{3\}}^{C,T_1(\{1,2\},\{1,2,3\})} &\leq w_{\{3\}}^{C} && (14)\\
w_{\{3,4\}}^{C,T_1(\{1,2\},\{1,2,3\})} &\leq w_{\{3,4\}}^{C} && (15)\\
w_{\{1,2\}}^{B,T_1(\{1,2\})} &= w_{\emptyset}^{C,T_1(\{1,2\},\{1,2\})} + w_{\{4\}}^{C,T_1(\{1,2\},\{1,2\})} \\
w_{\emptyset}^{C,T_1(\{1,2\},\{1,2\})} &\leq w_{\emptyset}^{C} \\
w_{\{4\}}^{C,T_1(\{1,2\},\{1,2\})} &\leq w_{\{4\}}^{C}. && (16)
\end{aligned}
$$

For the current solution (w^l) as defined above, equation (10) is satisfied $w_{\{1,2\}}^{A} = 0.5 = 0 + 0.5$ since (11) requires $w_{\{1,2\}}^{B,T_1(\{1,2\})} = 0$ and (12) allows $w_{\{1,2,3\}}^{B,T_1(\{1,2\})} = 0.5$. However, equation (13) is violated because $w_{\{1,2,3\}}^{B,T_1(\{1,2\})} = 0.5 \neq 0 + 0$; the right-hand side is zero because of inequalities (14–15).

Therefore, having found a violated constraint, we could add this constraint along with the constraints for higher levels of the same tree to *SDCP+* and re-solve. We are still guaranteed an upper bound on *CDLP* because a feasible *CDLP* solution can always be projected onto the consideration sets of each segment so that all synchronization cuts are satisfied for certain weights. In the current formulation, we require

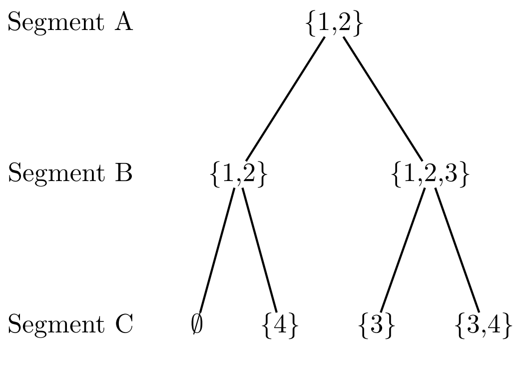

Figure 11: Full synchronization tree for $S_A = \{1, 2\}$.

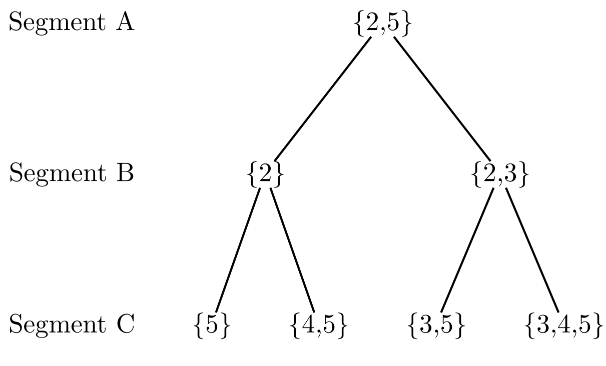

Figure 12: Full synchronization tree for $S_A = \{2, 5\}$.

an undesirably large number of additional variables (one per node in the synchronization tree). This can be remedied to some extent in two steps: first, note that all products that are considered by exactly one segment only can be removed from the tree, as they do not play a role as far as the offer set synchronization is concerned.

For example, product 4 is the only product consider by exactly one segment, hence we can remove it from the trees to obtain the reduced trees in Figures 13 and 14. The arising synchronization cuts corresponding to $\bar{S} := \{1,2\}$ are:

$$\begin{aligned}
w^{A}_{\{1,2\}} &= w^{B,T_1(\{1,2\})}_{\{1,2\}} + w^{B,T_1(\{1,2\})}_{\{1,2,3\}} \\
w^{B,T_1(\{1,2\})}_{\{1,2\}} &\leq w^{B}_{\{1,2\}} \\
w^{B,T_1(\{1,2\})}_{\{1,2,3\}} &\leq w^{B}_{\{1,2,3\}} \\
w^{B,T_1(\{1,2\})}_{\{1,2,3\}} &= w^{C,T_1(\{1,2\},\{1,2,3\})}_{\{3\}} \\
w^{C,T_1(\{1,2\},\{1,2,3\})}_{\{3\}} &\leq w^{C}_{\{3\}} + w^{C}_{\{3,4\}} \\
w^{B,T_1(\{1,2\})}_{\{1,2\}} &= w^{C,T_1(\{1,2\},\{1,2\})}_{\emptyset} \\
w^{C,T_1(\{1,2\},\{1,2\})}_{\emptyset} &\leq w^{C}_{\emptyset} + w^{C}_{\{4\}}.
\end{aligned}$$

The *SDCP+* solution still violates these constraints, so that adding them will tighten the bound. The

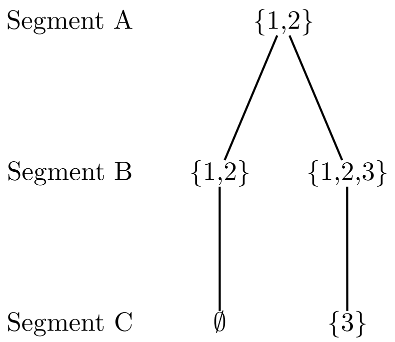

Figure 13: Reduced synchronization tree for $S_A = \{1,2\}$.

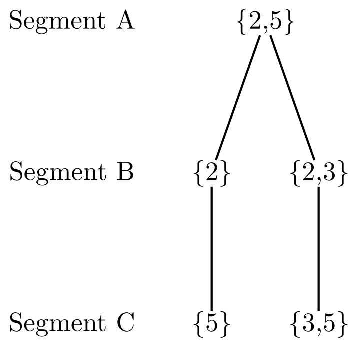

Figure 14: Reduced synchronization tree for $S_A = \{2,5\}$.

number of additional variables can be reduced further; in fact, we can obtain valid inequalities by introducing only one variable $y^{T_1}_p$ for each path p from the root S_A to each leaf of the reduced synchronization tree (rather than for each node). We illustrate this approach on the example in Figure 13: preservation of flow dictates that

$$w^{A}_{\{1,2\}} = \sum_{p} y^{T_1}_p. \tag{17}$$

The flow along path p is limited by the minimum over the flows on all edges contained in this path, which itself is bounded by the weights $w^{l}_{S_l}$ on each node in the tree. For the synchronization tree in Figure 13, we have two paths, namely $p_1 : \{1,2\} \to \{1,2\} \to \emptyset$ and $p_2 : \{1,2\} \to \{1,2,3\} \to \{3\}$. This gives us the following inequalities:

$$y_{p_1}^{T_1} \le w_{\{1,2\}}^{B} \tag{18}$$
$$y_{p_1}^{T_1} \le w_{\emptyset}^{C} + w_{\{4\}}^{C} \tag{19}$$
$$y_{p_2}^{T_1} \le w_{\{1,2,3\}}^{B} \tag{20}$$
$$y_{p_2}^{T_1} \le w_{\{3\}}^{C} + w_{\{3,4\}}^{C}. \tag{21}$$

The constraints (17–21) are violated by the solution of our running example since $w_{\{1,2\}}^{B} = 0$ and $w_{\{3\}}^{C} + w_{\{3,4\}}^{C} = 0$ imply $y_{p_1}^{T_1} = y_{p_2}^{T_1} = 0$, so that the equality constraint

$$w_{\{1,2\}}^{A} = 0.5 \ne 0 = y_{p_1}^{T_1} + y_{p_2}^{T_1}$$

is violated.

Proposition 4. *The constraints (17–21) are valid inequalities.*

Proof
As we noted earlier, as any solution to $CDLP$ is a convex combination of weights given to sets S, it is sufficient to show that if w_S is the weight of a set S, the weights it induces in each consideration set in $SDCP$, namely $w^i_{S\cap C_i} = w_S$, satisfy the inequalities. Also, to avoid needless notation and clutter, we show the validity for the specific synchronization tree of Figure 13. All the new variables are specific to the tree and therefore are independent of variables introduced for the other cuts.

Let $S \subseteq J$ be a set with $w_S > 0$ in a solution to $CDLP$, and let $w^l_{S\cap C_l} := w_S$ and $w^l_{S_l} := 0$, for all $S_l \subset C_l$, $S_l \ne S \cap C_l$, for all segments l along the cycle. The synchronization constraints (17–21) are associated with Figure 13. If $S \cap C_A \ne \{1, 2\}$, then we can set $y_p^{T_1} = 0$ for all paths p in this tree, and all constraints are satisfied. Otherwise, we have $S \cap C_A = \{1, 2\}$, and we show that the induced weights form a unique path in the tree with root in $S \cap C_A$. It holds that $(S \cap C_B) \cap C_A = (S \cap C_A) \cap C_B$, hence the set $S^B = S \cap C_B$ is on the tree in level B. Next, $(S \cap C_B) \cap C_C = (S \cap C_C) \cap C_B$, and since also $(S \cap C_C) \cap C_A = (S \cap C_A) \cap C_C$—note that this is the end of the cycle—, the node $S \cap C_C$ is under the branch of $S \cap C_B$. Therefore, there is exactly one path $\tilde{p}$ along the tree with positive weights of w_S^{CDLP} at every level, and setting $y_{\tilde{p}}^{T_1} := w_S$ and $y_p^{T_1} := 0$ otherwise satisfies the constraints.

□

7 Numerical Results

In this section we implement the two classes of cuts to examine their power and running times. All methods were tested in Matlab R2012a, running on an Intel i7-2600K CPU at 3.4 GHz.

7.1 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. [3], 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. 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).
- $SDCP$: Segment-based deterministic concave program as defined in §3.2.
- $SDCP^{+}$: Segment-based deterministic concave program with product constraints (3) for subsets $|S_{lk}| \leq 2$.
- $SDCP^{+F}$: We solve $SDCP^{+}$ and add any flow constraints of the form (8) that are violated by searching over all segments i and all combinations of subsets $\{\{S_{1,2}\}, \{S_{2,3}\}, \ldots, \}$, checking both directions around each cycle, and re-solve. Our reported results are based on full enumeration of all subsets $S_{i,i+1}$ for each cycle , and each segment on the cycle when searching for violated constraints. If there are many products in the overlap $C_i \cap C_{i+1}$, then a heuristic approach would need to be chosen. We pre-computed the sets $\mathcal{T}_i$ since they have few elements and only depend on the definition of the consideration sets.
- $SDCP^{+S}$: We solve $SDCP^{+}$ and add any synchronization constraints that are violated by its solution as discussed in §6.2, and re-solve. The cycle constraints are generated as follows: for each cycle, we define an arbitrary segment on the cycle as the root segment. For a given segment-level solution, we generate inequalities for a synchronization tree with root in an offer set with positive weight (of course only if the corresponding constraints have not been included yet). We use the formulation that introduces only one additional variable for each leaf in the sync tree. The iterations are stopped once no more violated constraints are found. Note that the tree structures can be pre-computed since they only depend on the segments' consideration sets.

7.2 Test Networks

We test the effectiveness of the various approaches on a small network with 3 resources, as well as on a larger network.

7.2.1 Parallel Flights Example With a Cycle

The first network example consists of three parallel flight legs as shown in Figure 15 with initial leg capacity 100. On each flight there is a low and a high fare class, L and H, respectively, with fares as specified in

Table 4. We define three customer segments as in Table 5; the preference values for the no-purchase option is assumed to be negligible (we set it to 0.001). The sales horizon consists of 100 time periods.

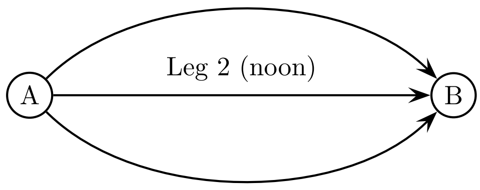

Figure 15: Parallel flights example with cycle.

Product	Leg	Class	Fare
1	1	L	50
2	1	H	70
3	2	L	100
4	2	H	110
5	3	L	50
6	3	H	70

Table 4: Product definitions for parallel flights example with cycle.

Segment	Consideration set	Pref. vector	λ_l	Description
1	{1,2,3,4}	[7,6,6,5]	0.23	Morning preference
2	{3,4,5,6}	[8,5,8,6]	0.26	Noon preference
3	{1,5}	[3,5]	0.2	Cheap flights only

Table 5: Segment definitions for the parallel flights example with cycle.

In Table 6 we report upper bounds on the optimal expected revenue obtained from our various approaches. Both $SDCP^{+F}$ and $SDCP^{+S}$ remove the gap to $CDLP$ completely. Runtimes are not reported since the example is too small (< 0.01 seconds for all scenarios).

7.2.2 Two Hubs Network Example

To test the performance of the two classes of cuts with regard to both runtime and upper bound, we consider a larger hub and spoke network of a type similar to the ones used in [11], but with segments defined to have many cycles in the intersection tree. 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 $2B$).

α	v_0	SDCP	SDCP+	SDCP+F	SDCP+C	CDLP	ΔSDCP+	ΔSDCP+F	ΔSDCP+S
	[0.01 0.01 0.01]	6378	5728	5610	5610	5610	2.1	0.0	0.0
0.6	[0.1 0.1 0.1]	6272	5647	5553	5553	5553	1.7	0.0	0.0
	[0.2 0.2 0.2]	6158	5562	5492	5492	5492	1.3	0.0	0.0
	[0.01 0.01 0.01]	6378	5728	5610	5610	5610	2.1	0.0	0.0
0.8	[0.1 0.1 0.1]	6272	5647	5553	5553	5553	1.7	0.0	0.0
	[0.2 0.2 0.2]	6158	5562	5492	5492	5492	1.3	0.0	0.0
	[0.01 0.01 0.01]	6378	5728	5610	5610	5610	2.1	0.0	0.0
1	[0.1 0.1 0.1]	6272	5647	5553	5553	5553	1.7	0.0	0.0
	[0.2 0.2 0.2]	6158	5562	5492	5492	5492	1.3	0.0	0.0
	[0.01 0.01 0.01]	6378	5728	5610	5610	5610	2.1	0.0	0.0
1.2	[0.1 0.1 0.1]	6272	5647	5553	5553	5553	1.7	0.0	0.0
	[0.2 0.2 0.2]	6158	5562	5492	5492	5492	1.3	0.0	0.0
	[0.01 0.01 0.01]	6378	5728	5610	5610	5610	2.1	0.0	0.0
1.4	[0.1 0.1 0.1]	6272	5647	5553	5553	5553	1.7	0.0	0.0
	[0.2 0.2 0.2]	6158	5562	5492	5492	5492	1.3	0.0	0.0

Table 6: Upper bounds for Parallel Flights Example. "ΔX" denotes the percentage gap to CDLP, computed by (X-CDLP)/CDLP*100. α is a scaling parameter for the leg capacities, and v_0 denotes the no-purchase preference vector.

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. In Figure 16 represents an example with $B = 2$.

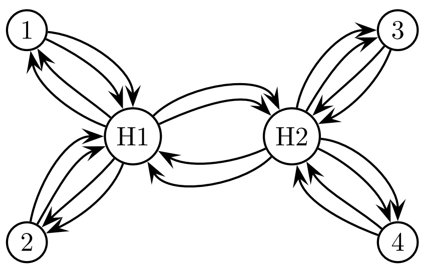

Figure 16: Two Hubs Network example with two hubs and $B = 2$ spokes each.

There are $4B^2 + 6B + 2$ 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 7. If the fares are not in the order $Y > M > Q > G > T$, then we re-sample until that order is obtained.

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 7: Mean fares for different itinerary types and booking classes.

The customer segments are defined as follows: for each origin-destination (OD) pair, there are between one and three itineraries possible, and there is a customer segment for each itinerary considering all booking classes on that itinerary only. These segments represent customers who are less price sensitive and not flexible in their choice of itinerary. In addition, there is a segment of less price sensitive customers who are flexible with regard to itinerary but not with regard to advance purchase requirements, so that this segment considers classes Y and M for all itineraries for this OD. Likewise, a final segment for this OD consists of customers who are price sensitive and flexible both in itinerary as well as in time, and hence only consider the advance purchase classes G and T across all itineraries. Therefore, for every OD pair, we have between three and five customer segments. For illustration, we depict the consideration sets for an OD with three itineraries in Figure 17.

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 (γ,β) defined in Table 8. The preference values v_0 for the non-purchase option, denoted by 0, are identical for all segments and are stated with the results.

Type	Description	β	γ
1	{Y,M} only, all itineraries	-0.001	15
2	{G,T} only, all itineraries	-0.01	20
3	Fixed itinerary only, all classes	-0.005	poissrnd(17)

Table 8: Types of customer segments for every OD pair. Parameters β and γ define the mean of preference value distribution. For all segments of type 3, γ is sampled from the Poisson distribution with mean 17.

The arrival rate for each segment is constructed by sampling a vector $b \in \mathbb{Z}^L$ from the discrete uniform distribution between 1 and L, and setting $\lambda = (0.7/\vec{1}^T b). * b$. There are 3,000 time periods.

For the network with $B = 2$ $(B = 4)$ spokes per hub, all short-haul flight legs have a capacity of 100 (70)

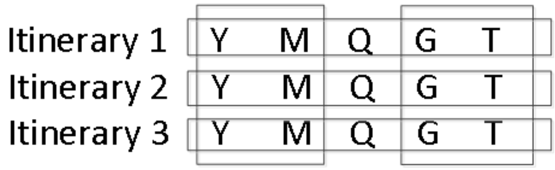

Figure 17: Illustration of consideration sets of five segments for an OD with three possible itineraries.

seats, and all long-haul flight legs have capacity of 200 (120) seats. These capacities are jointly being scaled up or down via a factor $\alpha \in \{0.6, 0.8, 1.0, 1.2, 1.4, 1.6\}$ in order to observe the effect of varied network load. To illustrate the size of the network instances, we report the specifications for the cases $B \in \{2, 4\}$ in Table 9. The corresponding intersection graph has cycles; in fact, every OD pair with more than one itinerary has cycles. For $SDCP^{+S}$, we dynamically generated synchronization constraints if they where violated by the

B	Legs	OD pairs	Itineraries	Products	Segments
2	20	30	56	280	116
4	36	90	172	860	352

Table 9: Two Hubs Network specifications.

incumbent solution. For each OD with more than one itinerary, we tested for such violations on one cycle. The upper bounds produced by the various approaches are reported in Tables 10 and 11 for $B = 2$ and $B = 4$, and the respective required CPU times in Tables 12 and 13.

Both $SDCP^{+F}$ and $SDCP^{+S}$ produce tight bounds in time considerably shorter than $CDLP$; however, the improvements over $SDCP^{+}$ are not as strong as in the Parallel Flights example. We also tested checking all cycles on the intersection tree for a given OD pair, but the results not not substantially improved whilst runtime suffered. A disadvantage of the synchronization cuts is that many constraints with new variables need to be generated if there are many segments on the cycle; therefore, it should be applied only if there are relatively few segments on a cycle.

8 Conclusions

The $CDLP$ formulation gives an upper bound on the dynamic program value function, but is computationally intractable for realistically sized applications when segment consideration sets overlap. The Sales-based Linear Program ($SBLP$) and the Segment-based Deterministic Concave Program ($SDCP$) are two weaker approximations that can be tightened ($SBLP+$ and $SDCP+$), while still maintaining tractability, by adding valid inequalities. Their performance on test problems has been outstanding. Naturally, this raises the question on what conditions can guarantee equivalence of the formulations.

In this paper, we obtain a structural result to this end, namely that $CDLP$ and $SDCP+$ are equivalent

α	v_0	$SDCP$	$SDCP^+$	$SDCP^{+F}$	$SDCP^{+S}$	$CDLP$	$\Delta SDCP$	$\Delta SDCP^+$	$\Delta SDCP^{+F}$	$\Delta SDCP^{+S}$
0.6	1	244889	243232	242634	242708	241835	1.3	0.6	0.3	0.4
	2	240609	238551	237764	237894	236825	1.6	0.7	0.4	0.5
	4	231417	228234	227976	227847	226444	2.2	0.8	0.7	0.6
0.8	1	306040	300366	298699	299233	295861	3.4	1.5	1.0	1.1
	2	295446	288829	287816	287981	284739	3.8	1.4	1.1	1.1
	4	278149	271170	271123	270598	267448	4.0	1.4	1.4	1.2
1	1	352696	342458	341036	341369	336285	4.9	1.8	1.4	1.5
	2	336233	327009	326254	326067	320030	5.1	2.2	1.9	1.9
	4	312040	304363	304362	303497	298795	4.4	1.9	1.9	1.6
1.2	1	387777	376175	374663	374557	366891	5.7	2.5	2.1	2.1
	2	366922	356647	356469	355278	347560	5.6	2.6	2.6	2.2
	4	334616	326028	326028	324769	320236	4.5	1.8	1.8	1.4
1.4	1	413773	400737	400361	398483	390050	6.1	2.7	2.6	2.2
	2	387027	374863	374719	373294	366554	5.6	2.3	2.2	1.8
	4	349735	340044	340044	338970	333447	4.9	2.0	2.0	1.7
1.6	1	430934	415821	415341	413586	405149	6.4	2.6	2.5	2.1
	2	401335	388445	388337	387215	378320	6.1	2.7	2.6	2.4
	4	359965	350366	350366	349512	342254	5.2	2.4	2.4	2.1

Table 10: Upper bounds for Two Hubs Network example with $B = 2$. "ΔX" denotes the rounded percentage gap to $CDLP$, computed by (X-$CDLP$)/$CDLP$*100.

α	v_0	$SDCP$	$SDCP^+$	$SDCP^{+F}$	$SDCP^{+S}$	$CDLP$	$\Delta SDCP$	$\Delta SDCP^+$	$\Delta SDCP^{+F}$	$\Delta SDCP^{+S}$
0.6	1	223958	222024	221782	221787	221028	1.3	0.5	0.3	0.3
	2	219996	217427	217067	216912	215929	1.9	0.7	0.5	0.5
	4	212611	209783	209506	209204	208247	2.1	0.7	0.6	0.5
0.8	1	278877	273846	272847	272572	270422	3.1	1.3	0.9	0.8
	2	269869	264675	264288	263736	261989	3.0	1.0	0.9	0.7
	4	255626	251298	251183	250636	249102	2.6	0.9	0.8	0.6
1	1	320221	312273	311331	310978	307957	4.0	1.4	1.1	1.0
	2	308133	301253	300775	300145	297475	3.6	1.3	1.1	0.9
	4	291107	285563	285471	284793	282182	3.2	1.2	1.2	0.9
1.2	1	355860	346876	345889	345483	341382	4.2	1.6	1.3	1.2
	2	342687	334703	334212	333554	329644	4.0	1.5	1.4	1.2
	4	322473	315660	315616	314841	310592	3.8	1.6	1.6	1.4
1.4	1	389140	379454	378452	378062	372356	4.5	1.9	1.6	1.5
	2	372510	364044	363644	362967	356909	4.4	2.0	1.9	1.7
	4	343654	336861	336834	336060	330415	4.0	2.0	1.9	1.7
1.6	1	415444	406025	405327	404791	396708	4.7	2.4	2.2	2.0
	2	391198	382847	382683	382022	373839	4.6	2.4	2.4	2.2
	4	355267	348075	348070	347497	340263	4.4	2.3	2.3	2.1

Table 11: Upper bounds for Two Hubs Network example with $B = 4$. "ΔX" denotes the rounded percentage gap to $CDLP$, computed by (X-$CDLP$)/$CDLP$*100.

α	v_0	$CDLP$	$SDCP$	$SDCP^+$	$SDCP^{+F}$	$SDCP^{+S}$
	1	23.8	0.3	0.7	6.9	0.9
0.6	2	30.6	0.3	0.7	9.0	1.2
	4	41.0	0.3	0.7	6.9	2.0
	1	25.1	0.3	0.7	7.0	1.5
0.8	2	36.5	0.3	0.7	6.9	1.8
	4	46.2	0.3	0.7	6.9	1.5
	1	36.3	0.3	0.7	6.9	1.8
1	2	27.6	0.3	0.7	9.0	3.5
	4	53.5	0.3	0.7	6.9	1.9
	1	45.5	0.3	0.7	8.9	1.8
1.2	2	37.7	0.3	0.7	7.0	3.3
	4	44.3	0.3	0.7	2.9	7.2
	1	28.5	0.3	0.7	9.0	2.3
1.4	2	38.5	0.3	0.7	6.9	7.7
	4	41.1	0.3	0.7	2.8	7.3
	1	32.0	0.3	0.7	9.0	7.5
1.6	2	25.8	0.3	0.7	6.9	7.8
	4	33.6	0.3	0.7	2.9	5.3

Table 12: Runtime in seconds for Two Hubs Network with $B = 2$.

α	v_0	$CDLP$	$SDCP$	$SDCP^+$	$SDCP^{+F}$	$SDCP^{+S}$
	1	668.2	1.1	4.8	28.7	5.7
0.6	2	746.8	1.1	4.7	34.5	9.8
	4	949.6	1.1	4.7	28.8	9.4
	1	741.8	1.1	4.7	35.0	9.7
0.8	2	944.7	1.1	4.7	29.1	10.3
	4	1160.9	1.1	4.7	22.6	15.2
	1	692.3	1.1	4.7	29.2	18.3
1	2	725.0	1.1	4.7	28.8	24.0
	4	980.2	1.1	4.7	16.9	23.2
	1	608.2	1.1	4.7	40.6	20.2
1.2	2	828.9	1.1	4.7	22.9	39.5
	4	1210.1	1.1	4.7	22.5	67.9
	1	791	1.1	4.7	34.8	54.4
1.4	2	971.5	1.1	4.7	28.7	86.4
	4	1147.2	1.1	4.7	16.7	67.7
	1	1049.9	1.1	4.7	29.1	153.5
1.6	2	681.5	1.1	4.7	22.5	94.9
	4	700.2	1.1	4.7	16.7	105.7

Table 13: Runtime in seconds for Two Hubs Network with $B = 4$.

if the intersection graph of the segment consideration sets is a tree (or a forest). This implies that these efficient solution methods will be guaranteed to yield the same control policies as $CDLP$ if demand can be modeled without cycles in the overlap structure of the segment consideration sets.

For consideration set structures with cycles, we propose two classes of valid inequalities that are relatively easy to generate systematically and which tighten the upper bounds on the optimal expected revenue significantly. We conduct extensive numerical experiments to validate the performance of these classes of cuts. Our structural result and the valid inequalities are applicable for all discrete choice models.

References

[1] Balseiro, S., C. Gocmen, G. Gallego, R. Phillips. 2011. Revenue management of consumer options for sporting events. Tech. rep., IEOR, Columbia University, New York.

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

[3] 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.

[4] Brumelle, S. L., J. I. McGill, T. H. Oum, K. Sawaki, M. W. Tretheway. 1990. Allocation of airline seat between stochastically dependent demands. *Transportation Science* **24** 183–192.

[5] 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.

[6] Gallego, G., R. Ratcliff, S. Shebalov. 2010. A general attraction model and an efficient formulation for network revenue management. Tech. rep., Department of IEOR, Columbia University.

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

[8] 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** 575–590.

[9] 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.

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

[11] Meissner, J., A. K. Strauss, K. Talluri. 2011. An enhanced concave programming method for choice network revenue management. *Production and Operations Management* (Forthcoming).

[12] Méndez-Díaz, I., J. Miranda Bront, G. Vulcano, P. Zabala. 2012. A branch-and-cut algorithm for the latent-class logit assortment problem. Forthcoming in *Discrete Applied Mathematics*.

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

[14] Shocker, A.D., M. Ben-Akiva, B. Boccara, P. Nedungadi. 1991. Consideration set influences on consumer decision-making and choice: Issues, models, and suggestions. *Marketing Letters* **2**(3) 181–197.

[15] 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.

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

[17] 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.

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