Choice-Based Network Revenue Management under Weak Market Segmentation

Joern Meissner
Kuehne Logistics University, Hamburg, Germany
joe@meiss.com

Arne K. Strauss
Department of Management Science, Lancaster University Management School, Lancaster, United Kingdom
a.strauss@lancaster.ac.uk

June 11, 2011

We present improved network revenue management methods that assume customers to choose according to the multinomial logit choice model with the specific feature that the sets of considered products of the different customer segments are allowed to overlap. This approach can be used to model markets with weak segmentation: For example, high-yield customer segments can be modeled to also consider low yield products intended for low-yield customers, introducing implicit buy-down behavior into the model.

The arising linear programs are solved using column generation and involve NP-hard mixed integer sub problems. However, we propose efficient polynomial-time heuristics that considerably speed-up the solution process. We numerically investigate the effect of varying the intensity of overlap on the respective policies and find that improvements are most pronounced in the case of high overlap, rendering the method highly interesting for weakly segmented market applications.

Key words: revenue management. dynamic programming/optimal control: applications, approximate.

1. Introduction

Many service providers like airlines or car rentals currently face a problem that shakes the foundations of their revenue management systems, namely an increasingly inefficient underlying market segmentation. Revenue management (RM) is based on exploiting the different price sensitivities of segments such as business and leisure customers. However, the pressure from low cost competition has led the service providers to cut many restrictions while their corporate customers increased control on their travel expenses. For instance, the business travel report Verband Deutsches Reisemanagement e.V. (2008) identified growing cost awareness and more systematic cost control as in former years in business travel for Germany, and a global one of American Express Business Travel (2008) even stated that "companies have focused on buying smarter by re-visiting, re-writing and enforcing travel policies, while encouraging and often requiring employees to book their trips online and further in advance via pre-trip auditing tools." As a result, business travelers increasingly also consider purchasing products that are intended for other segments and therefore undermine RM systems and cause revenue dilution since, in contrast, most RM models assume that the offered range of products can be partitioned into disjoint sets corresponding to the respective segments. In addition, often they furthermore assume that demand for the individual products is independent of the availability of alternatives.

In this paper, we extend network RM methods that are based on a multinomial logit customer choice model to allow for products to be under consideration for purchase by more than one segment corresponding to the more realistic situation described earlier. Very little work exists on this issue; so far we only know of Miranda Bront et al. (2009) who present a model that likewise assumes customers to choose according to a multinomial logit model that allows for overlapping consideration sets. They construct policies in the following way: First, to obtain approximations of the opportunity cost for the sale of a product, they propose a linear programming formulation that provides this information through the dual values of its capacity constraints. These values can then be used in a dynamic programming decomposition by the flight legs to obtain improved estimates that are eventually being used in the actual control policies to compute the set of products to offer at a given state. A specific issue with this approach is the large number of variables in the linear program so that large-scale techniques such as column generation need to be applied. The affiliated column generation subproblem of identifying the next column to include in the master problem is NP-hard as shown by Miranda Bront et al. (2009), however, they propose a polynomial time heuristic to accelerate the solution process.

We use their approach as a benchmark and contribute to the field by (1) extending the approaches of Zhang and Adelman (2009) and Meissner and Strauss (2009) to allow for overlapping consideration sets where we frequently need to solve NP-hard column generation subproblems, and thus (2) we propose efficient polynomial time heuristics that can be employed to considerably speed up these two methods. The resulting (3) policies are shown to outperform the policy of Miranda Bront et al. (2009). Solving the linear programs corresponding to the extended approaches of Zhang and Adelman (2009) and Meissner and Strauss (2009) is considerably more expensive than the benchmark method, however, policies derived from their solutions exhibit revenue improvement even without the expensive dynamic programming decomposition that is needed in the benchmark approach. (4) We numerically investigate the effect of varying the intensity of overlap on the respective policies and find that improvements are most pronounced in the case of high overlap of consideration sets, rendering them particularly attractive for application in weakly segmented markets.

The paper is organized as follows: In the next section we review the most related literature on network RM with and without accounting for choice behavior, before we present the problem framework, corresponding dynamic programming formulation and different solution approaches via linear programming in Section 3. These linear programs (LP) are solved with column generation, where the arising sub-problems can be solved via mixed integer LPs or greedy heuristics as discussed in Section 4. Having investigated solution techniques of the LP, we then turn our attention in Section 5 to the question of how to design dynamic policies based on the LP solution. We present numerical results in Section 6 and summarize our findings in Section 7.

2. Literature Review

A comprehensive description of both scientific and applied Revenue Management (RM) can be found in the book of Talluri and van Ryzin (2004b), and the reader interested in an overview of research over the last decades shall be referred to the reviews of McGill and van Ryzin (1999) and Chiang et al. (2007).

Network RM was in former years mostly researched under the assumption that demand for the offered products is independent of which alternatives the firm makes available to the customer. It is a valid assumption in the case that customer segments are well fenced off, and recent work includes for example Adelman (2007) and Topaloglu (2009). Optimal policies for the network RM problem can be obtained from solving a dynamic program, however, its high dimension makes this approach intractable. Therefore, one needs to approximate the value function to obtain heuristic policies. Adelman (2007) proposes a time-dependent approximation and shows that upper bounds on the optimal objective value are tightened relative to the standard so-called deterministic linear programming (DLP) approach, and that the obtained policies perform better in a simulation study. Similarly, Topaloglu (2009) improves on the DLP by using Lagrangian relaxation to obtain a time- and inventory-level dependent approximation. Farias and Van Roy (2007) are closely connected to this research in that they base their approximation on Adelman's using a linear programming approach to approximate dynamic programming that depends on both time and inventory level.

The same approximation was independently proposed by Talluri (2008) who focuses on the relationships of upper bounds on the optimal objective value of the aforementioned approaches by Topaloglu and Adelman, respectively, as well as the DLP and a randomized linear programming model.

The incorporation of choice behavior into network RM has increasingly gained attention as the means of segmentation erode in many markets. A prominent example is the airline industry, where the rise of the low-cost carriers has caused the incumbents to cut down fare restrictions, resulting in a weakly segmented market. As of the time of this writing, the financial crisis contributes even more to this as many business travelers show increased price sensitivity that further weakens the already rather low fences. In this situation, choice behavior becomes a crucial element in a RM system. Among the first approaches with a general model of customer choice is Talluri and van Ryzin (2004a) for a single flight leg problem. This work was generalized to the network context in Liu and van Ryzin (2008) with particular attention being paid to the multinomial logit choice model (MNL) with disjoint consideration sets, that means, customers from different segments do not consider the same product for purchase. Their approach was pioneered by Gallego et al. (2004) and improved by several groups of researchers including Zhang and Adelman (2009), Kunnumkal and Topaloglu (2008) and Meissner and Strauss (2009). Another major step was taken by Miranda Bront et al. (2009) in that they drop the assumption of disjoint consideration sets, since this provides the opportunity to model a situation where some customers do not obey the product fences any more.

Our work builds on the paper of Miranda Bront et al. (2009) and is closely related to Meissner and Strauss (2009) and Zhang and Adelman (2009) as outlined in the introduction.

3. Problem Formulation

3.1. Notation

In this section we introduce the basic notation that is used to describe the network revenue management models. Let us start with defining a product j: It consists of a unit of one or several resources, a fare price f_j and possibly certain restrictions and rules that the firm imposes in an attempt to segment the market. In total, let there be m resources indexed by i, each with an initial inventory level of c_i homogeneous units that can be used to accommodate all requests. The corresponding network capacity vector c is then given by $c=[c_1,\dots,c_m]^T$. There are n products defined, indexed by $j\in N$, where $N=\{1,\dots,n\}$ denotes the set of all products. We store information on which product is using which resource (and vice versa) in the incidence matrix $A\in\{0,1\}^{m\times n}$ by defining $a_{ij}:=1$ iff product j uses a unit of resource i, and setting $a_{ij}:=0$ otherwise. Therefore, the columns A^j of matrix A indicate which resources product j uses. Resource availability is reflected by the vector $x=[x_1,\dots,x_m]^T$, with x_i being the remaining inventory of resource i. Upon selling product j we need to commit capacity, hence available inventory levels change to $x-A^j$.

Customers are divided into L segments where customers within a given segment $l\in\{1,\dots,L\}=:\tilde{L}$ are considered to be homogenous in that they all have the same consideration set $C_l\subset N$ and product preferences v_{lj} for all products $j\in C_l$ in their consideration set. A very important point is that we allow for overlapping consideration sets reflecting the weak market segmentation. Every product can potentially be considered by several segments, and therefore from the firm's perspective demand cannot be affiliated with a certain segment.

The time horizon is partitioned into τ time periods that are small enough such that there is at most one customer arrival according to a Poisson process with arrival rate λ. Maturity of all resources is at time $\tau+1$ at which they become worthless. Decisions take place at the beginning of each time period before this period's demand can be observed (since demand depends on the offer set). An arriving customer belongs to segment l with probability p_l, $\sum_l p_l=1$, so that we can also define Poisson processes with rate $\lambda_l:=p_l\lambda$ for every segment. Taken together we have $\lambda=\sum_l\lambda_l$.

Choice Model. We are interested in modeling weak product segmentation, namely that the firm cannot prevent that customers from a certain segment might consider products targeted at some other segment as well. This situation can be represented by using the multinomial logit model (MNL) with overlapping consideration sets. The MNL model is constructed by using the preference vector v_l for each segment l as mentioned earlier to define the probability that a segment l customer purchases product j when the product set $S \subseteq N$is offered by

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

where v_{l0} is the preference for not buying anything. We set v_{lj} equal to zero if the product j is not offered or not in the consideration set C_l. A major advantage of this model is that every product which is considered by some segment l can be compared to the others in consideration set C_l, and intuitive probabilities can be derived that reflect preferences and offer set. The MNL model has been criticized for its so-called *independence from irrelevant alternatives* (IIA) property, namely, that

$$P_{j_1}(S_1)P_{j_2}(S_2) = P_{j_1}(S_2)P_{j_2}(S_1),$$

for all sets S_1, $S_2 \subset N$ and all products $j_1, j_2 \in S_1 \cap S_2$. In words, the probability that both products j_1 and j_2 are chosen is independent from whatever other products are available. Nevertheless, the model is attractive due to its intuitive design and particularly due to its analytical tractability which makes it one of the most popular choice models in the area of marketing and revenue management. Some choice models that avoid the IIA property are available, for an introduction see Talluri and van Ryzin (2004b).

Since the firm cannot distinguish with certainty between different segments, the purchase probability for product $j \in N$ given the offer set $S \subseteq N$ and the arrival of a customer is defined by

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

3.2. Optimality Equation

The RM problem can be stated as the following dynamic program, where $v_t(x)$ denotes the optimal expected revenue from having uncommitted network capacity vector x at time t:

$$v_t(x) = \max_{S \subseteq N(x)} \left\{ \sum_{j \in S} \lambda P_j(S)[f_j - (v_{t+1}(x) - v_{t+1}(x - A_j))] \right\} + v_{t+1}(x). \tag{2}$$

The boundary conditions are given by $v_{\tau+1}(x) = 0, \forall x \geq 0$. The set $N(x) := \{j \in N : x \geq A^j\}$ is the set of all feasible products that we can offer given available network capacity x. Theoretically, it is possible to solve this problem quite easily via backward dynamic programming, but the curse of dimensionality of the decision and state space makes it intractable for practical implementation. Thus we need methods to approximate the optimal value function but which reduce the computational load. The following linear programming formulation (**EQ**) will serve as the starting point of our considerations. It is equivalent to the dynamic program (2) which can be shown from fundamental results of value iteration, see Theorem 3.4.1 in Powell (2007).

$$(\mathbf{EQ}) \quad \min_{v(\cdot)} v_1(c)$$
$$v_t(x) \geq \lambda \sum_{j \in S} P_j(S) \left[f_j - (v_{t+1}(x) - v_{t+1}(x - A^j)) \right] + v_{t+1}(x) \qquad \forall t, x, S \subseteq N(x).$$

The decision variables are $v_t(x)$, $\forall\, t, x$, and (**EQ**) has a large number of constraints so that the problem becomes intractable for a large state space. Thus, the question is how to reduce the number of decision variables such that solving the LP becomes tractable (e.g. by using column generation on the dual). The basic idea is to approximate $v_t(\cdot)$ in (**EQ**) by given K basis functions $\phi_k(\cdot)$ in order to reduce the number of variables, that means one considers the approximation

$$v_t(x) \approx \sum_{k=1}^{K} V_{t,k}\phi_k(x) \qquad \forall\, t, x.$$

3.3. Time-Sensitive Approximation (TSA)

Zhang and Adelman (2009) considered the affine approximation

$$v_t(x) \approx \theta_t + \sum_{i=1}^{m} V_{t,i}x_i \qquad \forall\, t, x,$$

with boundary conditions $\theta_{\tau+1} = 0$ and $V_{\tau+1,i} = 0$ for all resources $i \in \{1, \ldots, m\}$. In this approximation, $V_{t,i}$ estimates the marginal value of an inventory unit of resource i in period t. Note that this does not take into account how much inventory is still available. The authors substitute the resulting approximation into (**EQ**) and construct its dual which is given below:

$$(\textbf{TSA}) \quad z_{\text{TSA}} = \max_{Y} \sum_{t,x,S\subseteq N(x)} \left[\sum_{j\in S} \lambda P_j(S) f_j \right] Y_{t,x,S}$$

$$\sum_{x,S\subseteq N(x)} x_i Y_{t,x,S} = \begin{cases} c_i & \text{if } t = 1, \\ \sum_{x,S\subseteq N(x)} (x_i - \sum_{j\in S} \lambda P_j(S) a_{ij}) Y_{t-1,x,S} & \forall\, t = 2, \ldots, \tau, \end{cases} \qquad \forall\, i,$$

$$\sum_{x,S\subseteq N(x)} Y_{t,x,S} = \begin{cases} 1 & \text{if } t = 1, \\ \sum_{x,S\subseteq N(x)} Y_{t-1,x,S} & \forall\, t = 2, \ldots, \tau, \end{cases}$$

$$Y_{t,x,S} \geq 0, \qquad \forall\, t, x, S \subseteq N(x).$$

The last set of equality constraints together with the non-negativity of Y induce that $Y_{t,x,S}$ can be seen as state-action probability in a fixed time period t. In this light, the first set of equality constraints can be interpreted as expected inventory at time $t > 1$ being equal to expected inventory at the previous time period minus the expected resource consumption in between, and being equal to the full capacity c_i at the beginning of the time horizon. Accordingly, the objective is maximization of expected revenue over the whole time horizon.

3.4. Time- and Inventory-Sensitive Approximation (TISA)

Meissner and Strauss (2009) present an approximation based on Farias and Van Roy (2007) which does not only account for dependency of the value function on time but also on the level of inventory that is still available. The motivation behind this approach is the intuitive notion that marginal utility of a resource decreases in its inventory level, hence a non-linear approximation should provide a better estimate of the true value function. The increased computational workload that can be traded off against approximation accuracy by splitting the inventory of every resource i into K_i inventory level ranges, and then to assign for each range k a variable $V_{t,i,k}$ which estimates the marginal resource value at any inventory level within this range at time period t. The number of inventory levels contained within range k is denoted by s_k^i, and can reach from unit size 1 to resource capacity c_i. Note, in particular, that it can vary between resources. For notational convenience, we also introduce for each resource i a function

$$r(\cdot) : \{0, 1, \ldots, c_i\} \to \mathbb{N},$$

for which $r(0) := 0$ and for $x_i > 0$ we set $r(x_i) := k$ if and only if inventory level x_i is contained in range k. We approximate the value function with

$$v_t(x) \approx \theta_t + \sum_{i=1}^{m} \left[\sum_{k=1}^{r(x_i)-1} s_k^i V_{t,i,k} + (x_i - \sum_{k=1}^{r(x_i)-1} s_k^i) V_{t,i,r(x_i)} \right]. \tag{3}$$

We further assume that $V_{\tau+1,i,k} = 0$ for all i, k, and that $\theta_{\tau+1} = 0$. The extreme cases are either to define only one inventory range for each resource resulting in less required computational effort, or to define as many ranges on every resource as it has capacity resulting in better policies through better approximation quality. In the former case, the resulting approximation is identical to the one used in TSA and its marginal values of capacity are independent of the inventory level, in the latter we are having the finest approximation which takes every possible inventory level into account. Therefore, in the following we confine ourselves to investigating these extreme cases, as they will provide the frame in between which the policy performance can be traded off against computational burden by adjusting the degree of inventory level aggregation.

Hence we consider the case $K_i = c_i$ for all resources i, which reduces the approximation (3) to

$$v_t(x) \approx \theta_t + \sum_{i=1}^{m} \sum_{h=1}^{x_i} V_{t,i,h}.$$

Substituting this approximation into (**EQ**) yields a linear program with a reduced number of variables, and forming its dual results in:

$$(\textbf{TISA}) \quad z_{\text{TISA}} = \max_{Y} \sum_{t,x,S \subseteq N(x)} (\lambda \sum_{j \in S} P_j(S) f_j) Y_{t,x,S} \tag{4}$$

$$\sum_{x,S \subseteq N(x)} Y_{t,x,S} \mathbf{1}_{\{x_i \geq h\}} = 1 \quad \forall h \in \{1, \dots, c_i\}, i, t = 1, \tag{5}$$

$$\sum_{x,S \subseteq N(x)} (Y_{t,x,S} - Y_{t-1,x,S}) \mathbf{1}_{\{x_i \geq h\}} - \lambda \sum_{x,S} \sum_{j \in S} P_j(S) a_{ij} Y_{t-1,x,S} \mathbf{1}_{\{x_i = h\}} = 0 \quad \forall h \in \{1, \dots, c_i\}, i, t > 1, \tag{6}$$

$$\sum_{x,S \subseteq N(x)} Y_{t,x,S} = 1 \quad \text{for } t = 1, \tag{7}$$

$$\sum_{x,S \subseteq N(x)} (Y_{t,x,S} - Y_{t-1,x,S}) = 0 \qquad \forall t > 1, \tag{8}$$

$$Y_{t,x,S} \geq 0 \quad \forall t, x, S \subseteq N(x). \tag{9}$$

To provide some intuition about this linear program, note that we can interpret $Y_{t,x,S}$ as state-action probabilities because of constraints (7,8,9). It follows that $\sum_S Y_{t,x,S}$ is the probability of being in state x at time t. Therefore, constraints (5) can be seen as probability of being in time $t = 1$ in a state x such that we have at least an inventory level of h at resource i. Since $t = 1$ is the start of the booking horizon, this probability must equal 1 for all h and i. Constraints (6) describe the further evolution over time, namely the probability of being at time t in a state x such that we have at least an inventory level of h at resource i is equal to the probability of the same situation at the previous time period $t - 1$ minus the expected consumption between decision time points $t - 1$ to t if the inventory x_i is currently at level h. The objective is again maximization of expected revenue over the whole time horizon.

3.5. Static Approximation CDLP

We use the approach of Miranda Bront et al. (2009) as a benchmark for our policies. Let us define the expected revenue from offering set S by $R(S) := \sum_{j\in S} f_j P_j(S)$, and the expected consumption of resource i by $Q_i(S) := \sum_{j\in S} a_{ij} P_j(S)$, and $Q(S) := [Q_1(S), \dots, Q_m(S)]^T$. They start from the following choice-based deterministic linear program (CDLP):

$$\begin{aligned} &z_{\text{CDLP}} = \max \sum_{S\subseteq N} \lambda R(S) t(S) \\ &\sum_{S\subseteq N} \lambda Q(S) t(S) \le c, \\ &\sum_{S\subseteq N} t(S) \le T, \\ &t(S) \ge 0, \quad \forall S \subseteq N. \end{aligned}$$

CDLP is solved via column generation and yields the optimal dual values π_i^* to the capacity constraints as static estimates for the marginal opportunity cost of each resource i. With this information one could define the approximation $v_t(x) \approx \sum_i \pi_i^* x_k$, but since it is static, a dynamic programming decomposition by the flight legs is used to refine the value function approximation and to introduce time- and capacity-dependence. We briefly outline this method when discussing policies.

4. Column Generation

Structural properties of the two approaches TSA and TISA were discussed in Zhang and Adelman (2009) and Meissner and Strauss (2009), respectively. Although both papers focus on the MNL model with disjoint consideration sets, all results related to upper bounds, efficient sets etc carry over since the choice probabilities were kept general.

However, matters become more difficult when it comes to the actual question of how to solve the arising large-scale problems. Both approaches generate linear programs with many variables but relatively few constraints. Hence column generation lends itself naturally to being the solution technique of choice, but only as long as we can relatively cheaply identify which columns shall enter the master problem. The column pricing problems are NP-hard as shown by Miranda Bront et al. (2009) and consume considerable effort as we might need to solve them several thousand times. In the case of disjoint consideration sets, this column generation subproblem was shown to be a simple ranking procedure. However, this is not possible any more if consideration sets may overlap. Thus we extend in the following the mixed integer linear programming formulations of Zhang and Adelman (2009) and Meissner and Strauss (2009) to the case of having a MNL choice model with overlapping consideration sets and propose greedy heuristics for both approaches that exhibit good practical performance as demonstrated in numerical experiments.

4.1. Approaches for TSA

The maximum reduced profit of the time-sensitive approximation (TSA) is given by

$$\max_{t,x,S\subseteq N(x)} \sum_{j\in S} \lambda P_j(S) \left[f_j - \sum_{i=1}^{m} a_{ij} V_{t+1,i} \right] - \sum_{i=1}^{m} (V_{t,i} - V_{t+1,i}) x_i - \theta_t + \theta_{t+1}.$$

This can be simplified by considering the problem for a fixed time period $t \ge 1$. We abbreviate the expression by defining the *worth* of a product as its associated revenue minus the approximate opportunity cost to provide this product in a given time period: $w_j := f_j - \sum_i a_{ij} V_{t+1,i}$. This leads us to

$$\max_{x, S \subseteq N(x)} \sum_{j \in S} \lambda P_j(S) w_j - \sum_{i=1}^{m} \underbrace{(V_{t,i} - V_{t+1,i})}_{=:\Delta V_i} x_i - \theta_t + \theta_{t+1}. \tag{10}$$

Before we transform this non-linear problem into a linear one, let us have a look whether we can eliminate x and thereby simplify the problem further. Zhang and Adelman (2009) showed that $V_{t,i}$ is non-increasing in time so that $\Delta V_i \geq 0$ for all i. Suppose we keep a certain set $S \subseteq N$ fixed, $S \neq \emptyset$, and we define

$$x(S) = [x_1(S), \ldots, x_m(S)]^T,$$

such that

$$x_i(S) := \max_{j \in S} a_{ij} \qquad \forall\, i \in \{1, \ldots, m\}.$$

We substitute this $x(S)$ into (10) and obtain a problem without dependence on x:

$$\max_{S \subseteq N} \sum_{j \in S} \lambda P_j(S) w_j - \sum_{i=1}^{m} \Delta V_i (\max_{j \in S} a_{ij}) - \theta_t + \theta_{t+1} \tag{11}$$

Proposition 1*An optimal solution to (11) is also optimal for (10), and for an optimal solution (x^*, S^*) to (10) S^* is optimal for (11) with the same objective value.*

Proof Let $S^* \subseteq N$ be an optimal solution to (11). Defining $x_i^* := \max_{j \in S^*} a_{ij}$, we clearly have feasibility to (10), that is $S^* \subseteq N(x^*)$. We refer to the objective value under the maximum in (10) as $f(x, S)$, and to the one of (11) as $g(S)$.

Suppose now that $(\tilde{x}, \tilde{S})$ is an optimal solution to (10) with $f(\tilde{x}, \tilde{S}) - g(\tilde{S}) > 0$. It follows that

$$\sum_{i=1}^{m} \overbrace{\Delta V_i (\max_{j \in \tilde{S}} a_{ij} - \tilde{x}_i)}^{\leq 0} > 0. \quad \lightning$$

Note that feasibility of $(\tilde{x}, \tilde{S})$ to (10) implies $\tilde{x}_i \geq a_{ij}$ for all $j \in \tilde{S}$, hence $\tilde{x}_i \geq \max_{j \in \tilde{S}} a_{ij}$ for all i. It follows that there is no optimal solution to (10) yielding a strictly higher objective in (11). The proposition follows from

$$f(\tilde{x}, \tilde{S}) \leq g(\tilde{S}) \leq g(S^*) = f(x^*, S^*).$$

For the second part of the proposition, let (x^*, S^*) be an optimal solution to (10). It follows that $f(x^*, S^*) \leq g(S^*)$. Suppose $f(x^*, S^*) < g(S^*)$. However, $(x(S^*), S^*)$ is feasible to (10) and $f(x(S^*), S^*) = g(S^*)$, which contradicts the optimality of (x^*, S^*).

□

Therefore, we can drop x from the optimization problem.

Mixed Integer Linear Programming approach. We transform now the nonlinear problem (11) into a mixed integer linear program: To that end, we replace the term $\max_{j \in S} a_{ij}$ by a non-negative variable ξ_i with appropriate constraints, resulting in the problem (12) below where we described the set S in terms of an availability vector $u \in \{0,1\}^n$ such that $u_j = 1$ if and only if $j \in S$. Accordingly, the general choice probabilities are now expressed in terms of u instead of S. Note that instead of integer variables x_i we have now real valued variables ξ_i.

$$\begin{aligned} \max_{\xi, u} \sum_{j \in N} \lambda P_j(u) w_j - \sum_{i=1}^{m} \Delta V_i \xi_i - \theta_t + \theta_{t+1} \\ \xi_i \geq a_{ij} u_j, \qquad \forall\, j \in N, \forall\, i : \Delta V_i > 0 \text{ and } a_{ij} > 0, \\ u \in \{0,1\}^n, \\ \xi \geq 0. \end{aligned} \tag{12}$$

We defined the incidence matrix A to be binary, that means, all products use at most one unit of any resource. This implies that $\max_{j\in N} a_{ij}u_j \in \{0,1\}$, and equal to 1 if and only if $\sum_{j\in N} a_{ij}u_j > 0$. Therefore, we can describe the maximum with a binary variable $\xi_i := \max_{j\in N} a_{ij}u_j$ in the following way: we enforce that $\xi_i = 0$ if the maximum is zero by imposing the constraints

$$\xi_i \leq \sum_{j\in N} a_{ij}u_j, \qquad \forall\, i : \Delta V_i > 0, \tag{13}$$

and likewise $\xi_i = 1$ if $\sum_{j\in N} a_{ij}u_j > 0$ by

$$\Big(\sum_{j\in N} a_{ij}\Big)\xi_i \geq \sum_{j\in N} a_{ij}u_j, \qquad \forall\, i : \Delta V_i > 0. \tag{14}$$

Overall we obtain:

$$\begin{aligned}
\max_{\xi,u} \sum_{j\in N} & \lambda P_j(u)w_j - \sum_{i=1}^{m} \Delta V_i \xi_i - \theta_t + \theta_{t+1} \\
& \text{subject to (13)–(14)}, \\
& u \in \{0,1\}^n, \\
& \xi_i \in \{0,1\}, \qquad \forall\, i : \Delta V_i > 0.
\end{aligned}$$

Having dealt with x, we turn our attention now towards the terms involving the choice probabilities $P_j(u)$. Depending on the choice model, these terms can cause difficulties in solving the problem since $P_j(\cdot)$ might be a complicated nonlinear function of u. We are interested in using the MNL choice model with overlapping consideration sets which provides the following purchase probabilities,

$$\lambda P_j(u) = \sum_{l=1}^{L} \lambda_l \frac{v_{lj}u_j}{\sum_{k\in C_l} v_{lk}u_k + v_{l0}},$$

as we defined them in (1). We substitute them into problem (12):

$$\begin{aligned}
\max_{\xi,u} \sum_{l=1}^{L} \sum_{j\in C_l} & \lambda_l \frac{v_{lj}u_j}{\sum_{h\in C_l} v_{lh}u_h + v_{l0}} w_j - \sum_{i=1}^{m} \Delta V_i \xi_i - \theta_t + \theta_{t+1} \\
& \xi_i \geq a_{ij}u_j, \qquad \forall\, j \in N, \forall\, i : \Delta V_i > 0 \text{ and } a_{ij} > 0, \\
& u \in \{0,1\}^n, \\
& \xi \geq 0.
\end{aligned}$$

This problem is nonlinear in the first term but can be transformed into a linear program by a change of variables. We introduce new variables $y_l := 1/(\sum_{h\in C_l} v_{lh}u_h + v_{l0})$ and $z_{lj} := u_j y_l$. The latter relationship is enforced by the following constraints:

$$z_{lj} \leq y_l, \qquad \forall\, l \in \tilde{L}, j \in C_l, \tag{15}$$
$$z_{lj} \geq 0, \qquad \forall\, l \in \tilde{L}, j \in C_l, \tag{16}$$
$$z_{lj} \leq M u_j, \qquad \forall\, l \in \tilde{L}, j \in C_l, \tag{17}$$
$$z_{lj} \geq y_l - M(1-u_j), \qquad \forall\, l \in \tilde{L}, j \in C_l, \tag{18}$$
$$u_j \in \{0,1\}, \qquad \forall\, j \in N. \tag{19}$$

We need to choose the constant M sufficiently large, but as small as possible since it is well-known that "Big M" methods can cause numerical difficulties and slow convergence for large M. Setting

$M := \max_l 1/v_{l0}$ is sufficiently large and a tight upper bound on y_l and z_{lj}. The definition of y_l is enforced by

$$y_l \geq 0, \qquad \forall\, l \in \tilde{L}, \tag{20}$$
$$\sum_{j \in C_l} v_{lj} z_{lj} + y_l v_{l0} = 1, \qquad \forall\, l \in \tilde{L}. \tag{21}$$

We obtain eventually the linear program below:

$$\begin{aligned} \max_{\xi,y,z,u} & \sum_{l=1}^{L} \sum_{j \in C_l} \lambda_l v_{lj} w_j z_{lj} - \sum_{i=1}^{m} \Delta V_i \xi_i - \theta_t + \theta_{t+1} \\ & \xi_i \geq a_{ij} u_j, \qquad \forall\, j \in N, \forall\, i : \Delta V_i > 0 \text{ and } a_{ij} > 0, \\ & \text{subject to (15)–(21)}, \\ & \xi \geq 0. \end{aligned}$$

The problem is similar to the one obtained by Zhang and Adelman (2009) in the special case of disjoint consideration sets, however, we replaced the integer variables x_i with continuous variables ξ_i. Furthermore, in the presence of overlapping consideration sets we might have considerably more variables than in the disjoint consideration set case since there are $\sum_l |C_l| > n$ variables z_{lj} in the above problem in contrast to the n variables z_{lj} for the case that consideration sets are disjoint. Potentially, there might be as many as nL variables z_{lj} if every segment would consider every product. Even without this added complexity, since the problem is NP-hard and has to be solved frequently it is of considerable practical interest to obtain polynomial time heuristics which reduce the computational effort. In the next section we propose such a heuristic that exploits our observation that x can be removed from the problem.

Greedy Heuristic for TSA. Based on the work of Prokopyev et al. (2005) on heuristics for general linear-fractional programmes, Miranda Bront et al. (2009) propose a heuristic for solving the column pricing problem under the MNL choice model with overlapping consideration sets for a simpler approximation, namely the choice-based deterministic linear program (CDLP).

For the affine approximation that we consider here, the column pricing problem (10) has an objective that not only depends on the offer set S, but also on the vector of available capacity x, both of which are interdependent due to the constraints $S \subseteq N(x)$. However, due to Proposition 1, we can focus on solving the problem (11) where we eliminated x instead of the original formulation (10). Our proposed Algorithm 1 attempts to solve problem (11) for a fixed time period t by adding products consecutively to an initially empty set in a greedy fashion. We start in line 1 by sorting out all products with non-positive worth w_j. Among the ones that remain, we then look for the single product that maximizes the objective (lines 2–10); in line 6 we make use of our earlier observation that an optimal x vector is given by $x(S)$ for any given offer set. In what follows, we continue to add products to the set $S = \{j^*\}$ in the same way. That is, we compute the objective for the current set S with a new product temporarily added, and choose to permanently add the one yielding the highest objective. We stop this process once (x, S) are not changed any more. The heuristic has worst-case complexity $O(n^2(L+m))$ and performs very well as exhibited by the numerical experiments reported below.

We emphasize that the heuristic can also be applied to the disjoint consideration case, therefore our heuristic also accelerates the solution of the model presented by Zhang and Adelman (2009).

Algorithm 1 Greedy heuristic for TSA column generation subproblem at time t

Let $\tilde{S} \subseteq N$ be the set of products with positive worth $w_j > 0\ \forall j \in \tilde{S}$
for all $j \in \tilde{S}$ **do**
  $\xi_j \leftarrow \sum_{l \in \tilde{L}} \lambda_l w_j v_{lj}/(v_{lj} + v_{l0})$
  for $i = 1$ to m **do** {define $x(\{j\})$}
    $\Delta V_i \leftarrow V_{t,i} - V_{t+1,i}$
    Define $x_i^j \leftarrow a_{ij}$
  end for
  $\zeta_j \leftarrow \Delta V^T x^j$
end for
$j^* \leftarrow \arg\max_{j \in \tilde{S}} (\xi_j - \zeta_j)$, set $S = \{j^*\}$, $\tilde{S} = \tilde{S} \setminus \{j^*\}$, $x \leftarrow x^{j^*}$
repeat
  for all $j \in \tilde{S}$ **do**
    $\xi_j \leftarrow \sum_{l \in \tilde{L}} \lambda_l \sum_{h \in C_l \cap (S \cup \{j\})} w_h v_{lh} / (\sum_{k \in C_l \cap (S \cup \{j\})} v_{lk} + v_{l0})$
    for $i = 1$ to m **do** {define $x(S \cup \{j\})$}
      $x_i^j \leftarrow \max_{h \in S \cup \{j\}} a_{ih}$
    end for
    $\zeta_j \leftarrow \Delta V^T x^j$
  end for
  $j^* \leftarrow \arg\max_{j \in \tilde{S}} (\xi_j - \zeta_j)$
  if reduced profit of $(t, x^{j^*}, S \cup \{j^*\}) >$ reduced profit of (t, x, S) **then**
    Set $S = S \cup \{j^*\}$, $\tilde{S} = \tilde{S} \setminus \{j^*\}$, $x \leftarrow x^{j^*}$
  end if
until (S, x) is not changed
return (S, x) which identifies the new column

4.2. Approaches for TISA

The linear program (**TISA**) arising from the time- and inventory-dependent approximation has likewise many columns but relatively few constraints. We need to solve the following problem to obtain the variable $Y_{t,x,S}$ with maximum reduced profit:

$$\max_{t,x,S \subseteq N(x)} \left[\lambda \sum_{j \in S} P_j(S) (f_j - \sum_{i=1}^{m} a_{ij} V_{t+1,i,x_i}) - \sum_{i=1}^{m} \sum_{h=1}^{x_i} (V_{t,i,h} - V_{t+1,i,h}) + \theta_{t+1} - \theta_t \right]. \tag{22}$$

We briefly present a mixed integer linear programming approach adapted to case of MNL choice model with overlapping consideration sets. Since this problem is of similar structure like the previous one for TSA, it follows that it is also NP-hard. We develop a polynomial time heuristic in the second part of this section.

Mixed Integer Linear Programming approach for TISA. Parallel to our course of action for the affine approximation, we substitute the MNL choice probabilities into the reduced profit equation (22) and obtain the following problem:

$$\begin{aligned} \max_{x,u} & \sum_{l=1}^{L} \sum_{j \in C_l} \lambda_l \frac{v_{lj} u_j}{\sum_{k \in C_l} v_{lk} u_k + v_{l0}} w_{x,j} - \sum_{i=1}^{m} \sum_{h=1}^{x_i} (V_{t,i,h} - V_{t+1,i,h}) - \theta_t + \theta_{t+1} \\ & x_i \geq a_{ij} u_j, \qquad \forall\, j \in N, \forall\, i : a_{ij} > 0, \\ & x_i \in \{0, \ldots, c_i\}, \qquad \forall\, i. \end{aligned}$$

Note that the product worth $w_{x,j} := f_j - \sum_i a_{ij} V_{t+1,i,x_i}$ now also depends on the currently available capacity x which makes the problem more complicated. However, similar manipulations as described in Meissner and Strauss (2009) can be used to reformulate the problem by introducing the variables

$$y_l := \frac{1}{\sum_{h \in C_l} v_{lh} u_h + v_{l0}} \text{ and } z_{lj} := u_j y_l,$$

with corresponding constraints as in the linear program for TSA. We eventually obtain the following linear program, which is equivalent to solving the reduced profit maximization (22):

$$\max_{x,y,z,\nu,u} \quad \sum_l \sum_{j\in C_l} \sum_i (-\lambda_l v_{lj} a_{ij}) \left[V_{t+1,i,1}\nu_{lj}^{1,i} + \sum_{k=2}^{c_i} (V_{t+1,i,k} - V_{t+1,i,k-1})\nu_{lj}^{ki} \right] + \\ + \sum_l \sum_{j\in C_l} (\lambda_l v_{lj} f_j) z_{lj} + \sum_i \sum_{k=1}^{c_i} (V_{t+1,i,k} - V_{t,i,k}) x^{ki} - \theta_t + \theta_{t+1}$$

$$\begin{aligned}
&\sum_{j\in C_l} v_{lj} z_{lj} + v_{l0} y_l = 1, && \forall\, l, \\
&y_l \geq 0, && \forall\, l, \\
&\sum_{k=1}^{c_i} x^{ki} \geq a_{ij} u_j, && \forall\, i, j \in N : a_{ij} > 0, \\
&x^{k-1,i} \geq x^{ki}, && \forall\, i, k \in \{2,\dots,c_i\}, && (23) \\
&x^{ki} \in \{0,1\}, && \forall\, i, k \in \{1,\dots,c_i\}, && (24) \\
&\nu_{lj}^{ki} \leq x^{ki}, && \forall\, l, j \in C_l, k, i, && (25) \\
&\nu_{lj}^{ki} \leq z_{lj}, && \forall\, l, j \in C_l, k, i, && (26) \\
&\nu_{lj}^{ki} \geq z_{lj} - (1 - x^{ki}), && \forall\, l, j \in C_l, k, i, && (27) \\
&\nu_{lj}^{ki} \geq 0, && \forall\, l, j \in C_l, k, i, && (28) \\
&u_j \in \{0,1\}, && \forall\, j \in N, && (29) \\
&z_{lj} \geq 0, && \forall\, l, \forall\, j \in C_l, && (30) \\
&z_{lj} \leq y_l, && \forall\, l, \forall\, j \in C_l, && (31) \\
&M(1-u_j) + z_{lj} \geq y_l, && \forall\, l, \forall\, j \in C_l, && (32) \\
&M u_j \geq z_{lj}, && \forall\, l, \forall\, j \in C_l. && (33)
\end{aligned}$$

In order to linearize the objective with respect to x, we represent x_i by $\sum_k x^{ki}$ and impose the constraints (23–24) to ensure that $x^{ki} = 1$ for $k = 1,\dots,x_i$, and $x^{ki} = 0$ otherwise. The variable ν_{lj}^{ki} replaces $z_{lj}x^{ki}$, and the constraints (25–28) ensure that $\nu_{lj}^{ki} = z_{lj}$ if $x^{ki} = 1$ and $\nu_{lj}^{ki} = 0$ otherwise. The constraints (29–33) represent $z_{lj} \in \{0, y_l\}$. The constant M can be chosen $M := \max_l 1/v_{l0}$ since $y_l \leq 1/v_{l0}$.

Greedy Heuristic for TISA. In order to avoid having to solve the above presented mixed integer linear problem, we propose Algorithm 2 to attempt to solve the column pricing problems (22) for each fixed time period t. Additional complexity is added by the fact that the objective now also depends on the respective inventory levels of all resources. In particular, the first term is now also dependent on x, making the two expressions inseparable. Overall worst-case complexity of Algorithm 2 is $O(n^2(mL + \sum_i c_i))$. Usually a good solution is found in much less operations so that considerable time can be saved as shown in the numerical experiments.

The underlying idea of this heuristic is, similarly to the previous one, to find a column—characterized by (t,x,S)—for a fixed time period t by consecutively adding products to an initially empty set in a greedy fashion, and to construct the best x for any product that is under consideration of being added. That is, we start with $S = \emptyset$, add a product $j \in N$ and compute the x^j that maximizes the reduced profit $\pi(t, x^j, \{j\})$ by exploiting that the reduced profit function is separable in the resources i. This is done for each product $j \in N$, so that we can choose the new product j^* which shall enter the set S to be the one with the largest reduced profit. Subsequently, for a given set S we look for the next—still unassigned—product $j \notin S$ which would contribute the most to the reduced profit, along with the best x vector which is computed for each candidate in lines 21 and 24 of Algorithm 2.

Algorithm 2 Greedy heuristic for TISA column pricing problem at time t

Set $S \leftarrow \emptyset$, $\tilde{S} \leftarrow N$
$\Delta V_{i,k} \leftarrow V_{t,i,k} - V_{t+1,i,k}$ for all resources i, for all $k = 1, \dots, c_i$
for all $j \in \tilde{S}$ **do**
    $\xi_j \leftarrow \sum_{l=1}^{L} \lambda_l v_{lj} / (v_{lj} + v_{l0})$
    for $i = 1$ to m **do** {Compute optimal vector x for any given j:}
        if $i \in A^j$ **then**
            $x_i^j \leftarrow \arg\max_{x_i a_{ij} \in \{a_{ij}, \dots, c_i\}} (-\xi_j a_{ij} V_{t+1,i,x_i} - \sum_{k=1}^{x_i} \Delta V_{i,k})$
        else
            $x_i^j \leftarrow \arg\max_{x_i \in \{0,1,\dots,c_i\}} (-\sum_{k=1}^{x_i} \Delta V_{i,k})$
        end if
    end for
    $\zeta_j \leftarrow \xi_j (f_j - \sum_i a_{ij} V_{t+1,i,x_i^j}) - \sum_i \sum_{k=1}^{x_i^j} \Delta V_{i,k}$
end for
Define $j^* \leftarrow \arg\max_{j \in \tilde{S}} \zeta_j$
Set $S = \{j^*\}$, $\tilde{S} = \tilde{S} \setminus \{j^*\}$, $x \leftarrow x^{j^*}$
repeat
    for all $j \in \tilde{S}$ **do** {Find best x if we offer $S \cup \{j\}$ with some fixed $j \in \tilde{S}$}
        for all resources i which are required to provide products $S \cup \{j\}$ **do**
            $\xi_i^j \leftarrow \sum_{l=1}^{L} \lambda_l \sum_{h \in C_l \cap (S \cup \{j\})} v_{lh} a_{ih} / (\sum_{k \in C_l \cap (S \cup \{j\})} v_{lk} + v_{l0})$
            $\alpha_i \leftarrow \max_{k \in S \cup \{j\}} a_{ik}$
            $x_i^j \leftarrow \arg\max_{x_i \in \{\alpha_i, \dots, c_i\}} (-\xi_i^j V_{t+1,i,x_i} - \sum_{k=1}^{x_i} \Delta V_{i,k})$
        end for
        for all other resources i **do**
            $x_i^j \leftarrow \arg\max_{x_i \in \{0,1,\dots,c_i\}} (-\sum_{k=1}^{x_i} \Delta V_{i,k})$
        end for
        Compute $\pi(t, x^j, S \cup \{j\}) :=$ reduced profit of column associated with $(t, x^j, S \cup \{j\})$
    end for
    Set $j^* \leftarrow \arg\max_{j \in \tilde{S}} \pi(t, x^j, S \cup \{j\})$
    if $\pi(t, x^{j^*}, S \cup \{j^*\}) > \pi(t, x, S)$ **then**
        Set $S \leftarrow S \cup \{j^*\}$, $\tilde{S} \leftarrow \tilde{S} \setminus \{j^*\}$, $x \leftarrow x^{j^*}$
    end if
until (x, S) are not changed
return (x, S) which identifies the new column (t is fixed)

5. Policies

Policies can be obtained by using the dual values from the linear programming solutions directly, or by using some kind of network decomposition. Essentially, dynamic programming decomposition schemes reduce the network problem to a set of leg-level problems which can be solved via backward dynamic programming due to their one-dimensionality. The key to this approach is the approximation of opportunity cost, or as it can equivalently be seen, the estimation of a revenue share for a particular flight leg. Since the dual values from the LP solutions provide us with information on marginal value of capacity, we can use this to formulate the single-leg problems. One feature of DP decomposition is that one obtains both time- as well as capacity-dependent bid prices. However, their quality is based upon the quality of the dual values which serve as an input to the DP decomposition procedure, so we expect better results with more accurate inputs.

The actual control policies are derived from the optimal policy prescribed by the dynamic programming formulation (2). Supposing that we knew the true value function $v_t(x)$, we would offer the following set S^* in time period t with remaining available network capacity x:

$$S^* = \arg\max_{S \subseteq N(x)} \sum_{j \in S} \lambda P_j(S) \Big[f_j - (v_{t+1}(x) - v_{t+1}(x - A^j)) \Big]. \tag{34}$$

Although we do not know the true value function, we do have knowledge about an approximation of it. Namely, one possibility is to directly use the result from the linear programs developed above to approximate the opportunity cost $v_{t+1}(x)-v_{t+1}(x-A^j)$, and another to refine it by conducting a dynamic programming procedure in between. In this section, we present both techniques applied to the approaches TSA and TISA. Furthermore, we briefly review the dynamic programming approach for CDLP of Miranda Bront et al. (2009) which we use as a benchmark.

5.1. Policies based on TSA

For the case of using the MNL choice model with disjoint consideration sets, Zhang and Adelman (2009) proposed both a dynamic programming decomposition method as well as a policy based on direct usage of the dual values obtained from solving a linear program. The basic approach remains the same in the context of overlapping consideration sets, however, the resulting problem that needs to be solved in any of the two methods in order to determine which set of products shall be offered at a given time period and for given remaining capacity is more difficult since it shares the same structure with the column generation subproblem. While in the disjoint consideration set case it is possible to solve the arising problem efficiently by a ranking procedure, we now need to resort to similar means as described for the column pricing problem: Either we transform the problem to a linear mixed integer program, or we use an adapted version of Algorithm 1.

5.1.1. Using Dual Values Directly. Given the dual solution V^* of (**TSA**), Zhang and Adelman (2009) approximated the opportunity cost by

$$v_t(x)-v_t(x-A^j)\approx\sum_i a_{ij}V^*_{t,i}.$$

They substitute this into the optimal policy to obtain a heuristic indicating which set S to offer at time t and given remaining capacity x:

$$\begin{aligned} S=\underset{u\in\{0,1\}^n}{\arg\max}&\sum_{l=1}^{L}\sum_{j\in C_l}\lambda_l\frac{v_{lj}u_j}{\sum_{j\in C_l}v_{lj}u_j+v_{l0}}\Big(f_j-\sum_i a_{ij}V_{t,i}\Big) \\ a_{ij}u_j\le x_i &\qquad \forall\, i,j\in N. \end{aligned} \tag{35}$$

The ranking procedure employed by Zhang and Adelman (2009) (originally proposed by Gallego and Phillips (2004)) does not work if the consideration sets overlap, thus we need to solve this problem either by transforming it into a linear mixed integer program like for the column generation subproblem, or we employ Algorithm 1 with slight amendments, that is initially we would remove all products j for which $x_i<a_{ij}$, and set $\Delta V_i=0$.

5.1.2. Dynamic Programming Decomposition. We can use the marginal capacity value estimates from (**TSA**) to decompose the network problem into a collection of m singe-resource dynamic programmes. Zhang and Adelman (2009) used the following approximation of the value function:

$$v_t(x)\approx v_t^i(x_i)+\sum_{k\neq i}V^*_{t,k}x_k \qquad \forall t,\, x.$$

When we substitute this into the optimal dynamic program (2), we obtain for each resource i a one-dimensional dynamic program:

$$\begin{aligned} v_t^i(x_i)=\max_{x_{-i},S\subseteq N(x)}&\sum_{j\in S}\lambda P_j(S)\Big(f_j-v^i_{t+1}(x_i)+v^i_{t+1}(x_i-a_{ij})-\sum_{k\neq i}V^*_{t+1,k}a_{kj}\Big) \\ &-\sum_{k\neq i}(V^*_{t,k}-V^*_{t+1,k})x_k+v^i_{t+1}(x_i). \end{aligned} \tag{36}$$

The term x_{-i} represents optimization over inventory vectors with the ith component removed (since it is fixed at x_i). On the boundary we define $v^i_{\tau+1}(x_i)=0$ for all resources i and all inventory levels x_i since the inventory is assumed to become worthless at the end of the booking horizon. We can interpret equation (36) as the maximization of expected *adjusted* revenue. The adjustment consists of two parts: The opportunity cost $(v^i_{t+1}(x_i)-v^i_{t+1}(x_i-a_{ij}))$ incurred by using a_{ij} units of resource i and the opportunity cost $V^*_{t+1,k}a_{kj}$ incurred by using a_{kj} units of resource k. The terms $(V^*_{t,k}-V^*_{t+1,k})x_k$ can be seen as the value deterioration of inventory of resource k with respect to time. The maximization is again of the same structure as the column pricing problem, therefore both the mixed integer linear program as well as the Algorithm 1 can be adapted to provide a (potentially sub-optimal) solution to (36).

Once we have solved for $v^i_t(x_i)$ for all t, i, x_i, a policy is given by

$$\max_{S\subseteq N(x)}\sum_{j\in S}\lambda P_j(S)\left[f_j-\sum_{i=1}^{m}a_{ij}(v^i_{t+1}(x_i)-v^i_{t+1}(x_i-1))\right]. \tag{37}$$

5.2. Policies based on TISA

5.2.1. Using Dual Values Directly Denote the optimal dual values to the capacity constraints (5) and (6) of (**TISA**) by V^*. We can use V^* directly to obtain the policy below, which needs to be solved in each time period for a given state x:

$$\max_{S\subseteq N(x)}\sum_{j\in S}\lambda P_j(S)\left[f_j-\sum_{i=1}^{m}a_{ij}V^*_{t+1,i,x_i}\right]. \tag{38}$$

5.2.2. Dynamic Programming Decomposition Let V^* be the optimal dual values associated with the capacity constraints (5) and (6) of (**TISA**). Then we can approximate the value function $v_t(x)$ in a similar manner as in the last section in order to obtain a set of resource-level problems. More specifically, for each resource i we use the approximation

$$v_t(x)\approx v^i_t(x_i)+\sum_{k\neq i}\sum_{h=1}^{x_k}V^*_{t,k,h}\qquad \forall t,x.$$

Plugging this into (**EQ**), we obtain for each resource i the following problem:

$$(\mathbf{LP_i})\quad \min_{v^i_t(\cdot)} v^i_1(c_i)+\sum_{k\neq i}\sum_{h=1}^{c_k}V^*_{1,k,h}$$

$$v^i_t(x_i)+\sum_{k\neq i}\sum_{h=1}^{x_k}V^*_{t,k,h}\geq\lambda\sum_{j\in S}P_j(S)\Bigg[f_j-(v^i_{t+1}(x_i)-v^i_{t+1}(x_i-a_{ij})$$
$$+\sum_{k\neq i}\sum_{h=x_k-a_{kj}+1}^{x_k}V^*_{t+1,k,h})\Bigg]+v^i_{t+1}(x_i)+\sum_{k\neq i}\sum_{h=1}^{x_k}V^*_{t+1,k,h}\qquad \forall t,x,S\subseteq N(x). \tag{39}$$

The proposition below shows that the objective functions to the problems ($\mathbf{LP_i}$) constitute upper bounds on the exact objective value $v_t(x)$ for each resource i. It is possible to refine bounds of TSA using dynamic programming decomposition since we essentially would relax the problem in as far as we allow the value function approximation to be non-linear. However, for TISA the dynamic programming bound turns out to be identical to the one provided by linear program (**TISA**) which provides evidence that DP decomposition will probably not improve our direct policies.

Proposition 2 *(i)* $v^i_t(x_i)+\sum_{k\neq i}\sum_{h=1}^{x_k}V^*_{t,k,h}\geq v_t(x)\qquad \forall t,i,x.$

(ii) $z_{\text{CDLP}} \geq z_{\text{TSA}} \geq z_{\text{TISA}} = v_1^i(c_i) + \sum_{k\neq i}\sum_{h=1}^{c_k} V^*_{1,k,h} \geq v_1(c) \qquad \forall i.$

Proof. For **Part (i)**, we parallel a similar proof of Zhang and Adelman (2009) which is by induction over time t. Fix a resource $i \in \{1,\dots,m\}$. Note that the boundary conditions give us $v^i_{\tau+1}(x) = 0$ for all x and $V^*_{\tau+1,k,h} = 0$ for all k,h. Thus for time period τ we have by conditions (39):

$$v^i_\tau(x_i) + \sum_{k\neq i}\sum_{h=1}^{x_k} V^*_{\tau,k,h} \geq \lambda \sum_{j\in S} P_j(S) f_j \qquad \forall x, S \subseteq N(x).$$

Since the inequality above holds for all $S \subseteq N(x)$, it holds in particular for the offer set $\tilde{S}$ that maximizes the right hand side. For $\tilde{S}$, the right hand side is then equal to $v_\tau(x)$.

Next, assume the inequality holds for $t+1$. Then conditions (39) together with the induction assumption yield

$$\begin{aligned} v^i_t(x_i) + \sum_{k\neq i}\sum_{h=1}^{x_k} V^*_{t,k,h} &\geq \lambda \sum_{j\in S} P_j(S)\big[f_j + v^i_{t+1}(x_i - a_{ij}) + \sum_{k\neq i}\sum_{h=1}^{x_k - a_{kj}} V^*_{t+1,k,h}\big] \\ &\quad + (v^i_{t+1}(x_i) + \sum_{k\neq i}\sum_{h=1}^{x_k} V^*_{t+1,k,h})(1 - \lambda \sum_{j\in S} P_j(S)) \\ &\geq \lambda \sum_{j\in S} P_j(S)(f_j + v_{t+1}(x - A^j) - v_{t+1}(x)) + v_{t+1}(x) \qquad \forall x, S \subseteq N(x). \end{aligned}$$

Comparison with the dynamic programming formulation (2) yields the desired result.

Part (ii): The first inequality has been shown by Zhang and Adelman (2009), the second by Meissner and Strauss (2009). The last one follows from part (i) of this proposition. For the equality we first show that the optimal solution V^* to the dual of (**TISA**), referred to as (**D**), yields a feasible solution for ($\mathbf{LP_i}$) for all i with the same objective value. Setting $v^i_t(x_i) := \theta_t + \sum_{h=1}^{x_i} V^*_{t,i,h}$ for all i, x_i, t, we obtain such a solution where feasibility follows from feasibility of V^*. To prove equality it is sufficient to show for any i the existence of a feasible solution to (**D**) which has the same objective as an optimal solution to ($\mathbf{LP_i}$). For any i, suppose $v^i_t(h)$ is an optimal solution to ($\mathbf{LP_i}$). Define $\theta_t := 0$ for all t, $V_{t,k,h} := V^*_{t,k,h}$ for all t, $k \neq i$, h and $V_{t,i,h} := v^i_t(h) - v^i_t(h-1)$ for all t, h. This solution is feasible to (**D**) due to feasibility of $v_t(\cdot)$ to ($\mathbf{LP_i}$) and has the same objective. □

The systems ($\mathbf{LP_i}$) were introduced to obtain these upper bound results; however, they do not provide a very effective way of actually solving the problem because the problems ($\mathbf{LP_i}$) have both many constraints as well as many decision variables. Instead, we use a dynamic programming decomposition approach:

$$\begin{aligned} \hat{v}^i_t(x_i) = \max_{x_{-i}, S\subseteq N(x)} \Bigg\{ &\sum_{j\in S} \lambda P_j(S)\Bigg(f_j - (\hat{v}^i_{t+1}(x_i) + \sum_{k\neq i}\sum_{h=1}^{x_k} V^*_{t+1,k,h} - \hat{v}^i_{t+1}(x_i - a_{ij}) \\ &- \sum_{k\neq i}\sum_{h=1}^{x_k - a_{kj}} V^*_{t+1,k,h})\Bigg) + \sum_{k\neq i}\sum_{h=1}^{x_k}(V^*_{t+1,k,h} - V^*_{t,k,h})\Bigg\} + \hat{v}^i_{t+1}(x_i), \end{aligned} \tag{40}$$

and $\hat{v}^i_{\tau+1}(x_i) = 0$ for all i, x_i. The maximization (40) has very similar structure like the column pricing problem (22) for TISA and therefore can be solved by a linear mixed integer program or heuristic in a similar manner.

Proposition 3 $\quad v^i_1(c_i) = \hat{v}^i_1(c_i) \qquad \forall i$

Proof. Fix a resource $i \in \{1,\dots,m\}$. By induction, we first show $v_t^i(x_i) \ge \hat{v}_t^i(x_i)$. The boundary conditions give $v_{\tau+1}^i(x_i) = \hat{v}_{\tau+1}^i(x_i) = 0$ for all x_i, so the equality holds in particular for c_i.

Assume now that the inequality holds for $t+1$. Then constraints of $(\mathbf{LP_i})$ give:

$$v_t^i(x_i) + \sum_{k\neq i}\sum_{h=1}^{x_k} V_{t,k,h}^* \ge \lambda \sum_{j\in S} P_j(S)\left(f_j + v_{t+1}^i(x_i - a_{ij}) + \sum_{k\neq i}\sum_{h=1}^{x_k - a_{kj}} V_{t+1,k,h}^* \right) + (v_{t+1}^i(x_i) + \sum_{k\neq i}\sum_{h=1}^{x_k} V_{t+1,k,h}^*)(1-\lambda\sum_{j\in S} P_j(S)) \qquad \forall x, S\subseteq N(x).$$

Since this holds for all $x, S\subseteq N(x)$, together with the induction assumption we have $v_t^i(x_i) \ge \hat{v}_t^i(x_i)$ as postulated.
In order to show $v_t^i(x_i) \le \hat{v}_t^i(x_i)$, note that the DP recursion (40) is such that $\hat{v}_t^i(\cdot)$ is feasible to the constraints of $(\mathbf{LP_i})$, and thus the optimal solution $v_t^i(\cdot)$ to the minimization problem $(\mathbf{LP_i})$ satisfies $v_t^i(x_i) \le \hat{v}_t^i(x_i)$ for all t, x_i.

□

After computing $\hat{v}_t^i(\cdot)$ for all resources i we approximate the network value function by

$$v_t(x) \approx \sum_i \hat{v}_t^i(x_i).$$

Plugging this approximation into the original dynamic programming formulation (2) yields the the maximization that we would need to solve in a given time period for a given state x:

$$\max_{S\subseteq N(x)} \lambda \sum_{j\in S} P_j(S)\left(f_j - \sum_i (\hat{v}_{t+1}^i(x_i) - \hat{v}_{t+1}^i(x_i - a_{ij}))\right).$$

Again, this problem can be solved with the mixed integer linear program or heuristic as elaborated above. We do not report numerical results for the DP decomposition for TISA since it requires to solve the associated LP close to optimality—which is expensive given the tailing off behavior of column generation—and since the policies performed similar or worse like the direct policies. This behavior is intuitive because DP decomposition essentially introduces time- and inventory-dependence. However, the dual values arising in TISA have already this property and, furthermore, the underlying LP can take network effects better into account.

5.3. Policies based on CDLP

For CDLP, we only consider the decomposition approach as used by Miranda Bront et al. (2009) because their results show a better performance for this approach as opposed to using dual values directly, even if re-solving is employed. Solving the CDLP provides us with the optimal dual values of the capacity constraints, denoted by π^*. The network is again being decomposed by the resource and the value function is approximated by $v_t(x) \approx v_t^i(x_i) + \sum_{k\neq i}\pi_k^* x_k$, where $v_t^i(x_i)$ is computed by the single resource dynamic program

$$v_t^i(x_i) = \max_{S\subseteq N} \sum_{j\in S} \lambda P_j(S)\left[f_j - (v_{t+1}^i(x_i) - v_{t+1}^i(x_i - 1) - \pi_i^*)\mathbf{1}_{\{i\in A_j\}} - \sum_{k\in A_j} \pi_k^* \right] + v_{t+1}^i(x_i),$$

with $v_{\tau+1}^i(x_i) = 0$ for all x_i and $v_t^i(0) = 0$ for all t on the boundary. Miranda Bront et al. (2009) proposed a linear mixed integer program to solve the dynamic programming subproblem and, alternatively, a heuristic.

Another approach would be full enumeration which is clearly not possible for large number of products. However, Talluri and van Ryzin (2004a) showed that any optimal solution to

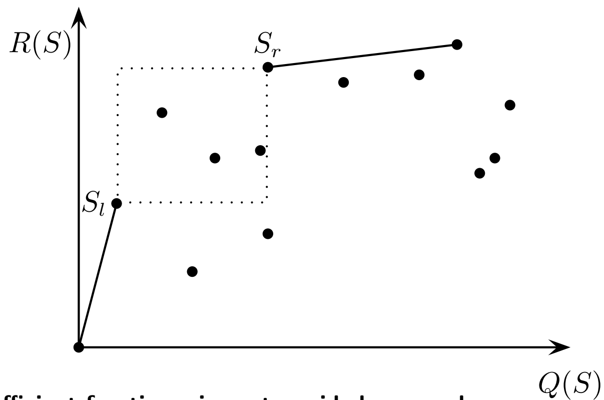

Figure 1 **Identification of efficient frontier using a two-sided approach.**

Algorithm 3 Two-sided marginal revenue procedure

Set $S_l \leftarrow \emptyset$, $S_r \leftarrow \arg\max_{S\in\mathcal{P}(N)} R(S)$
$E \leftarrow \{S_l, S_r\}$ (efficient sets)
loop
    $S_l \leftarrow \arg\max_{S\in\mathcal{S}(S_l,S_r)} \frac{R(S)-R(S_l)}{Q(S)-Q(S_l)}$
    if $S_l = S_r$ **then**
        break
    end if
    $E \leftarrow E \cup \{S_l\}$
    $S_r \leftarrow \arg\min_{S\in\mathcal{S}(S_l,S_r)} \frac{R(S_r)-R(S)}{Q(S_r)-Q(S)}$
    if $S_r = S_l$ **then**
        break
    end if
    $E \leftarrow E \cup \{S_r\}$
end loop

this type of single resource dynamic program is efficient in the following sense: Let us denote the expected displacement-adjusted revenue by $R(S) := \sum_{j\in S} \lambda P_j(S)(f_j - \sum_{k\in A_j, k\neq i} \pi_k^*)$, and the expected resource consumption by $Q(S) := \sum_{j\in S} \lambda P_j(S) a_{ij}$. Then, we define a set T to be called *inefficient* if there exists a set of convex weights $\alpha(S)$, $\sum_S \alpha(S) = 1$, $\alpha(S) \geq 0$ for all S, such that $R(T) < \sum_S \alpha(S) R(S)$ and $Q(T) \geq \sum_S \alpha(S) Q(S)$. Otherwise, T is called *efficient*. A useful characterization of efficient sets is shown in the following proposition:

Proposition 4 (Talluri and van Ryzin (2004a)) *A set T is* efficient *if and only if it maximizes* $\max_S \{R(S) - Q(S)\mu\}$ *for some value* $\mu \geq 0$.

Therefore, we can restrict the search for an optimal solution to only efficient sets, given that there is some efficient means to identify them. However, the problem of identifying efficient sets is computationally intensive. We propose a modification of the so-called "largest marginal revenue procedure" presented in Talluri and van Ryzin (2004a) that identifies efficient sets for general choice models. This method is based on the observation that once an efficient set S_k is given, the next one can be identified as the set with highest marginal revenue ratio $(R(S) - R(S_k))/(Q(S) - Q(S_k))$ among all sets S with $R(S) \geq R(S_k)$ and $Q(S) \geq Q(S_k)$. The empty set is always efficient and thus provides a starting point for this approach.

Our essential idea is to recognize that the set S_r with the highest expected revenue $R(S_r)$ will always be efficient (set $\mu = 0$ in Proposition 4), as well as the empty set, denoted by S_l. We can restrict the search to sets in

$$\mathcal{S}(S_l, S_r) := \{S \in \mathcal{P}(N) : R(S) \in [R(S_l), R(S_r)], Q(S) \in [Q(S_l), Q(S_r)]\},$$

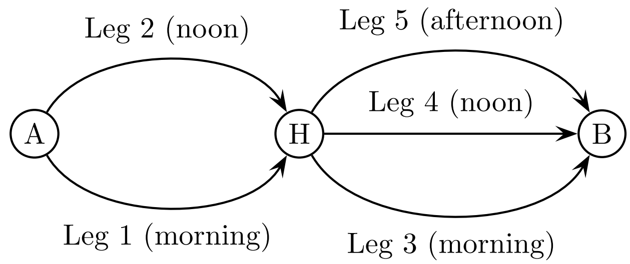

Figure 2 **Hub & Spoke network example.**

where $\mathcal{P}(N)$ denotes the power set of N, and identify the next efficient set by maximizing the slope of a line through $(Q(S_l), R(S_l))$ and a point out of $\mathcal{S}(S_l, S_r)$, or by similar minimization if coming from the right-hand side. Algorithm 3 implements this idea, where we alternate between coming from the left and right side to find the next facet of the efficient frontier. Figure 1 illustrates the process. The advantage of the two sided approach is to have further bounds that can be used to restrict the search space $\mathcal{S}(S_l, S_r)$ better. Of course, the number of subsets becomes very quickly intractable, therefore one might consider heuristical approach such as identifying efficient sets not over the whole power set $\mathcal{P}(\mathcal{N})$ but rather, for example, over sets used in former booking processes enriched with some randomly sampled ones. The strength of the concept of efficient sets is that for some choice models these sets correspond to those in a simple nested fare order, however, in the context of overlapping consideration sets this does unfortunately not hold any more in general as exemplified by Miranda Bront et al. (2009). Still, it might prove advantageous since we need to solve the dynamic programming subproblem over all time periods while the efficient sets always remain the same, only the cost value changes. Therefore, once the efficient sets are identified, we could solve all these subproblems quickly.

When all resource-level value functions $v_t^i(x_i)$ have been computed, we solve the following problem at each time period for the given state x:

$$\max_{S \subseteq N(x)} \sum_{j \in S} \lambda P_j(S) \left[f_j - \sum_{i=1}^{m} (v_{t+1}^i(x_i) - v_{t+1}^i(x_i - 1)) a_{ij} \right]. \tag{41}$$

6. Numerical Experiments

In this section, we investigate numerically how the performance of the time-sensitive approach (TSA) and time- and inventory-sensitive approach (TISA) compare with the linear programming approach of Miranda Bront et al. (2009), in particular for different degrees of market segmentation weakness, where performance refers to upper bound quality, simulated revenue results and time consumption. The purpose is to obtain insights into which method might be the most beneficial in situations where products fences are broken. Furthermore, we investigate the performance of the proposed heuristics in terms of average time reduction and achieved solution quality.

6.1. Experiment design

A hub and spoke network example is depicted in Figure 2 and consists of two parallel flights from location A to H, and further three parallel flights from H to B. The initial capacity vector is defined by $c := [3, 2, 2, 2, 2]^T$. On each itinerary we again have two fare classes, High and Low. The network example allows us to have short-haul and long-haul products compete for the same resources, in addition to customers who can decide whether to fly early or late on parallel flights and, depending on the level of overlap that we allow in the model setup, whether to buy up or down. The products are defined in Table 1.

We wish to explore the impact of weak segmentation on the performance of the various policies, hence we consider three scenarios of consideration sets in Table 2 corresponding to high, medium and low overlap between the consideration sets. While we have in all cases the same number of segments, we allow the price insensitive segments to consider a from scenario to scenario growing number of lower fares, reflecting an increasingly weak segmented market.

Furthermore, we vary the expected capacity tightness ρ by scaling the vector of arrival rates λ. This is measured by

$$\rho := \frac{\sum_{t=1}^{\tau} \sum_{l=1}^{L} \lambda_l \sum_{j \in S^*} \sum_{i=1}^{m} a_{ij} P_j(S^*)}{\sum_{i=1}^{m} c_i},$$

where the set S^* is the optimal offer set given ample capacity, that is

$$S^* := \arg\max_{S \subseteq N} \sum_{j \in S} f_j P_j(S).$$

Again we consider three scenarios, namely high, medium and low ρ. We are interested in network configurations with $\rho \geq 1$, because otherwise the problem has on average non-binding capacity constraints and therefore the optimal solution is trivial, namely to offer the set S^*.

Finally, we scale capacity vector and time horizon up to obtain insights into policy performance for larger systems but still solve small problems as compared to the practice since we intend to demonstrate the maximum potential of the proposed policies which, however, implies to carry out expensive computations for TISA.

In order to find the dual values needed for the policies and the dynamic programming decomposition, the CDLP is always solved until optimality, where we employ the heuristic proposed by Miranda Bront et al. (2009) to solve the column generation subproblem. If the heuristic cannot find any further column with positive reduced profit, the mixed integer formulation is used. Similarly we use Algorithm 1 for the column generation subproblem of TSA and Algorithm 2 for TISA. As a stopping criterion for the approaches TSA and TISA we compute the sum over all time periods of respectively maximal reduced profits (so to speak the estimated potential of improvement), and stop generating columns if this value is within 1% and 5%, respectively, of the current optimal objective value plus the improvement potential. For all approaches, we retain all columns that have been generated in the master problem.

All simulations are run until the relative percentage error of the mean is less than 0.5% with 95% confidence, which corresponds to sample sizes of about 4000 to 6000 depending on the network configuration. The empirical average network load factor is measured as the mean of the sum of sold capacity units divided by the network capacity. This indicates how restrictive a policy is on average, though a higher load factor in general does not need to imply higher revenues, of course. In every simulated time step, a new offer set needs to be computed. All simulations use the mixed integer programming approach to solve the problem (34) for approximation of the value function corresponding to the respective policy used.

The performance of Algorithms 1 and 2 is measured in several ways: Firstly, we measure the total CPU time that it takes to solve the corresponding master problem with using the heuristic for the column generation subproblem, and only resorting to the mixed integer formulation once the heuristic does not find a column with positive reduced profit any more. This time is compared to the one that it takes to solve the master problem using only the mixed integer formulation for the subproblems. Secondly, we measure the CPU time for each individual run of the heuristic and compare this to the time that the mixed integer program takes to solve the same subproblem. We compute the ratio of these times and accept the column found by the heuristic as long as it has positive reduced cost. Once the solution to the master problem has been found, we compute the average of all the time ratios of heuristic versus mixed integer problem. In order to quantify the

Table 1 Product definitions for HS.

Product	Resources	OD	Class	Fare
1	1	A → H	L	400
2	1	”	H	800
3	2	”	L	300
4	2	”	H	600
5	3	H → B	L	400
6	3	”	H	800
7	4	”	L	300
8	4	”	H	600
9	5	”	L	400
10	5	”	H	800
11	1,4	A → B	L	500
12	1,4	”	H	1000
13	1,5	”	L	450
14	1,5	”	H	900
15	2,5	”	L	400
16	2,5	”	H	800

“Resources” indicates the resources which the respective product utilizes.

Table 2 Segments and consideration sets for HS.

#	Segment	Consideration set	Pref. vector	λ_l (%)	v_{l0}
1	A → H, price insensitive, early preferred	{{{2,4},1},3}	[[[10,5],10],7]	7	1
2	A → H, price insensitive, late preferred	{{{2,4},3},1}	[[[5,10],10],7]	5	1
3	A → H, price sensitive	{{{1,3}}}	[[[10,8]]]	15	5
4	H → B, price insensitive, early preferred	{{{6,8,10}5}7,9}	[[[10,5,1],10],7,3]	7	1
5	H → B, price insensitive, late preferred	{{{6,8,10}9}7,5}	[[[1,5,10],10],7,3]	5	1
6	H → B, price sensitive	{{{5,7,9}}}	[[[5,10,5]]]	15	5
7	A → B, price insensitive, early preferred	{{{12,14},11},13}	[[[10,5],10],7]	7	1
8	A → B, price insensitive, late preferred	{{{16},15},11,12}	[[[10],10],7,5]	5	1
9	A → B, price sensitive	{{{11,13,15}}}	[[[5,8,10]]]	15	5

The column “Consideration Set” lists the products that any segment considers for three different scenarios with different degree of overlap (low, medium, high). For example, {{{2,4},1},3} means $C_1 = \{2,4\}$ in the low, $C_1 = \{2,4,1\}$ in the medium and $C_1 = \{2,4,1,3\}$ in the high overlap case.

level of quality achieved in the solution of the heuristic, we compute in a similar fashion the ratio of reduced profit of the column found by the heuristic to the maximum reduced profit determined by the mixed integer program. Again, we report the average over all column generation iterations.

Finally, we measure the proportions on the total CPU time for solving the master problem that are spent on running the heuristic, running the mixed integer problem in the case that the heuristic failed, and for solving the master problems. This identifies the tasks where most time was spent on and where future work would be most effective in speeding up the solution.

The tested policies are the following:

- D-CDLP: The decomposition policy (41) based on CDLP; it will serve us as a benchmark.
- TSA: The policy (35) based on directly using the dual values of (**TSA**) to approximate opportunity cost.
- D-TSA: The decomposition policy (37) based on the affine approximation.
- TISA: The policy (38) based on directly using the dual values of (**TISA**).

6.2. Upper bounds and policy performance

We report upper bounds on the optimal expected revenue $v_1(c)$ in Table 3. It is known that $z_{\text{CDLP}} \geq z_{\text{TSA}} \geq z_{\text{TISA}}$ and that the decomposition bound of CDLP and TSA is tighter than z_{CDLP} and z_{TSA}, respectively. Comparison of the bounds for TSA and D-CDLP indicate that neither bound dominates the other. In contrast, D-TSA seems to dominate D-CDLP, and in turn TISA seems to dominate D-TSA. Both observations intuitively make sense since the input of the DP decomposition contains increasingly more information. TISA provides the strongest bound with improvements between 2% and 10% over D-CDLP. The improvements in the bound tend to become smaller as capacity and time horizon scale up, which can be attributed to the asymptotic optimality of the CDLP. In most cases, the differences are largest for the medium overlap scenario and lie there in the range of 0–2% for TSA, 0–5% for D-TSA and 3–7% for TISA. With respect to changes in the capacity tightness the bounds do not vary much, though there seems to be slightly tighter bounds for TSA, D-TSA and TISA for medium capacity tightness.

As for policy performance, in Table 4 we report the sample means of simulations for the different discussed policies. TSA is in 4 out of 27 cases weaker than D-CDLP, but only in two cases by more than one percent. D-TSA improves D-CDLP in all cases, however, surprisingly it does not always improve policy performance compared to TSA. The reason for this behaviour is most likely to be found in the fact that we solved (**TSA**) only approximately, so that the input for the decomposition procedure is somewhat erratic. It is an interesting observation that DP decomposition is not very robust with respect to disturbed input values and can actually result in poorer policy performance if the input is not the exact dual solution of the underlying linear program.

TISA surpasses the CDLP-based decomposition policy D-CDLP in every scenario by 2–23%. For better illustration of TISA's results, we plotted the average revenue improvements of Table 4 in Figure 3. With regard to tightness of capacity, best performance is shown for the medium case of $\alpha = 1$ which coincides with intuition since revenue management decisions will be most critical if there is neither so little demand that capacity constraints can be ignored, nor so much demand that high value demand can fill up the plane itself. Though the scalings of λ do not yield demand scenarios that extreme, yet the tendency is reflected in the results. The improvements become smaller as capacity and time horizon are scaled up, but are still between 4-6% in the medium capacity tightness case. An important observation is related to performance depending on the degree of overlap. Of course, average revenues decrease as overlap increases since formerly price-insensitive segments consider more buy-down opportunities. More interesting is the relative performance of the approaches under investigation: In particular for TISA the largest improvements usually occur under high overlap, most pronounced under medium capacity tightness. This supports our claim that TISA is particularly advantageous in weakly segmented markets due to its refined opportunity cost estimates. For practical application, however, we need to give up some accuracy in favor of reduced run time. To that end, note that results for policies based on general aggregation of inventory can be expected to yield results between the benchmarks provided by TSA and TISA.

6.3. Performance of the heuristics

We investigate the performance of Algorithm 1 and 2 in terms of quality and time consumption. For the test problems we use medium capacity tightness and high overlap where we observed the best revenue improvements.

In Table 5 we report the number of columns that were generated to solve (**CDLP**) to optimality and (**TSA**)/(**TISA**) both with the 5% stopping criterion where we always first used the heuristic to find a new column and MIP only if the heuristic could not find one with positive reduced profit. The number of variables and constraints in (**CDLP**) does not depend on time or capacity, and accordingly the number of generated columns stays constant across the considered test problems. As expected, for TISA this number increases quickly underlining the need for aggregation techniques for large problems. For both TSA and TISA almost all columns are found by the heuristic.

The quality of the heuristics was assessed by running both MIP and heuristic for a given subproblem and subsequently recording the ratio of reduced profit found by the heuristic divided by reduced profit of the column identified by MIP. The average of these ratios over all subproblems

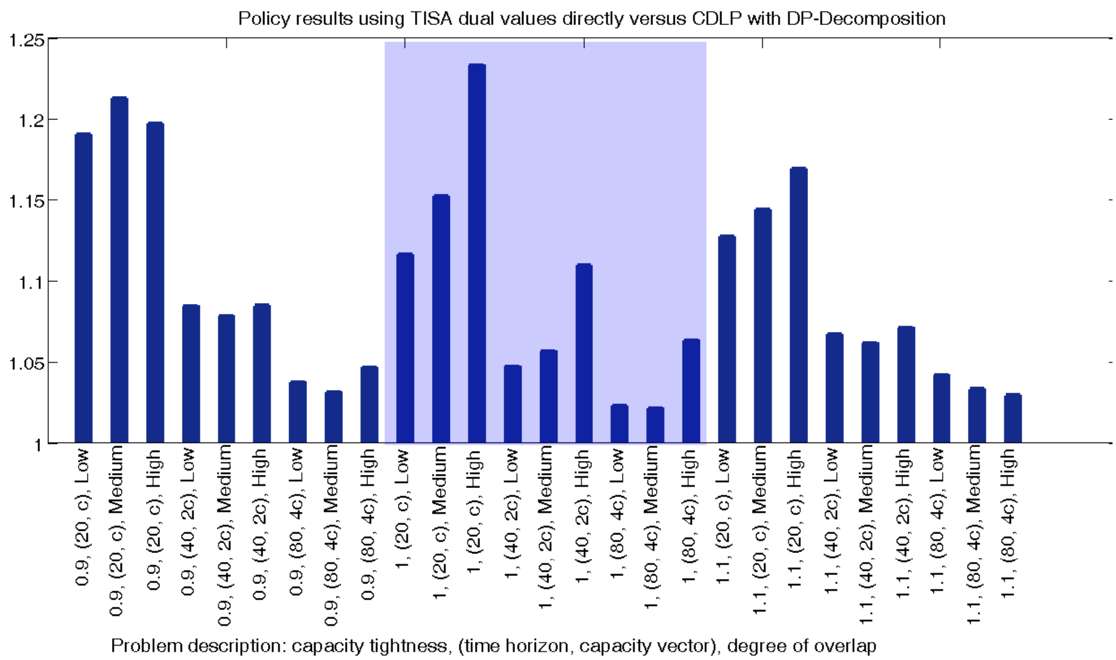

Figure 3 **Policy performance.**

Table 3 **Upper bounds**

α	(τ, c)	Overlap	CDLP	D-CDLP	TSA	TISA	D-TSA	D-CDLP/TSA	D-CDLP/D-TSA	D-CDLP/TISA
0.9	(20, 3)	Low	5927	5458	5419	5176	5275	1.01	1.03	1.05
0.9	(20, 3)	Medium	5629	5227	5162	4945	5034	1.01	1.04	1.06
0.9	(20, 3)	High	5498	5078	5104	4912	4943	1.00	1.03	1.03
0.9	(40, 6)	Low	11854	11336	11340	10907	11176	1.00	1.01	1.04
0.9	(40, 6)	Medium	11259	10850	10822	10416	10640	1.00	1.02	1.04
0.9	(40, 6)	High	10996	10545	10610	10281	10439	0.99	1.01	1.03
0.9	(80, 12)	Low	23709	23143	23241	22505	23114	1.00	1.00	1.03
0.9	(80, 12)	Medium	22518	22112	22069	21497	22006	1.00	1.00	1.03
0.9	(80, 12)	High	21992	21518	21566	21139	21471	1.00	1.00	1.02
1.0	(20, 3)	Low	6175	5768	5660	5453	5543	1.02	1.04	1.06
1.0	(20, 3)	Medium	6008	5576	5453	5219	5290	1.02	1.05	1.07
1.0	(20, 3)	High	5882	5428	5427	5224	5277	1.00	1.03	1.04
1.0	(40, 6)	Low	12350	11889	11816	11450	11688	1.01	1.02	1.04
1.0	(40, 6)	Medium	12016	11522	11451	10999	11269	1.01	1.02	1.05
1.0	(40, 6)	High	11764	11271	11333	10945	11118	0.99	1.01	1.03
1.0	(80, 12)	Low	24700	24190	24164	23603	24125	1.00	1.00	1.02
1.0	(80, 12)	Medium	24031	23478	23414	22784	23349	1.00	1.01	1.03
1.0	(80, 12)	High	23527	23005	23074	22569	22895	1.00	1.00	1.02
1.1	(20, 3)	Low	6372	5971	5875	5667	5767	1.02	1.04	1.05
1.1	(20, 3)	Medium	6273	5833	5713	5493	5583	1.02	1.04	1.06
1.1	(20, 3)	High	6216	5724	5726	5216	5601	1.00	1.02	1.10
1.1	(40, 6)	Low	12744	12289	12245	11881	12096	1.00	1.02	1.03
1.1	(40, 6)	Medium	12546	12037	11978	11529	11795	1.00	1.02	1.04
1.1	(40, 6)	High	12432	11868	11944	11575	11727	0.99	1.01	1.03
1.1	(80, 12)	Low	25489	24983	24975	24421	24920	1.00	1.00	1.02
1.1	(80, 12)	Medium	25092	24497	24507	23813	24281	1.00	1.01	1.03
1.1	(80, 12)	High	24864	24224	24341	23823	24215	1.00	1.00	1.02

Tightness of capacities is controlled by scaling arrival rate λ with α.

Table 4 **Policy performance**

α	(τ, c)	Overlap	D-CDLP	TSA	TISA	D-TSA	TSA/D-CDLP	D-TSA/D-CDLP	TISA/D-CDLP
0.9	(20, c)	Low	4077	4797	4854	4520	1.18	1.11	1.19
0.9	(20, c)	Medium	3839	4571	4654	4291	1.19	1.12	1.21
0.9	(20, c)	High	3839	4549	4595	4188	1.19	1.09	1.20
0.9	(40, 2c)	Low	9625	10239	10438	9992	1.06	1.04	1.08
0.9	(40, 2c)	Medium	9216	9626	9941	9516	1.04	1.03	1.08
0.9	(40, 2c)	High	9003	9328	9767	9179	1.04	1.02	1.08
0.9	(80, 4c)	Low	21073	21425	21867	21438	1.02	1.02	1.04
0.9	(80, 4c)	Medium	20156	19190	20793	20324	0.95	1.01	1.03
0.9	(80, 4c)	High	19419	19365	20326	19628	1.00	1.01	1.05
1.0	(20, c)	Low	4594	5043	5129	4793	1.10	1.04	1.12
1.0	(20, c)	Medium	4291	4827	4945	4577	1.12	1.07	1.15
1.0	(20, c)	High	3986	4839	4916	4441	1.21	1.11	1.23
1.0	(40, 2c)	Low	10414	10536	10910	10576	1.01	1.02	1.05
1.0	(40, 2c)	Medium	9987	10092	10557	10159	1.01	1.02	1.06
1.0	(40, 2c)	High	9452	10101	10490	9971	1.07	1.05	1.11
1.0	(80, 4c)	Low	22385	21802	22904	22537	0.97	1.01	1.02
1.0	(80, 4c)	Medium	21544	21257	22013	21710	0.99	1.01	1.02
1.0	(80, 4c)	High	20495	21338	21801	21207	1.04	1.03	1.06
1.1	(20, c)	Low	4777	5186	5386	4960	1.09	1.04	1.13
1.1	(20, c)	Medium	4536	5098	5189	4743	1.12	1.05	1.14
1.1	(20, c)	High	4390	5051	5134	4708	1.15	1.07	1.17
1.1	(40, 2c)	Low	10707	10811	11429	10980	1.01	1.03	1.07
1.1	(40, 2c)	Medium	10412	10636	11056	10615	1.02	1.02	1.06
1.1	(40, 2c)	High	10287	10780	11020	10497	1.05	1.02	1.07
1.1	(80, 4c)	Low	22828	23021	23789	23351	1.01	1.02	1.04
1.1	(80, 4c)	Medium	22367	22490	23116	22633	1.01	1.01	1.03
1.1	(80, 4c)	High	22325	22043	22986	22501	0.99	1.01	1.03

All simulations were run until the relative percentage error was less than 0.5% with 95% confidence. Tightness of capacity is controlled via scaling arrival rate λ with α.

until the master problem was solved is reported in Table 6 and demonstrate how close we are on average to the optimal solution of the subproblems.

Time savings are considerable in particular for TISA—in Table 7 average time ratios are reported in a similar manner as for the quality ratios. Note that the mean for CDLP and TSA does not change since the subproblem is defined for a fixed time step, so there is no dependence on the time horizon. The drop to 34% for CDLP is most likely because the ratio consists of small numbers and therefore is prone to disturbances. TISA exhibits a strong decrease to only 5% of the required time by the MIP since the latter increases with the capacity vector. Although the time required to identify a new column has been decreased, we potentially might offset these gains by having considerably more columns to identify. In order to check whether this is the case we compared in Table 8 and 9 the total CPU time that it took to solve the respective master problems. Furthermore, we split up the total time into the required time for solving the master problems and subproblem via MIP and heuristic respectively. The overall speed-up is about 50% for TSA over all test problems. For TISA, the run time is also substantially reduced but not to the amount that the reduction per subproblem would lead us to expect. The reason for this is that our column generation algorithm never deletes any column, hence the master problem requires increasingly more time. While the approach without heuristics required most of the run time to solve the subproblems, for the last problem with heuristics more than 90% of total time are required to solve the master problem. In this light, implementing more sophisticated column generation procedures that control the pool of columns in conjunction with our proposed heuristics should yield good results and constitutes an interesting area of future research.

Table 5 **Number of columns found by heuristic and MIP, respectively**

			CDLP		TSA		TISA	
α	(τ, c)	Overlap	Heu	MIP	Heu	MIP	Heu	MIP
1.0	(20, c)	High	18	0	567	9	863	25
1.0	(40, 2c)	High	18	0	996	4	2386	27
1.0	(80, 4c)	High	18	0	1917	3	9599	103

Table 6 **Average quality ratio over subproblems**

			CDLP		TSA		TISA	
α	(τ, c)	Overlap	Mean	Stddev	Mean	Stddev	Mean	Stddev
1.0	(20, c)	High	0.97	0.10	0.94	0.18	0.90	0.22
1.0	(40, 2c)	High	0.97	0.10	0.98	0.10	0.93	0.18
1.0	(80, 4c)	High	0.97	0.10	0.99	0.06	0.93	0.18

Quality is measured as the objective of the heuristic solution divided by the optimal objective. Sample created by running both methods for every column generation subproblem. Column *Stddev* reports standard deviation.

Table 7 **Average time ratio over subproblems**

			CDLP		TSA		TISA	
α	(τ, c)	Overlap	Mean	Stddev	Mean	Stddev	Mean	Stddev
1.0	(20, c)	High	0.46	0.32	0.40	0.29	0.27	0.25
1.0	(40, 2c)	High	0.45	0.20	0.43	0.26	0.11	0.11
1.0	(80, 4c)	High	0.34	0.23	0.43	0.25	0.05	0.05

Time ratio is CPU time of running the heuristic divided by CPU time required for the mixed integer program. Column *Stddev* denotes standard deviation.

Table 8 **Total run times to solve CDLP, TSA and TISA without heuristics**

	CDLP			TSA			TISA		
(τ, c)	Total	Master	MIP	Total	Master	MIP	Total	Master	MIP
(20, c)	1s	19%	81%	21s	3%	97%	9 min	1%	99%
(40, 2c)	1s	20%	80%	39s	5%	95%	118 min	4%	96%
(80, 4c)	1s	15%	85%	78s	8%	92%	1735 min	20%	80%

Master reports the CPU time overall used to solve the growing master problems relative to the *Total* CPU time needed until the stopping criterion of column generation procedure is satisfied, *MIP* likewise for the overall time needed to solve the column generation subproblems. Medium capacity tightness ($\alpha = 1.0$), high overlap scenario.

Table 9 **Run times to solve CDLP, TSA and TISA using heuristics**

	CDLP-Heuristics				TSA-Heuristics				TISA-Heuristics			
(τ, c)	Total	Master	Heu	MIP	Total	Master	Heu	MIP	Total	Master	Heu	MIP
(20,c)	1s	34%	58%	8%	12s	8%	83%	9%	2 min	5%	78%	16%
(40,2c)	1s	26%	65%	9%	20s	11%	88%	1%	9 min	37%	49%	14%
(80,4c)	1s	27%	61%	12%	42s	20%	80%	0%	626 min	91%	4%	5%

Master reports the CPU time overall used to solve the growing master problems relative to the *Total* CPU time needed until the stopping criterion of column generation procedure is satisfied, *Heu* likewise for the overall time needed to solve the column generation subproblems with heuristics and *MIP* for the case that the heuristic failed. Medium capacity tightness ($\alpha = 1.0$), high overlap scenario.

7. Conclusion

In this work we extend the choice-based network revenue management approaches of Zhang and Adelman (2009) and Meissner and Strauss (2009) to the multinomial logit (MNL) choice model with overlapping consideration sets. The extension of the former work is referred to as TSA for time-sensitive approach, the latter as TISA for time- and inventory-sensitive approach. We use the MNL choice model to describe weakly segmented markets where product fences cannot keep customer segments fully separated. We solve large-scale linear programs with column generation to obtain opportunity cost estimates that can be used subsequently to construct policies. In the presented extensions, a major issue is the question of how to solve the column generation subproblems efficiently since they are NP-hard. We propose polynomial-time heuristics and numerically analyze their performance with respect to quality and time consumption. Based on the solution of the linear program we construct policies either by using dynamic programming (DP) decomposition or by using the solution of the linear program directly to estimate opportunity cost of product sales. DP decomposition was found to be unable to further improve the disaggregated approach TISA. Indeed, we prove that the upper bound resulting from DP decomposition is in fact the same as the one resulting from solving the linear program for TISA directly which hints at no additional revenue performance being achievable by this method. In our experiments, DP decomposition turned out to be not very robust facing somewhat inaccurate input in the form of the approximate linear program solution. This is a reason why the policies directly based on TSA performed in many cases even better than the ones based on DP decomposition using TSA.

For practical implementation, the full-blown approach is computationally too expensive, however, using inventory aggregation as described in Meissner and Strauss (2009) we can trade off run time against accuracy. Our results for TSA and TISA represent the two extremes for this aggregation method, and accordingly revenue performance of any aggregated approach can be expected to lie between these two benchmarks. The work gives therefore insight into how much revenue improvement is possible by using any aggregated approach and, furthermore, it also provides heuristics that can likewise be used to speed up existing methods of Zhang and Adelman (2009) and Meissner and Strauss (2009) in the disjoint consideration set context. We find that TSA and TISA exhibits good revenue results as compared to the current benchmark method of Miranda Bront et al. (2009) based on dynamic programming decomposition, and that the improvements become the more pronounced the more the consideration sets overlap, that is in particular for the case of weak segmentation. An intuitive reason for this observation is that the refined opportunity cost estimates of TISA have an increased impact on revenue performance the more segments overlap since the decisions to be made become more difficult in this situation.

References

Adelman, D. 2007. Dynamic bid prices in revenue management. *Operations Research* **55** 647–661.

American Express Business Travel. 2008. Global Business Travel Forecast.

Chiang, W.-C., J. C. H. Chen, X. Xu. 2007. An overview of research revenue management: current issues and future research. *International Journal of Revenue Management* **1** 97–128.

Farias, V. F., B. Van Roy. 2007. An approximate dynamic programming approach to network revenue management. URL `http://web.mit.edu/\~{}vivekf/www/papers/ADP-rm-07-03.pdf`. Working paper.

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

Gallego, G., R. Phillips. 2004. Revenue management of flexible products. *Manufacturing & Service Operations Management* **6** 321–337.

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

Liu, Q., G. J. van Ryzin. 2008. On the choice-based linear programming model for network revenue management. *Manufacturing & Service Operations Management* **10** 288–311.

McGill, J., G.J. van Ryzin. 1999. Revenue management: Research overview and prospects. *Transportation Science* **33** 233–256.

Meissner, J., A.K. Strauss. 2009. Network revenue management with inventory-sensitive bid prices and customer choice. Working Paper, Lancaster University Management School.

Miranda Bront, J. J., I. Méndez-Díaz, G. Vulcano. 2009. A column generation algorithm for choice-based network revenue management. *Operations Research* **57** 769–784. URL `http://pages.stern.nyu.edu/\~{}gvulcano/ColGenChoiceRM-Rev2.pdf`.

Powell, W. 2007. *Approximate Dynamic Programming*. John Wiley & Sons, Hoboken, NJ.

Prokopyev, O., H. Huang, P. Pardalos. 2005. On complexity of unconstrained hyperbolic 0-1 programming problems. *Operations Research Letters* **33** 312–318.

Talluri, K. 2008. On bounds for network revenue management. Working Paper, Universitat Pompeu Fabra.

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

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

Topaloglu, H. 2009. Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research* **57** 637–649. URL `http://people.orie.cornell.edu/\~{}huseyin/publications/revenue\_man.pdf`. Forthcoming in *Operations Research*.

Verband Deutsches Reisemanagement e.V. 2008. VDR Business Travel Report Germany.

Zhang, D., D. Adelman. 2009. An approximate dynamic programming approach to network revenue management with customer choice. *Transportation Science* **43** 381–394. URL `http://faculty.chicagogsb.edu/daniel.adelman/research/network.pdf`.