Lancaster University Management School
Working Paper
2010/017

Network Revenue Management with Inventory-Sensitive Bid Prices and Customer Choice

Joern Meissner and Arne Karsten Strauss

The Department of Management Science
Lancaster University Management School
Lancaster LA1 4YX
UK

Network Revenue Management with Inventory-Sensitive Bid Prices and Customer Choice

Joern Meissner, Arne K. Strauss
Department of Management Science, Lancaster University Management School, Lancaster, LA1 4YX, UK
joe@meiss.com, a.strauss@lancaster.ac.uk

April 16, 2010

We develop an approximate dynamic programming approach to network revenue management models with customer choice that approximates the value function of the Markov decision process with a non-linear function which is separable across resource inventory levels. This approximation can exhibit significantly improved accuracy compared to currently available methods. It further allows for arbitrary aggregation of inventory units and thereby reduction of computational workload, yields upper bounds on the optimal expected revenue that are provably at least as tight as those obtained from previous approaches, and is asymptotically optimal under fluid scaling. Computational experiments for the multinomial logit choice model with distinct consideration sets show that policies derived from our approach outperform available alternatives, and we demonstrate how aggregation can be used to balance solution quality and runtime.

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

1. Introduction

A particular area of revenue management (RM) that currently receives much interest is the approximate solution of the RM network problem including models of customer choice behavior. Network problems arise in many applications such as hospitality or transportation where the managed products might require more than one resource, for example a hotel that sells rooms over several nights. While network models have been around for some time already, only in recent years researchers devoted themselves to advancing discrete choice models where the purchase decisions also depend on the offered product alternatives. The need for such models is heightened by the rise of low cost service providers since they cut many of the traditional restrictions meant to segment the market, leaving the customer with similar products whose essentially only distinguishing feature is the price. Even if there are still some restrictions, customers increasingly tend to ignore them in their purchase decision so that in some business applications demand can only be observed for the product with the lowest available price, as pointed out by Boyd and Kallesen (2004). Such a behavior is in stark contrast to the traditional independent demand setting where it is assumed that demand is associated with a product and does not depend on market conditions such as which other products the firm offers. Therefore it is crucial to incorporate customer choice models into RM; more on the advantages of customer choice in the RM context can be found in van Ryzin (2005) and, for a comprehensive treatment of RM, in Talluri and van Ryzin (2004b).

We base our investigations on the particularly interesting work of Zhang and Adelman (2009) who extend the previous independent demand RM model of Adelman (2007) to incorporate customer choice behavior. Their approach differs from others in that they use an affine function of the state vector to approximate the value function of the exact dynamic programming formulation with a linear program (LP) in a way such that it yields time-dependent estimates of the marginal capacity values. The optimal objective of this LP constitutes an upper bound on the exact optimal expected revenue which is tighter than those obtained by several other currently available methods. Since the LP possesses many variables, solving the problem by column generation is shown for the multinomial logit choice model (MNL) with disjoint consideration sets to reduce essentially to solving smaller mixed integer linear programs and is thus implementable in practice. They construct policies directly from the dual solution as well as through a dynamic programming decomposition scheme and show that both perform very well. The most important reason for the improved performance is that the LP naturally generates time-dependent marginal capacity value estimates which gives this approach a cutting edge compared to methods that generate static values.

However, intuitively these values should not only depend on time to departure (for the ease of presentation we will stick to airline terminology), but also on the inventory levels. This dependence on intermediate capacity levels of the resources is not captured by current approaches to network RM with choice behavior. In the independent demand setting, a suitable approximation function was recently proposed by Farias and Van Roy (2007). Instead of using constraint generation to deal with the many constraints of the arising linear program they propose using a constraint sampling procedure which is based on the work of de Farias and Van Roy (2003) and de Farias and Van Roy (2004). The same approximation was independently proposed by Talluri (2008) under the name of *strong affine relaxation* and shown to provide tighter upper bounds on the optimal expected revenue than other available methods for the no-choice setting. Also Topaloglu (2009) recently focussed on time- *and* capacity-dependent bid prices: He proposed a network RM approach based on Lagrangian relaxation, but again without inclusion of choice behavior.

Our key contributions are the following:

- We propose a new linear programming approach to approximate dynamic programming that approximates the value function with a nonlinear function of the state vector which is separable over arbitrarily chosen ranges of resource inventory levels. As a special case, we can choose this approximation to be separable over each possible inventory level, which then corresponds to the approximation proposed by Farias and Van Roy (2007), but, in contrast to their approach, our model also accounts for customer choice behavior.

- We show that all the linear programs of Liu and van Ryzin (2008), Zhang and Adelman (2009) and Kunnumkal and Topaloglu (2008) can be seen as special cases of our linear programming formulation. In particular, for that reason we obtain tighter upper bounds on the objective value than these other approaches and that are asymptotically optimal as time horizon, demand and capacities are linearly scaled up.
- We prove that column generation essentially reduces to solving small mixed integer linear programs. Policies for the MNL model with disjoint consideration sets are numerically tested and show significantly improved results.
- Due to the larger number of constraints, our approach is considerably more expensive than others if we allow the marginal capacity value estimates to change from any possible inventory level to another. However, we find that sensitivity to inventory levels is most pronounced only relatively close to the departures: Therefore, in order to cut down computational requirements for large networks without much deterioration of the solution quality, we can exploit the flexibility of our model with respect to arbitrary aggregations of inventory levels to solve it with high inventory aggregation at the beginning of the booking horizon, and later to re-solve it with lower aggregation and thus higher accuracy so that we capture the typically more pronounced nonlinearity in inventory levels of the value function closer to the end of the time horizon.
- A seemingly new upper bound relationship between the approaches of Zhang and Adelman (2009) and Kunnumkal and Topaloglu (2008) is shown, namely that the former provides a tighter upper bound on the objective value than the latter.

The paper at hand is organized as follows: In the next section we briefly review the related literature, then in Section 3 we present our model including the required notation followed by the resulting Markov decision process and its equivalent linear programming form in Section 4. We introduce the linear programming models that we compare our approach with in Sections 4.1, 4.2 and 5. Our own approach is derived in Section 5 as well. We show that the column generation subproblem is reducible to a mixed integer linear program in Section 6 and describe bid price policies in Section 7. Finally, we present the computational results in Section 8 and conclude in Section 9.

2. Literature Review

The earliest contributions to single leg RM with choice behavior include Brumelle et al. (1990) and Belobaba and Weatherford (1996), amongst others, and for networks the PODS simulation studies by Belobaba and Hopperstad (1999). Zhang and Cooper (2005) consider an inventory control problem of a set of parallel flights including a customer choice model yielding a stochastic optimization problem which is being solved by simulation-based methods. Another simulation-based approach is van Ryzin and Vulcano (2008), who compute virtual nesting controls by constructing a stochastic steepest ascent algorithm designed to find stationary points of the expected revenue function. More contributions have been made, but we refer at this point to the literature reviews of McGill and van Ryzin (1999) and Chiang et al. (2007) and instead focus on papers closer related our approach. The underlying theory of approximate dynamic programming is presented in the well-written books of Bertsekas and Tsitsiklis (1996) and Powell (2007).

Network problems are computationally intensive even without consideration of customer choice behavior, thus good heuristics need to be found. Among the efficient techniques that have been proposed is the so-called choice-based linear program (CDLP) of Gallego et al. (2004). Based on this work, Liu and van Ryzin (2008) present an extension of the standard deterministic linear program approach to include choice behavior. It returns a vector with as many components as there are possible offer sets, and each component represents the number of time periods out the finite time horizon that the corresponding offer set should be available. The notion of *efficient sets* introduced by Talluri and van Ryzin (2004a) for the single leg case is translated into the network context and the authors show that CDLP only uses efficient sets in its optimal solution. Unfortunately, for the network problem the exact optimal policy does not necessarily only use efficient sets like the single leg case, but Liu and van Ryzin (2008) can show asymptotic optimality of the CDLP which indicates that using efficient sets only might be a good choice. A dynamic programming decomposition approach is taken to obtain policies from the static solution of the CDLP and applied to the multinomial logit (MNL) choice model with disjoint consideration sets. Furthermore, the solution to the CDLP constitutes an upper bound on the optimal expected revenue. A generalization of the CDLP that can also handle the MNL choice model with overlapping consideration sets is presented in Miranda Bront et al. (2009), who employ column generation to solve the arising large linear program.

Kunnumkal and Topaloglu (2008) propose an alternative deterministic linear programming approach (ADLP) that exhibits a very similar structure like the CDLP, but they extend the latter to allow for time dependent bid prices in contrast to the static ones produced by the CDLP.

Although no formulation can be proven to dominate the other, their numerical experiments indicate tighter upper bounds on the optimal expected revenue and better policies as well. They also apply their model to the MNL choice model with disjoint consideration sets. Similar results like for the CDLP are presented, including asymptotic optimality, the fact that ADLP provides an upper bound on the objective value and a dynamic programming decomposition approach. The extension comes at the cost of having significantly more constraints in the arising linear program.

3. Model

Products. Let our network consist of m resources—that means flight legs in the airline application—and n products. A product consists of a seat on one or several flight legs in combination with a fare class and departure date. Each resource i has a fixed capacity of c_i, and the network capacity is given by the corresponding vector $c = [c_1, \ldots, c_m]^T$. The capacity is homogenous, that means all seats are perfectly substitutable and do not differ, hence allowing us to accommodate all kind of requests from the given general capacity on a given flight leg. The set of products is denoted by $N = \{1, \ldots, n\}$. Every product j has an associated revenue f_j. By defining $a_{ij} = 1$ if resource i is used by product j, and $a_{ij} = 0$ otherwise, we obtain the incidence matrix $A = (a_{ij}) \in \{0,1\}^{m \times n}$ whose columns shall be denoted by A^j. We assume that each product uses at most one unit of any resource, so $a_{ij} \leq 1$. Group requests can easily accommodated by allowing a_{ij} to be larger than 1. This does not affect the analysis within this paper, however, we will stick to the assumption $a_{ij} \leq 1$ since it simplifies the notation for our proposed aggregated model in Section 5. Each column A^j gives us information about which resources product j uses, and accordingly we write $i \in A^j$ if resource i is being used by product j. The state of the system is given by the vector of unused capacity $x = [x_1, \ldots, x_m]^T$, and selling product j changes x to $x - A^j$.

Customer Choice. Potential customers usually do not come with a predetermined idea of which product to purchase. Rather, they only know some particular features that the product should possess and compare several alternatives that have these features in common before coming to a purchase or non-purchase decision. For example, a customer might be interested in a flight from A to B, but considers several flights with close-by departure times, or several class options. The probability that the customer chooses product j given the set of offered fares S (conditioned to arrival of a customer) is denoted by $P_j(S)$. It satisfies $\sum_{j \in S} P_j(S) + P_0(S) = 1$ for any offer set S, where $j = 0$ denotes the non-purchase option. We keep the choice model general until discussing the column generation procedure where we assume that customers choose according to the multinomial logit choice model with distinct consideration sets.

Table 1 Notation.

m	quantity of resources in the network
$c=[c_1,\dots,c_m]^T$	vector of capacities
n	total number of products
i	index for resources
j	index for products
f_j	fare for product j
$A=(a_{ij})$	incidence matrix, $a_{ij}>0$ if and only if product j needs a_{ij} units (integer) of resource i
A^j	jth column of A
τ	amount of discrete time periods (departure in period $\tau+1$)
λ	arrival probability of a customer.
$N=\{1,\dots,n\}$	set of all products
$N(x)$	set of all feasible products under available capacity x
$u=[u_1,\dots,u_n]^T$	set of products that are available for purchase (expressed as binary vector with n components)
$S\subset N$	set of products that are available for purchase (expressed as subset of N)
$P_j(S)$	general purchase probability of an arrived customer for product j given offer set S
$X=\{0,\dots,,c_1\}\times\cdots\times\{0,\dots,c_m\}$	state space

Decisions on which products to offer are made at discrete points in time such that the time intervals are small enough to have a negligible probability that two or more arrivals occur. A customer arrives in time period t with probability λ. These decision time points are indexed with t starting at time $t=1$ until the end of the booking horizon $t=\tau$. All flights depart at time $t=\tau+1$. The index t can also refer to the time interval between decisions at t and $t+1$ and will be clear from the context.

At each time t we need to decide which products out of the total of n ones shall be offered during time period t. We represent this decision by the binary vector $u_t\in\{0,1\}^n$ or, when it is more convenient to do so, equivalently as a set $S\subset N$ where $j\in S\Leftrightarrow u_j=1$. All customers show up at departure and no cancelations are allowed, thus overbooking is redundant. As a consequence, an offer set is called feasible if there is sufficient network capacity x available to accommodate at least one request for an arbitrary product $j\in S$, and we denote the collection of all feasible offer sets by $N(x):=\{j\in N: a_{ij}\le x_i\,\forall\, i\}$. Note that we omit the time dependence of u, S and x in order to keep the notation simple. The notation is summarized in Table 1.

4. Current Solution Approaches

Let $v_t(x)$ denote the expected revenue-to-go from time period t until the final period τ, given the vector $x \in X$ of still available resources in the network. The well-known optimality equation for maximizing expected revenue is then given by

$$\begin{aligned} v_t(x) &= \max_{S \subseteq N(x)} \sum_{j \in S} \lambda P_j(S)(f_j + v_{t+1}(x - A^j)) + (\lambda P_0(S) + 1 - \lambda) v_{t+1}(x), \\ &= \max_{S \subseteq N(x)} \sum_{j \in S} \lambda P_j(S) \left[f_j - (v_{t+1}(x) - v_{t+1}(x - A^j)) \right] + v_{t+1}(x), \qquad \forall t, x, \end{aligned} \tag{1}$$

with boundary condition $v_{\tau+1}(x) = 0$ for all x. The decision to be made within each time period is which set of products to offer before we can observe demand in the corresponding period. Under the independent demand assumption, in contrast, decisions and demand are decoupled. If we could somehow compute the value function v, then, for given t and x, the optimal policy is to offer

$$S^*(t,x) := \arg\max_{S \subseteq N(x)} \sum_{j \in S} \lambda P_j(S) \left[f_j - \left(v_{t+1}(x) - v_{t+1}(x - A^j) \right) \right]. \tag{2}$$

Note that the value function v is only required for the expression $v_{t+1}(x) - v_{t+1}(x - A^j)$ that denotes the so-called *opportunity cost* of selling product j in time period t at capacity state x. Having correct opportunity costs thus means having the optimal policy. The problem (1) is intractable due to the large state space, so in the following we restate several heuristics that can be used to estimate the opportunity cost and thus to obtain a policy.

4.1. Choice-Based Deterministic LP

In order to reduce the problem to a tractable size, Gallego et al. (2004) and Liu and van Ryzin (2008) propose a choice-based deterministic linear program (CDLP) where demand is treated as known and being equal to its expected value. The problem reduces then to an allocation problem where we need to decide for how many time periods a certain set of products S shall be offered, denoted by $h(S)$. Denote the *expected total revenue* from offering S by

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

and the *expected total consumption of resource* i from offering S by

$$Q_i(S) = \sum_{j \in S} P_j(S) a_{ij}, \qquad \forall i.$$

Then the choice-based deterministic linear program is given by

$$(\textbf{CDLP}) \quad z_{\text{CDLP}} = \max_h \sum_{S \subseteq N} \lambda R(S) h(S)$$

$$
\begin{aligned}
&\sum_{S\subseteq N} \lambda Q_i(S)h(S) \leq c_i, && \forall i,\\
&\sum_{S\subseteq N} h(S) = \tau, &&\\
&h(S) \geq 0, && \forall S\subseteq N.
\end{aligned}
$$

It essentially parallels the well-known deterministic LP for the no-choice case, provides with z_{CDLP} an asymptotically tight upper bound on the optimal expected revenue under fluid scaling and is very fast, but has the disadvantage that it does not provide the order in which the optimal sets shall be used since every order yields the same expected revenue under these model assumptions. In order to construct a policy we can use the dual values π_i associated with the capacity constraints of CDLP. These values represent an estimation of the marginal value of capacity of each resource so that we can approximate the opportunity cost by

$$
v_{t+1}(x) - v_{t+1}(x - A^j) \approx \sum_{i\in A^j} \pi_i, \tag{3}
$$

and subsequently obtain a policy by substituting the opportunity cost with the above estimate in the optimal policy (2). However, these estimates π suffer from being neither time- nor inventory-level dependent. To remedy this shortcoming, Liu and van Ryzin (2008) propose a dynamic programming decomposition which we outline in Section 7.2.

4.2. Alternative Deterministic LP

Kunnumkal and Topaloglu (2008) propose to generate time-depdent marginal capacity value estimates with an alternative deterministic linear program (ADLP). This formulation also results in an asymptotically tight upper bound on the optimal expected revenue, but none of the bounds generated by CDLP and ADLP dominate each other in general. However, their numerical experiments indicate that the ADLP can provide tighter bounds than the CDLP.

$$
\begin{aligned}
(\textbf{ADLP}) \qquad z_{\text{ADLP}} = \max_h &\sum_{t=1}^{\tau} \sum_{S\subseteq N} \lambda R(S)h_t(S) &&\\
&\sum_{k=1}^{t-1} \sum_{S\subseteq N} \lambda Q_i(S)h_k(S) + \sum_{S\subseteq N} \mathbf{1}_{\{j\in S\}} a_{ij} h_t(S) \leq c_i, && \forall i,t,j\in N,\\
&\sum_{S\subseteq N} h_t(S) = 1, && \forall t,\\
&h_t(S) \geq 0, && \forall S\subseteq N, t.
\end{aligned}
$$

Note that the CDLP and ADLP have a similar structure: While $h(S)$ was in the CDLP the scalar that indicated how much time to allocate over the full time horizon for the offer set S, in the

ADLP, the variable $h_t(S)$ indicates how much time to allocate within time period t to the offer set S. We can also interpret $h_t(S)$ as the frequency of offering S in period t.

As a policy, we can use the dual values $\pi_{i,t,j}$ of the capacity constraints to approximate the opportunity cost by

$$v_{t+1}(x) - v_{t+1}(x - A^j) \approx \sum_{i \in A^j} \Big[\sum_{u=t+1}^{\tau} \sum_{k \in N} \pi_{i,u,k} \Big]. \tag{4}$$

The expression in brackets can be interpreted as the marginal value of resource i over the remaining time horizon.

5. Approximation Based on the Equivalent LP

The following linear programming formulation will serve as the starting point of our considerations. It is equivalent to the dynamic program (1) and, for that reason, we denote it by (**EQ**). The equivalence can be derived from fundamental results of value iteration, see Powell (2007), for example.

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

The decision variables are $v_t(x)$, for all t, x, and therefore the problem is also intractable for a large state space. The basic idea is now to approximate $v_t(\cdot)$ by a given set of κ basis functions $\phi_k(\cdot)$ in order to reduce the number of variables $v_t(x) \approx \sum_{k=1}^{\kappa} V_{t,k} \phi_k(x)$, for all t, x. Our approach is based on Zhang and Adelman (2009), who consider the affine approximation

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

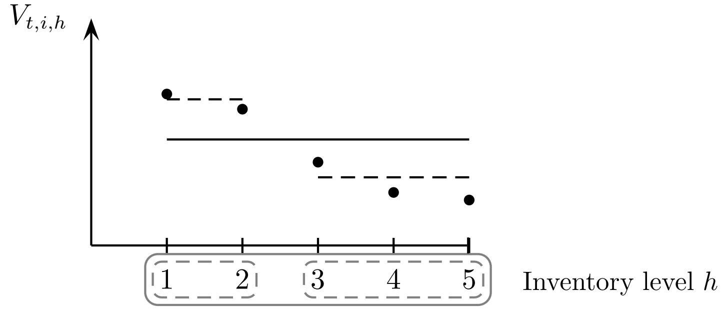
with boundary conditions $\theta_{\tau+1} = 0$ and $V_{\tau+1,i} = 0$ for all flight legs i. In this approximation, $V_{t,i}$ estimates the marginal inventory value on flight i in period t without taking into account how many seats are still available on this flight leg. The optimal marginal capacity values $V^*_{t,i}$ provide an estimation of the opportunity cost via

$$v_{t+1}(x) - v_{t+1}(x - A^j) \approx \sum_{i \in A^j} V^*_{t+1,i}. \tag{5}$$

Their basis functions are given by

$$\phi_k(x) := \begin{cases} x_i, & k = i \in \{1, \ldots, m\}, \\ 1, & k = m + 1. \end{cases}$$

Figure 1 **Examples of inventory aggregations on a resource i with $c_i = 5$.**



Substituting the resulting approximation into (**EQ**) and constructing its dual yields the linear program:

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

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

$$\sum_{x,S} Y_{t,x,S} = \begin{cases} 1, & \text{for } t=1, \\ \sum_{x,S} Y_{t-1,x,S}, & \forall t = 2,\dots,\tau, \end{cases}$$

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

Their approach can be seen as a special case of our approximation, which we elaborate in the following: Our essential refinement is to split 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^i_k, and can reach from unit size 1 to resource capacity c_i. Note, in particular, that it can vary between resources. Naturally, the sum of all range sizes must equal the total capacity of resource i, that is $\sum_{k=1}^{K_i} s^i_k = c_i$. For notational convenience, we introduce for each resource i a function

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

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. In particular, $r_i(c_i) = K_i$. For example, in Figure 1 we depict an aggregation into $K_i = 2$ ranges by a dotted line; the first range has size $s^i_1 = 2$ including the inventory levels 1 and 2, and the second has size $s^i_2 = 3$ including inventory levels 3, 4 and 5. Therefore, we have $r_i(1) = r_i(2) = 1$ and $r_i(3) = r_i(4) = r_i(5) = 2$ for this aggregation. The marginal capacity value $V_{t,i,k}$ corresponding to the two ranges is represented by the dotted lines. In the following, we omit the subscript i in the function $r_i(\cdot)$ because its argument, as for example in $r(x_i)$, will make it clear that the function depends on the resource i.

We approximate the value function with

$$v_t(x) \approx \theta_t + \sum_{i=1}^{m} \Bigg[\sum_{k=1}^{r(x_i)-1} s_k^i V_{t,i,k} + (x_i - \sum_{k=1}^{r(x_i)-1} s_k^i) V_{t,i,r(x_i)} \Bigg]. \tag{6}$$

On the boundary we define $V_{\tau+1,i,k} = 0$ for all i, k, and $\theta_{\tau+1} = 0$. Note that in the range of x_i (denoted by $r(x_i)$), there are only $x_i - \sum_{k=1}^{r(x_i)-1} s_k^i$ units of inventory left. The non-linear approximation has the particular advantage that the estimated marginal value of capacity depends on both time and inventory level. We therefore refer to our approach as the "Time and Inventory Sensitive Approach (TISA)".

Figure 1 gives three examples of how we could aggregate inventory levels: The solid line represents aggregation of the entire inventory of a resource, so that we only have one marginal inventory value $V_{t,i}$ for any inventory level. If done for all resources, then the problem reduces to the affine approximation by Zhang and Adelman (2009). On the other extreme, we might disaggregate completely so that we have a potentially different marginal value $V_{t,i,h}$ for each inventory level h, which would correspond to the dots in Figure 1, however, computationally it becomes quickly expensive to solve the associated problem for larger networks. Any other aggregation is possible, for example, we could split the inventory and obtain two dashed ranges. Furthermore, it is also possible to aggregate across time to further reduce the size of the linear program by exploiting that the marginal values of capacity typically stay nearly constant when there is much time left to departure. Likewise, inventory aggregations could be designed to change over time. In this paper, however, we stick to static inventory aggregation to increase readability.

Plugging the approximation (6) into (**EQ**) results in the following linear program, where we made use of the assumption $a_{ij} \leq 1$ in order to simplify notation, since it implies that $0 \leq r(x_i) - r(x_i - a_{ij}) \leq 1$.

$$
\begin{aligned}
(\mathbf{D}) \min_{\theta,V} & \sum_{i=1}^{m} \sum_{k=1}^{K_i} s_k^i V_{1,i,k} + \theta_1 \\
& \theta_t - \theta_{t+1} + \sum_{i=1}^{m} \Bigg[\sum_{k=1}^{r(x_i)-1} s_k^i V_{t,i,k} + \big(x_i - \sum_{k=1}^{r(x_i)-1} s_k^i\big) V_{t,i,r(x_i)} - \sum_{k=1}^{r(x_i)-2} s_k^i V_{t+1,i,k} \\
& \quad + \Bigg(- s^i_{r(x_i)-1} + \lambda \sum_{j \in S} P_j(S) \mathbf{1}_{\{r(x_i - a_{ij}) < r(x_i)\}} \big[s^i_{r(x_i)-1} - x_i + a_{ij} \\
& \quad + \sum_{k=1}^{r(x_i - a_{ij})-1} s_k^i \big] \Bigg) V_{t+1,i,r(x_i)-1} + \Bigg(\big(\sum_{k=1}^{r(x_i)-1} s_k^i - x_i \big) + \lambda \sum_{j \in S} P_j(S) \big[x_i
\end{aligned}
$$

$$- \sum_{k=1}^{r(x_i)-1} s_k^i - (x_i - a_{ij} - \sum_{k=1}^{r(x_i-a_{ij})-1} s_k^i)\mathbf{1}_{\{r(x_i-a_{ij})=r(x_i)\}}\Big]\Big)V_{t+1,i,r(x_i)}\Big]$$
$$\geq \lambda \sum_{j\in S} P_j(S) f_j, \qquad \forall t, x, S \subseteq N(x).$$

To increase readability, let us abbreviate the coefficients of $V_{t+1,i,r(x_i)-1}$ and $V_{t+1,i,r(x_i)}$ as stated in (**D**) by $\beta_{i,x,S}$ and $\gamma_{i,x,S}$, respectively. The dual of the above problem is given by:

$$(\mathbf{P}) \quad z_{\text{TISA}} = \max_Y \sum_{t,x,S} \lambda \sum_{j\in S} P_j(S) f_j Y_{t,x,S}$$
$$\sum_{x,S}\Big(s_k^i \mathbf{1}_{\{k\leq r(x_i)-1\}} + (x_i - \sum_{\tilde{k}=1}^{r(x_i)-1} s_{\tilde{k}}^i)\mathbf{1}_{\{k=r(x_i)\}}\Big)Y_{t,x,S} = s_k^i, \qquad \text{for } t=1, \forall i,k,$$
$$\sum_{x,S}\Big(-s_k^i\mathbf{1}_{\{k\leq r(x_i)-2\}} + \beta_{i,x,S}\mathbf{1}_{\{k=r(x_i)-1\}} + \gamma_{i,x,S}\mathbf{1}_{\{k=r(x_i)\}}\Big)Y_{t-1,x,S} + \sum_{x,S}\Big(s_k^i\mathbf{1}_{\{k\leq r(x_i)-1\}}$$
$$+ (x_i - \sum_{\tilde{k}=1}^{r(x_i)-1} s_{\tilde{k}}^i)\mathbf{1}_{\{k=r(x_i)\}}\Big)Y_{t,x,S} = 0, \qquad \text{for } t>1, \forall i,k,$$
$$\sum_{x,S} Y_{t,x,S} = 1, \qquad \text{for } t=1,$$
$$\sum_{x,S}(Y_{t,x,S} - Y_{t-1,x,S}) = 0, \qquad \forall t>1,$$
$$Y_{t,x,S} \geq 0, \qquad \forall\, t,x,S\subseteq N(x).$$

We refer to (**P**) in the special case that $K_i = c_i$, $s_k^i = 1$, $\forall\, k,i$ as TISAC, and for $K_i = 2$, $s_k^i = \lfloor c_i/2 \rfloor$, $\forall\, k,i$ as TISA2. To provide some intuition regarding constraints and variables of (**P**), note that the decision variables $Y_{t,x,S}$ can be interpreted as state-action probabilities since they are non-negative and satisfy $\sum_{x,S} Y_{t,x,S} = 1$ for all t. With this in mind, the first set of constraints in (**P**) can be understood as the expected available capacity on resource i within inventory range k at the beginning of the booking horizon, which has to equal the size of the range s_k^i. The second set of constraints can similarly be seen as expected available capacity in range k on resource i at time t, which equals expected available capacity in that range at time $t-1$ minus expected consumption within period $t-1$.

Proposition 1 *For an arbitrary inventory aggregation, any feasible solution to the corresponding linear program* (**P**) *yields a feasible solution to* (**AFF**) *having the same objective function value. We have the following upper bounds on the optimal expected revenue $v_1(c)$:*

$$z_{\text{CDLP}} \geq z_{\text{AFF}} \geq z_{\text{TISA}} \geq v_1(c).$$

In particular, the objective in problem (**P**) *is asymptotically optimal.*

Proof. Suppose (Y) solves $(\mathbf{P})$. In order to show the second inequality, we need to show that Y yields a feasible solution to $(\mathbf{AFF})$ yielding the same objective function value. Apparent from feasibility to $(\mathbf{P})$, we have

$$\begin{aligned}
&\sum_{x,S} Y_{t,x,S} = 1, && \text{for } t = 1,\\
&\sum_{x,S} Y_{t,x,S} = \sum_{x,S} Y_{t-1,x,S}, && \forall\, t > 1,\\
&Y_{t,x,S} \geq 0, && \forall\, t, x, S \subseteq N(x).
\end{aligned}$$

From the first set of constraints in $(\mathbf{P})$, we obtain for a fixed resource i and $t = 1$ by summation over all inventory level ranges $k \in \{1, \ldots, K_i\}$:

$$\begin{aligned}
\sum_{x,S} \Big(\sum_{k=1}^{r(x_i)-1} s_k^i + x_i - \sum_{k=1}^{r(x_i)-1} s_k^i \Big) Y_{t,x,S} &= \sum_{k=1}^{K_i} s_k^i\\
\Leftrightarrow \quad \sum_{x,S} Y_{t,x,S} x_i &= c_i.
\end{aligned}$$

For $t > 1$, fix a resource i and sum the second set of constraints in $(\mathbf{P})$ over all ranges $k = 1, \ldots, K_i$:

$$\begin{aligned}
&\sum_{x,S} Y_{t,x,S} x_i + \sum_{x,S} Y_{t-1,x,S} \Bigg\{ -\sum_{k=1}^{r(x_i)-2} s_k^i + \mathbf{1}_{\{r(x_i)>1\}} \Bigg(-s^i_{r(x_i)-1} + \lambda \sum_{j\in S} P_j(S) [s^i_{r(x_i)-1} - x_i + a_{ij}\\
&+ \sum_{k=1}^{r(x_i-a_{ij})-1} s_k^i] \mathbf{1}_{\{r(x_i-a_{ij})<r(x_i)\}} \Bigg) + \Bigg(\sum_{k=1}^{r(x_i)-1} s_k^i - x_i + \lambda \sum_{j\in S} P_j(S) [x_i - \sum_{k=1}^{r(x_i)-1} s_k^i\\
&- (x_i - a_{ij} - \sum_{k=1}^{r(x_i-a_{ij})-1} s_k^i) \mathbf{1}_{\{r(x_i-a_{ij})=r(x_i)\}}] \Bigg) \Bigg\} = 0.
\end{aligned}$$

Let us denote the term in curly brackets by ψ:

- If $r(x_i) = 0$, then $x_i = 0$, and due to $S \subseteq N(x)$ we have $a_{ij} = 0$ for all $j \in S$, resulting in $\psi = -x_i$.
- If $r(x_i) = 1$, then $\psi = -x_i + \lambda \sum_j P_j(S) a_{ij}$ follows directly.
- If $r(x_i) \in \{2, \ldots, K_i\}$: then we obtain the following: (without loss of generality, we assume $a_{ij} \leq 1$ to simplify notation)

$$\begin{aligned}
\psi &= -x_i + \lambda \sum_{j\in S} P_j(S) [(s^i_{r(x_i)-1} - x_i + a_{ij} + \sum_{k=1}^{r(x_i-a_{ij})-1} s_k^i) \mathbf{1}_{\{r(x_i-a_{ij})<r(x_i)\}}\\
&\quad + x_i - \sum_{k=1}^{r(x_i)-1} s_k^i - (x_i - a_{ij} - \sum_{k=1}^{r(x_i-a_{ij})-1} s_k^i) \mathbf{1}_{\{r(x_i-a_{ij})=r(x_i)\}}]\\
&= -x_i + \lambda \sum_{j\in S} P_j(S) a_{ij}.
\end{aligned}$$

Thus we obtain feasibility to $(\mathbf{AFF})$, and therefore validity of the inequality $z_{\text{AFF}} \geq z_{\text{conc}}$. Alternatively, we can obtain this result from using the dual instead of the primal problem by starting from an optimal dual solution $(\theta^*, V^*_{t,i})$ to (**AFF**). Setting $\theta := \theta^*$ and $V_{t,i,k} := V^*_{t,i}$ yields a solution feasible to (**D**) with the same objective, yielding the desired result.

The last inequality $z_{\text{TISA}} \geq v_1(c)$ follows from the fact that every feasible solution to (**EQ**) is an upper bound to the exact value function. This fact is a standard result in value iteration, see for example Theorem 3.4.1 in Powell (2007).

Zhang and Adelman (2009) showed that (**AFF**) has a tighter bound than the deterministic LP, and since Liu and van Ryzin (2008) proved that the deterministic LP is asymptotically optimal (that is, z_{CDLP} converges to $v_1(c)$ as demand, capacity and time horizon are linearly scaled up), both (**AFF**) and (**P**) are as well asymptotical optimal in that respect. □

Furthermore, we can also show that the affine approximation problem (**AFF**) results in a tighter upper bound on the optimal expected revenue than the ADLP, which likewise seems to be a new result.

Proposition 2 *Any feasible solution to* (**AFF**) *yields a feasible solution to* (**ADLP**) *having the same objective value. We have the following bounds on the objective value $v_1(c)$:*

$$z_{\text{ADLP}} \geq z_{\text{AFF}} \geq v_1(c).$$

Proof. Let Y be a feasible solution to (**AFF**). We define

$$h_t(S) := \sum_x Y_{t,x,S}, \qquad \forall S \subseteq N, t,$$

and need to show that this is a feasible solution to (**ADLP**) with the same objective value.

First, we have directly from $Y \geq 0$ and the definition of $h_t(S)$ that $h_t(S) \geq 0$ for all S, t. Next, note that the second set of equality constraints in (**AFF**) actually reduces to the condition $\sum_{x, S \subseteq N(x)} Y_{t,x,S} = 1$ for all t. Using the definition of $h_t(S)$, we obtain

$$\sum_{S \subseteq N(x)} h_t(S) = \sum_{S \subseteq N} h_t(S) = 1,$$

for all t as required, where the first equality stems from $Y_{t,x,S} = 0$ if $S \not\subseteq N(x)$ because of Y's feasibility to (**AFF**). It remains to show that the first set of inequalities in (**ADLP**) holds, and that the objective value stays the same. As for the objective value, we defined earlier the total expected revenue from offering set S by $R(S) := \sum_{j \in S} P_j(S) f_j$. Substituting this into the objective in (**AFF**) and making use of the definition of $h_t(S)$ shows the equivalence of the objective. Finally, in order to show that the first set of inequalities in (**ADLP**) holds, we keep i fixed and sum the first set of equality constraints of (**AFF**) over time from 1 to some fixed $t \in \{1, \ldots, \tau\}$:

$$\sum_{k=1}^{t} \sum_{x,S} x_i Y_{k,x,S} = c_i + \sum_{k=2}^{t} \sum_{x,S} x_i Y_{k-1,x,S} - \sum_{k=2}^{t} \sum_{x,S} \sum_{j \in S} \lambda P_j(S) a_{ij} Y_{k-1,x,S}.$$

Canceling terms and rearranging yields

$$c_i = \sum_{x,S} x_i Y_{t,x,S} + \sum_{k=1}^{t-1} \sum_{S} \lambda Q_i(S) h_k(S),$$

where the total expected consumption on resource i is $Q_i(S) := \sum_{j \in S} P_j(S) a_{ij}$ as defined earlier. Due to the feasibility of Y to (**AFF**), $Y_{t,x,S} > 0$ only if $S \subseteq N(x) = \{j \in N : a_{ij} \leq x_i \, \forall i \in A^j\}$. Therefore we have

$$\sum_{x,S} x_i Y_{t,x,S} \geq \sum_{S} \mathbf{1}_{\{j \in S\}} a_{ij} h_t(S), \qquad \forall j \in N,$$

which concludes the proof. □

From Proposition 1 and Proposition 2 it follows that our approach (**P**) provides also a tighter bound than the ADLP.

Corollary 1 *We have the following bounds on the optimal expected revenue $v_1(c)$:*

$$z_{\text{ADLP}} \geq z_{\text{TISA}} \geq v_1(c).$$

6. Solution via Column Generation

The problem (**P**) has $O(\tau \prod_{i=1}^{m} (c_i + 1) 2^n)$ variables and, for realistic network sizes, cannot be solved in moderate time unless techniques such as column generation are used to deal with problem size. This method builds upon the observation that for large problems most columns never enter the basis matrix and therefore do not need to be stored. Apparently, the main task is then to provide a way of how to find the next column to enter the basis without having to generate the whole coefficient matrix. We show in the following that if we use the *multinomial logit choice model with disjoint consideration sets* this so-called column generation subproblem reduces to solving a small linear mixed integer program. For the sake of improved readability, we confine ourselves to demonstrate the derivation of the pricing problem for a special case only, namely $K_i = c_i$ for every resource i. For any other choice of aggregation, the derivation works in the same way. The considered case is the approximation proposed by Farias and Van Roy (2007) in the no-choice context which reflects the fact that the marginal value of capacity also depends on the quantity of remaining unused inventory. Our approximation for general aggregation (6) reduces in this case to

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

with boundary conditions $V_{\tau+1,i,k} = 0$ for all i, k and $\theta_{\tau+1} = 0$. An initial feasible solution for the column generation procedure is given by

$$Y_{t,x,S} = \begin{cases} 1, & \text{if } x = c, S = \emptyset, \forall\, t \\ 0, & \text{otherwise.} \end{cases}$$

Next, we need to check for optimality and in case that it is not attained yet, we also need to find the next column that shall enter the basis. Given the dual values V at some iteration, this is achieved by finding the column with maximal reduced profit, the latter being given by

$$\max_{t,x,S\subseteq N(x)} \left(\lambda \sum_{j\in S} P_j(S) f_j - \sum_{i=1}^{m} \left[\sum_{k=1}^{x_i} V_{t,i,k} - \sum_{k=1}^{x_i-1} V_{t+1,i,k} - (1 - \lambda \sum_{j\in S} P_j(S) a_{ij}) V_{t+1,i,x_i} \right] + \theta_{t+1} - \theta_t \right). \tag{7}$$

We refer to the problem (7) as the *pricing problem.* If the result is nonpositive then optimality has been reached, otherwise we add the corresponding column to the basis. Several variants of the column generation algorithm exist, for example, we could retain all columns that once entered the basis and thus obtain a system of growing size, or we could remove all columns that exit the basis, or use some other rule in between. The most important question, however, is whether the maximal reduced profit can be found quickly and inexpensively. The maximization in (7) could potentially be expensive to solve, so let us focus on this subproblem. Rearrangement of terms yields:

$$\max_{t,x,S\subseteq N(x)} \lambda \sum_{j\in S} P_j(S)(f_j - \sum_{i=1}^{m} a_{ij} V_{t+1,i,x_i}) - \sum_{i=1}^{m} \sum_{k=1}^{x_i} (V_{t,i,k} - V_{t+1,i,k}) + \theta_{t+1} - \theta_t. \tag{8}$$

Difficulties stem from the probability term $P_j(S)$ since it makes the problem nonlinear, and in particular the requirement $S \subseteq N(x)$ forces S to be dependent on x which makes the two variables non-separable. So far, we have not specified a choice model from which we can derive $P_j(S)$, however, we need to do so now in order to solve the pricing problem. We consider choice probabilities $P_j(S)$ derived from the multinomial logit choice model with disjoint consideration sets. For this model, we divide customers 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. The means of segmentation are left unspecified; they could base for example on itinerary and departure time (early morning, midday etc). We assume that the consideration sets are disjoint, that is $C_{l_1} \cap C_{l_2} = \emptyset$ for any segments $l_1 \neq l_2 \in \tilde{L}$. The probability that a customer in segment l purchases product j when we offer the fare set S is given by $P_{lj}(S) = v_{lj} / (\sum_{j\in C_l \cap S} v_{lj} + v_{l0})$ for $S \subseteq N$, where v_{l0} is the preference for not buying anything. These preference values could, for example, be derived from the reservation price of the segment for a particular product, and set equal to the maximum of this reservation price minus the actual price and zero. An arriving customer belongs to segment l with probability p_l such that $\sum_l p_l = 1$, hence we can define arrival probabilities $\lambda_l := p_l \lambda$ for every segment. Taken together we have $\lambda = \sum_l \lambda_l$. For a given segment l, let the vector u_l describe the product availability such that $u_{lj} = 1$ if product $j \in C_l$ is available and $u_{lj} = 0$ otherwise. Accordingly, the probability that a customer from segment l purchases product j can be rewritten in the following form:

$$P_{lj}(u_l) = \frac{u_{lj} v_{lj}}{\sum_{k \in C_l} u_{lk} v_{lk} + v_{l0}}.$$

The purchase probability for product j given the arrival of a customer is then defined by

$$P_j(S) = p_l P_{lj}(u_l(S)),$$

where $p_l = \lambda_l / \lambda$ and $u_l(S)$ is a vector with $u_{lj} = 1$ if $j \in S \cap C_l$ and $u_{lj} = 0$ otherwise.

We substitute this choice probability into the pricing problem (8) and keep a time period t fixed. This results in a nonlinear maximization problem over the variables x and u:

$$\begin{aligned}
\max_{x,u} \sum_{l \in \tilde{L}} \sum_{j \in C_l} & \frac{\lambda_l v_{lj} u_{lj}}{\sum_{k \in C_l} v_{lk} u_{lk} + v_{l0}} \left[f_j - \sum_{i=1}^{m} a_{ij} V_{t+1,i,x_i} \right] \\
& - \sum_{i=1}^{m} \sum_{k=1}^{x_i} (V_{t,i,k} - V_{t+1,i,k}) + \theta_{t+1} - \theta_t \\
x_i \geq a_{ij} u_{lj}, \quad & \forall i, j \in C_l, l \in \tilde{L}, \qquad (9) \\
x_i \in \{0, \ldots, c_i\}, \quad & \forall i, \qquad (10) \\
u_{lj} \in \{0, 1\}, \quad & \forall j \in C_l, l \in \tilde{L}. \qquad (11)
\end{aligned}$$

Next, we perform a change of variables as done in Chapter 4.3.2 of Boyd and Vandenberghe (2004): Define $z_{lj} = u_{lj} / (\sum_{k \in C_l} v_{lk} u_{lk} + v_{l0})$ for all products $j \in C_l$ and all segments $l \in \tilde{L}$, furthermore, define $\alpha_l := 1 / (\sum_{k \in C_l} v_{lk} u_{lk} + v_{l0})$ for all segments $l \in \tilde{L}$. Note that $z_{lj} = u_{lj} \alpha_l$, hence $z_{lj} \in \{0, \alpha_l\}$. The latter constraint can be expressed by

$$\begin{aligned}
z_{lj} &\geq 0, & \forall j \in C_l, l \in \tilde{L}, \qquad (12) \\
z_{lj} &\leq \alpha_l, & \forall j \in C_l, l \in \tilde{L}, \qquad (13) \\
M(1 - u_{lj}) + z_{lj} &\geq \alpha_l, & \forall j \in C_l, l \in \tilde{L}, \qquad (14) \\
z_{lj} &\leq M u_{lj}, & \forall j \in C_l, l \in \tilde{L}. \qquad (15)
\end{aligned}$$

We use here a so-called "Big M"-method to enforce the correct relationship between z_{lj}, α_l and u_{lj}. It is well-known that this method can be very detrimental to solving mixed integer programmes, see Camm et al. (1990), for instance. To avoid numerical difficulties and slow convergence we should keep the scalar M as small as possible. It is not difficult to see that $M := \max_l 1/v_{l0}$ is both sufficiently large and constitutes a tight upper bound on z_{lj} and α_l. Even though some solvers might be able to handle the constraints $z_{lj} \in \{0, \alpha_l\}$ directly, we still give the formulation above in particular because it allows to link the variable u in capacity availability constraint (9) to z and thereby avoids having to replace u_{lj} by the nonlinear expression z_{lj}/α_l.

By definition of z_{lj} and α_l we have

$$\alpha_l \geq 0, \qquad \forall\, l \in \tilde{L}, \tag{16}$$

$$\sum_{j \in C_l} v_{lj} z_{lj} + v_{l0}\alpha_l = 1, \qquad \forall\, l \in \tilde{L}. \tag{17}$$

We call the resulting nonlinear auxiliary problem (**AUX**) for reference:

$$\max_{u,x,z,\alpha} \sum_{l \in \tilde{L}} \sum_{j \in C_l} \lambda_l v_{lj} \left[f_j - \sum_{i=1}^{m} a_{ij} V_{t+1,i,x_i} \right] z_{lj} - \sum_{i=1}^{m} \sum_{k=1}^{x_i} (V_{t,i,k} - V_{t+1,i,k}) + \theta_{t+1} - \theta_t$$

subject to (9)–(17).

The parameters V_{t+1,i,x_i} depend on x, so some more auxiliary binary variables are needed to reformulate the problem as a linear mixed integer program:

Proposition 3 *Suppose the preference for non-purchase is positive for all segments, that means $v_{l0} > 0$ for all $l \in \tilde{L} := \{1, \dots, L\}$, $a_{ij} \in \{0,1\}$ and let M be an arbitrary scalar greater than or equal to 1. We only need to solve the following linear mixed integer program to find the solution for problem (8) for each $t \geq 1$:*

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

$$\sum_{k=1}^{c_i} x^{ki} \geq a_{ij} v_{l0} z_{lj}, \qquad \forall\, i, l, j \in C_l, \tag{18}$$

$$x^{k-1,i} \geq x^{ki}, \qquad \forall\, i, k \in \{2, \dots, c_i\}, \tag{19}$$

$$y_{lj}^{ki} \leq x^{ki}, \qquad \forall\, l, j \in C_l, k, i, \tag{20}$$

$$y_{lj}^{ki} \leq z_{lj}, \qquad \forall\, l, j \in C_l, k, i, \tag{21}$$

$$y_{lj}^{ki} \geq z_{lj} - M(1 - x^{ki}) \qquad \forall\, l, j \in C_l, k, i, \tag{22}$$

$$x^{ki} \in \{0,1\}, \qquad \forall\, i, k \in \{1, \dots, c_i\},$$

$$y_{lj}^{ki} \geq 0, \qquad \forall\, l, j \in C_l, k, i, \tag{23}$$

subject to (11)–(17).

Proof. We start from problem (**AUX**) and introduce new binary variables $x^{ki} \in \{0,1\}$ for all $k \in \{1,\dots,n\}$ and all resources i such that $x_i = \sum_k x^{ki}$. With these new variables we can rewrite

$$\sum_i \sum_{k=1}^{x_i} (V_{t+1,i,k} - V_{t,i,k}) = \sum_i \sum_{k=1}^{c_i} (V_{t+1,i,k} - V_{t,i,k}) x^{ki}.$$

By imposing constraints (19) we ensure that x^{ki} is monotone decreasing in k for fixed i and hence that we have a one-to-one correspondence between a vector $[x^{1,i},\dots,x^{c_i,i}]$ and x_i. Furthermore,

$$V_{t+1,i,x_i} = \sum_{k=1}^{c_i-1} V_{t+1,i,k}(x^{ki} - x^{k+1,i}) + V_{t+1,i,c_i} x^{c_i,i}. \tag{24}$$

The constraints (9), which ensure that only allowable offer sets are used, become under the new variable (x^{ki}) the constraints (18). Note that by allowable offer sets we mean offer sets $S \subseteq N(x)$, that is we have sufficient capacity to accommodate at least one request for any product $j \in S$. We carry out the change of variables from x to (x^{ki}), which leaves us with an indefinite quadratic programme featuring the nonlinear terms $x^{ki} z_{lj}$ (originating from substituting (24) for V_{t+1,i,x_i}) in the objective. Hence we further introduce new variables $y_{lj}^{ki} = x^{ki} z_{lj} \in \{0, z_{lj}\}$. Imposing the constraints (20–23) guarantees that $y_{lj}^{ki} = z_{lj}$ if $x^{ki} = 1$ and $y_{lj}^{ki} = 0$ otherwise. Since $z_{lj} \le 1$ by definition, every $M \ge 1$ can be used in (22). □

For any segment l, the non-purchase preference v_{l0} must not be equal to zero because otherwise we would divide by zero in the definition of α_l if we choose to close all products in C_l. The assumption is reasonable because typically the customers' choices depend on their limited budget as well as on available products of competitors.

7. Policies

In this section, we address the question of how the solution to (**D**) can actually be used to obtain a control policy that tells us which set of fares S to offer at any given time t and state x of the network. Again, in order to improve readability we discuss the policies for the entirely disaggregated case, that is $K_i = c_i$ for all resources i. For any other aggregation of inventory, similar policies can be derived by the same argumentation.

7.1. Opportunity Cost Estimates Directly from (D)

A standard approach of finding such a policy is to use the optimal dual solution of the capacity constraints of the respective linear programme as a means to approximate the opportunity cost of the resources. In a given time period t and having a given available network inventory x, we approximate the opportunity cost of selling a product j with the sum of the marginal inventory values of the resources $i \in A^j$ that this product uses. In a formula, the opportunity cost approximation—using the optimal dual values V^* to (**D**)—is the following expression:

$$v_{t+1}(x) - v_{t+1}(x - A^j) \approx \sum_i \sum_{k=x_i-a_{ij}+1}^{x_i} V^*_{t+1,i,k}.$$

Since the consideration sets C_l are assumed to be disjoint, segment-wise maximization is feasible. We need to solve a maximization problem hence for each segment l in each time period t and state x to obtain the optimal offer set S_l^*, and Bellman equation (1) indicates that it has the form

$$S_l^* = \operatorname*{arg\,max}_{S_l \subseteq C_l \cap N(x)} \sum_{j \in C_l} P_j(S_l) \left[f_j - \sum_{i=1}^{m} \sum_{k=x_i-a_{ij}+1}^{x_i} V^*_{t+1,i,k} \right]. \tag{25}$$

The term in brackets is the "worth" of product j, that means its revenue minus its approximated opportunity cost. We abbreviate this term with w_j:

$$w_j := f_j - \sum_{i=1}^{m} \sum_{k=x_i-a_{ij}+1}^{x_i} V^*_{t+1,i,k}.$$

For the MNL choice model with preference vector v_l for segment l, rewriting the above maximization problem (25) in terms of a binary availability vector u_l yields a system of the form

$$\begin{aligned} \max_{u_l \in \{0,1\}^{|C_l|}} \sum_{j \in C_l} \frac{v_{lj} u_{lj} w_j}{\sum_{k \in C_l} v_{lk} u_{lk} + v_{l0}}, \qquad & \forall x, t, \\ u_{lj} \le \mathbf{1}_{\{x \ge A^j\}}, \qquad & \forall j \in C_l, \end{aligned} \tag{26}$$

The maximization (26) can be solved in the following way:

Proposition 4 *Consider the optimization problem (26). Rank the values w_j in a decreasing order; that is, $w_{[1]} \ge \cdots \ge w_{[i]} \ge \cdots \ge w_{[|C_l|]}$. Then there is a critical value h^*, $1 \le h^* \le |C_l|$, such that the optimal solution to the above problem is given by*

$$u^*_{lj} = \begin{cases} 1 & \text{if } w_j \ge w_{[h^*]} \text{ and } x \ge A^j, \\ 0 & \text{otherwise.} \end{cases}$$

Proof. Defining $\tilde{v}_{lj} := v_{lj} \mathbf{1}_{\{x \ge A^j\}} \forall j \in C_l$ in the maximization (26), the ranking procedure follows by applying Proposition 6 in Liu and van Ryzin (2008). The optimal policy $\tilde{u}_l^*$ is such that

$$\tilde{u}^*_{lj} = \begin{cases} 1 & \text{if } w_j \ge w_{[h^*]}, \\ 0 & \text{otherwise.} \end{cases}$$

It is trivial that this policy leads to the same objective value like using the policy u_l^* as defined above for the original preference vector v_l. □

We denote this policy by TISAC, where the "c" refers to the property that for each resource i inventory is split in as many ranges as we have capacity c_i. That means, TISAC is based on an entirely disaggregated approximation and is given in (25). In Section 8 we present numerical results for this policy, as well as for a so-called TISA2 policy. The latter relies on splitting the inventory of each resource i into two equal ranges and assuming that the marginal value of capacity $V_{t,i,a}$ and $V_{t,i,b}$ are constant across each range, respectively (see Figure 1). The policy TISA2 is defined in the same way as TISAC, except that the opportunity cost approximation is now $v_{t+1}(x) - v_{t+1}(x - A^j) \approx \sum_i \nu_i$, where ν_i is defined by

$$\nu_i = \begin{cases} a_{ij} V_{t+1,i,a}, & x_i \leq \lfloor c_i/2 \rfloor, \\ a_{ij} V_{t+1,i,b}, & x_i > \lfloor c_i/2 \rfloor, \end{cases} \text{ for all } i.$$

In the same manner, the solutions of (**CDLP**), (**AFF**) and (**ADLP**) can be used to construct policies based on the dual values of the corresponding capacity constraints. We call the resulting policies CDLP, AFF and ADLP, respectively.

7.2. Dynamic Programming Decomposition using CDLP

A popular method of solving network revenue management problems is to decompose them into a set of resource-level problems, that is for every resource i in the network we have one single leg problem with associated value function $v_t^i(x_i)$. One possible approach is to use the choice-based deterministic linear program which was introduced in Section 4.1: Given a resource i, we approximate the network value function by

$$v_t(x) \approx v_t^i(x_i) + \sum_{k \neq i} \pi_k^* x_k,$$

where π^* is the static vector of optimal bid prices obtained from (**CDLP**), that means the dual variables to the capacity constraints in (**CDLP**) at the optimal solution. We substitute this approximation into the dynamic programming formulation (1) and obtain a one-dimensional problem with displacement-adjusted revenues $f_j - \sum_{k \neq i} \pi_k^* a_{kj}$ which can be quickly solving by backwards induction. Having done that for all resources i, the network value function is then approximated by

$$v_t(x) \approx \sum_i v_t^i(x_i).$$

Again substituting this approximation into the Bellman equation (1) yields a maximization like in (26) but with $w_j := f_j - \sum_i (v_t^i(x_i) - v_t^i(x_i - a_{ij}))$. We call this policy DCDLP and refer to Liu and van Ryzin (2008) for a more detailed discussion of this approach. Through this procedure we obtain dynamic marginal capacity value estimates. However, their quality is based on the relatively

Figure 2 **Hub & Spoke network example HS-a.**

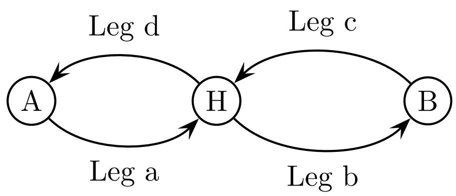
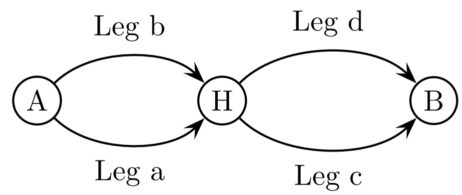


Figure 3 **Hub & Spoke network example HS-b.**



crude opportunity cost approximation in terms of π; the estimates resulting from our approach are already dynamic and approximate the opportunity cost better. For any aggregation of inventory levels, the resulting estimates can potentially also be improved by DP decomposition where we would need to solve a one-dimensional dynamic program for every resource. The corresponding Bellman equations share the same structure with the column generation sub-problem (7) so that we can solve them via mixed integer linear programs. Intuitively, the decomposition approach should exhibit better policy performance than DP decomposition based on the CDLP since the input of the decomposition procedure is more accurate. However, the finer the approximation the less improvement the decomposition seems to be able to achieve while even for the fully aggregated affine approach the results of Zhang and Adelman (2009) show little improvement of the decomposition approach based on (**AFF**) over AFF.

8. Numerical Results

In this section, we present the results of numerical experiments that shed light on the quality of the upper bounds and performance of policies obtained for our approach, compared with the above mentioned alternative approaches. We consider TISA with different aggregations. The rationale is that we intend to demonstrate the obtainable gains by splitting up the inventory while balancing the computational effort required to solve (**P**). Our numerical examples provide a framework of what improvements can be expected for approximations in between the demonstrated ones. All computations were carried out with MATLAB using CPLEX on a 3 GHz PC.

Problem Instances

We test our approach on two small networks called HS-a and HS-b, and on a somewhat larger network based on the so-called "Small Network Example" of Liu and van Ryzin (2008).

The first network HS-a is depicted in Figure 2 and represents a network with one hub and two non-hub nodes. There are four flight legs, and we assume that the segments are such that they consider all products with the same origin-destination (O-D) combination. In this case, we have six segments which correspond to the six possible O-D combinations. For each itinerary there are two products, a high fare class and a low fare class. Preference values were generated from the Poisson distribution with mean 80 for high fares, with mean 200 for low fares, and mean 10 for the no-purchase preference and are given in Table 2, which also provides an overview of products, divided into the disjoint consideration sets C_l for each segment l. Fares are likewise indicated; they were drawn from the Poisson distribution with mean for high and low fares on local flight 30 and 10, respectively, and for high and low fares on multi-leg itineraries 300 and 100, respectively. We use three arrival rates $\lambda = \sum_l \lambda_l$ to vary the load factor of the considered network instances to having a low, medium and high load factor, given in Table 4. The empirical load factor is obtainable by summing consumed capacity over the booking horizon for each sample path, averaging these numbers over all samples and dividing it by the total network capacity. Clearly, the load factor depends on the simulations, in particular with respect to the policies that were used. Since we intend to compare different policies under the same circumstances, we characterize the latter with the so-called *capacity tightness* instead of the empirical load factor. Capacity tightness is defined here as the total expected resource consumption of offering a specific set S^*, divided by the network capacity. In formulae,

$$\text{Capacity Tightness } \rho = \frac{\lambda \sum_{t=1}^{\tau} \sum_{j \in S^*} \sum_{i=1}^{m} a_{ij} P_j(S^*)}{\sum_{i=1}^{m} c_i},$$

where S^* is the revenue maximizing set given no capacity constraints,

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

The second network HS-b is depicted in Figure 3, and segments are described in Table 3 and Table 5. It consists of two parallel flights from location A to H, and further two parallel flights from H to B. On each itinerary we again have two fare classes high and low, and we assumed that segments correspond to O-D combinations. The fares and preference values are again taken from the Poisson distribution with mean as in the network HS-a. For both networks capacities are scaled up starting from $\hat{c} := [2, 4, 4, 2]$. We confine ourselves to very small network capacities since solution of the full-blown approach TISAC becomes quickly computationally expensive, yet it is of interest because it provides an excellent benchmark that can be used for testing other policies such as TISA2 and indicates the range of possible improvement due to inventory dependence. For practical implementations however, the aggregated approach must be used.

The final test network is displayed in Figure 4 and consists of 7 flight legs and 22 products. We took the product and customer segment definitions from Liu and van Ryzin (2008) and state them in Table 6 and Table 7. The time horizon comprises 200 periods and the capacity vector

Table 2 Products, Segments and Preference Values for HS-a

	C_1		C_2		C_3		C_4		C_5		C_6	
Prod. j	1	2	3	4	5	6	7	8	9	10	11	12
Segment	A → H		A → H → B		H → B		B → H		B → H → A		H → A	
Fare	30	12	294	97	39	10	26	10	289	121	25	10
Legs	a	a	a,b	a,b	b	b	c	c	c,d	c,d	d	d
Pref. v_l	72	198	76	203	89	200	74	228	87	209	87	214

Product definitions for network HS-a. "Legs" indicates the resources which the respective product utilizes. No-purchase preference $v_{l0} = [6, 14, 7, 6, 9, 7]$.

Table 3 Products, Segments and Preference Values for HS-b

	C_1				C_2								C_3			
Prod. j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Segment	A → H				A → H → B								H → B			
Fare	28	16	31	11	325	91	279	117	308	91	316	118	27	10	26	5
Legs	a	a	b	b	a,c	a,c	a,d	a,d	b,c	b,c	b,d	b,d	c	c	d	d
Pref. v_l	70	205	99	216	81	214	80	218	94	213	74	197	84	217	85	200

"Legs" indicates the resources which the respective product utilizes. No-purchase preference $v_{l0} = [3, 6, 14]$.

Table 4 HS-a: Arrival rates.

Seg. l	Low λ	Med. λ	High λ
1	0.0997	0.1189	0.1534
2	0.0605	0.0722	0.0932
3	0.0962	0.1147	0.1479
4	0.1033	0.1232	0.1589
5	0.0890	0.1062	0.1370
6	0.0712	0.0849	0.1096
Σ	0.52	0.62	0.8

HS-a: Arrival rates λ_l for each segment l, for the three considered cases of $\lambda \in \{0.52, 0.62, 0.8\}$.

Table 5 HS-b: Arrival rates.

Seg. l	Low λ	Med. λ	High λ
1	0.1327	0.1598	0.2051
2	0.1886	0.2271	0.2914
3	0.1187	0.1430	0.1835
Σ	0.44	0.53	0.68

HS-b: Arrival rates λ_l for each segment l, for the three considered cases of $\lambda \in \{0.44, 0.53, 0.68\}$.

is $c = [20, 30, 30, 30, 30, 15, 15]$. We consider various scenarios by scaling this capacity vector with a parameter $\alpha \in \{0.6, 0.8, 1, 1.2, 1.4\}$, and setting the no-purchase preference to 1,5 or 10 for the business segments, and 5, 10 or 20 for the leisure segments. For this so-called Small Network Example, we aggregate the inventory of each flight leg i in K_i ranges of uniform length, where $K := [4, 2, 2, 2, 2, 1, 1]$. We chose this aggregation because the dual values of the capacity constraints of CDLP indicated that leg 1 is typically the most valuable resource, and legs 6 and 7 are the least

Figure 4 **Small Network example.**

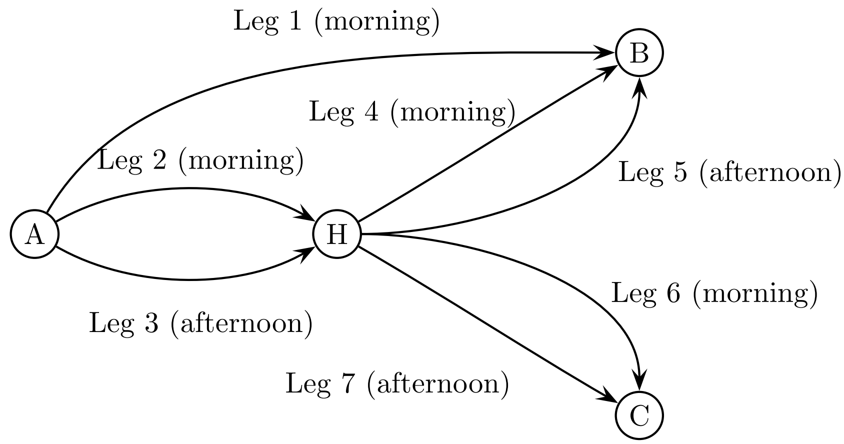


Table 6 **Small Network example**

Product	Legs	Class	Fare	Product	Legs	Class	Fare
1	1	H	1000	12	1	L	500
2	2	H	400	13	2	L	200
3	3	H	400	14	3	L	200
4	4	H	300	15	4	L	150
5	5	H	300	16	5	L	150
6	6	H	500	17	6	L	250
7	7	H	500	18	7	L	250
8	2,4	H	600	19	2,4	L	300
9	3,5	H	600	20	3,5	L	300
10	2,6	H	700	21	2,6	L	350
11	3,7	H	700	22	3,7	L	350

Product definitions.

valuable ones for the considered scenarios. We refer to the LP resulting from using this aggregation in (**P**) and to the corresponding direct opportunity cost policy as TISAK; which one is meant should be clear from the context.

Upper Bound Quality

Upper bounds are useful as benchmarks in simulation studies and also potentially in designing new policies: For example, the approach of Siddappa et al. (2007) uses upper and lower bounds on the value function to construct policies. As stated in Proposition 1, the optimal objective value to (**P**) constitutes an upper bound on the optimal expected revenue and, in particular, the bound is at least as good as the one provided by the affine approximation approach. The natural question arises whether the new bound might turn out to be identical to the latter, or, if there is improvement,

Table 7 Small Network example

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

Segment definitions.

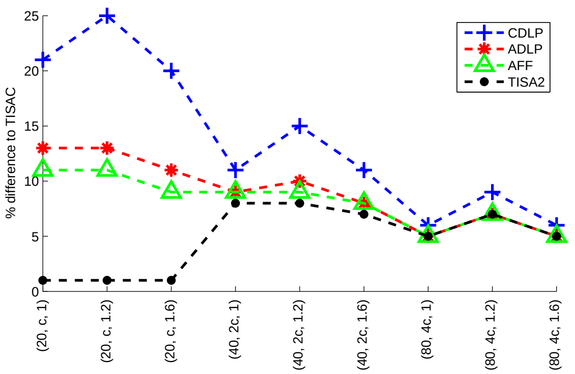
how much more accuracy was gained. We address this issue by comparing the upper bounds of the different solution approaches, all of which being applied to the problem instances as described above. All approaches were implemented using column generation. As stopping criterion for the column generation procedure we used the following "x% tolerance criterion": Stop generating columns if the sum over all time periods of the maximum reduced cost of each time period, say we denote it by $\mathcal{S}$, is within x% of objective value of the restricted master problem plus $\mathcal{S}$. The CDLP is solved to optimality, for TISAC we used the 1% and for all other approaches the 0.5% stopping criterion.

We solve the problem HS-a and HS-b with CDLP, ADLP and AFF, and compare their corresponding upper bounds with our two-ranges approach TISA2 and the individual seat-level approach TISAC. Tables 15 and 16 highlight the percentage improvement of TISAC relative to the other approaches over several problem instances. The highest gains in accuracy are observed for medium load factors, which is intuitive since very low load factors imply simply offering the unconstrained revenue maximizing set, and for very high load factors one would simply offer the highest fares. For network HS-b, however, the improvement converges quickly to only 1% over any of the other approaches. In all cases we can observe the following ordering of the arising bounds:

$$z_{\text{CDLP}} \geq z_{\text{ADLP}} \geq z_{\text{AFF}} \geq z_{\text{TISA2}} \geq z_{\text{TISAC}}.$$

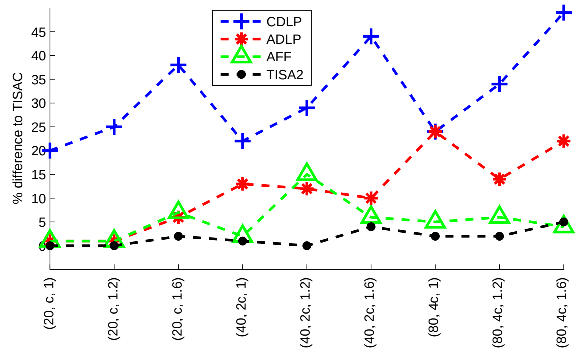
Small deviations from this ordering can occur due to stopping the column generation procedure according to the above mentioned tolerance criterion. This demonstrates that the bounds obtainable from the concave approach are indeed improvements. As we increase the time horizon and the leg capacity, the improvements are reduced but still are at least 5% compared to AFF in network

Figure 5 HS-a: Bound improvement.



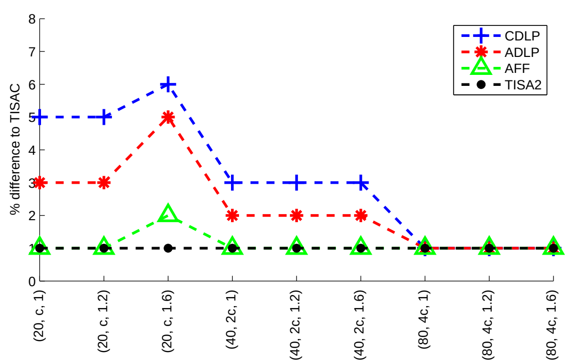
Note. Problem instances are described by (time horizon, capacity vector, load factor).

Figure 6 HS-a: Policy improvement.



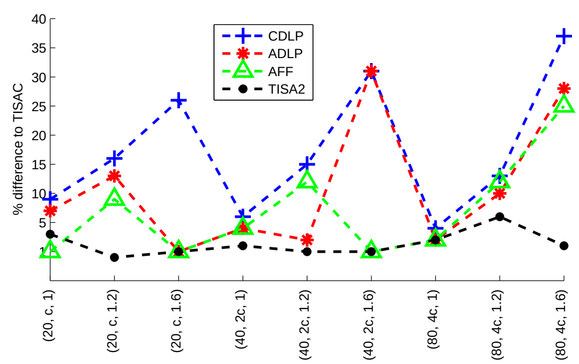
Note. Problem instances are described by (time horizon, capacity vector, load factor).

Figure 7 HS-b: Bound improvement .



Note. Problem instances are described by (time horizon, capacity vector, load factor).

Figure 8 HS-b: Policy improvement.



Note. Problem instances are described by (time horizon, capacity vector, load factor).

HS-a. This decreasing difference can be explained with the asymptotic behavior of all approaches, that means, they all approach the optimal expected revenue as time and capacity are scaled up.

The Small Network Example gives some insight into the behavior on somewhat larger networks. For this example, we solved CDLP to optimality while for AFF and TISAK we used the 1% stopping criterion. The results reported in Table 8 show that AFF and TISAK provide almost identical bounds. Keep in mind that these values do not always satisfy $z_{\text{AFF}} \geq z_{\text{TISAK}}$ as we would expect because we did not solve to optimality. All bounds are identical if capacity is ample.

Policy Performance

We claim that our proposed approach yields better opportunity cost estimates through a better approximation of the value function, and support this by numerical results that were the outcome

Table 8 Upper bounds for Small Network Example.

α	v_0	z_{CDLP}	z_{AFF}	z_{TISAK}	$\%z_{\text{AFF}} - z_{\text{CDLP}}$	$\%z_{\text{TISAK}} - z_{\text{CDLP}}$
0.6	(1,5)	36,187	35,796	35,775	-1.08	-1.14
0.6	(5,10)	33,158	32,654	32,728	-1.52	-1.30
0.6	(10,20)	29,960	29,524	29,531	-1.46	-1.43
0.8	(1,5)	43,202	42,832	42,862	-0.86	-0.79
0.8	(5,10)	38,900	38,494	38,551	-1.04	-0.90
0.8	(10,20)	34,678	34,347	34,403	-0.95	-0.79
1.0	(1,5)	48,822	48,497	48,496	-0.67	-0.67
1.0	(5,10)	43,767	43,407	43,417	-0.82	-0.80
1.0	(10,20)	35,103	35,100	35,101	-0.01	-0.01
1.2	(1,5)	53,564	53,249	53,238	-0.59	-0.61
1.2	(5,10)	44,690	44,686	44,637	-0.01	-0.12
1.2	(10,20)	35,103	35,103	35,102	-0.00	-0.00
1.4	(1,5)	55,257	55,068	55,084	-0.34	-0.31
1.4	(5,10)	44,690	44,687	44,640	-0.01	-0.11
1.4	(10,20)	35,103	35,102	35,102	-0.00	-0.00

CDLP was solved to optimality, AFF and TISAK were stopped with the 1% criterion.

of using the opportunity cost information obtained from the various LP approaches directly to construct policies as described in Section 7.

We tested the following policies:

- CDLP: Policy with static opportunity cost estimate (3) based on the optimal dual values of CDLP.
- DCDLP: Dynamic programming decomposition policy based on CDLP as explained in Section 7.2.
- ADLP: Policy with opportunity cost estimate (4) based on the optimal dual values of ADLP.
- AFF: Policy with opportunity cost estimate (5) based on the optimal dual values of AFF.
- TISA2, TISAC and TISAK: Policies with opportunity cost estimates based on an inventory split in 2, c_i and k_i parts for all i, respectively, as detailed in Section 7.
- DTISAK: Dynamic programming decomposition policy based on TISA with inventory split into $K = [4, 2, 2, 2, 2, 1, 1]$ range for the Small Network Example.

The booking process was simulated for each network instance by generating customer arrivals, letting the respective policy decide on which set of products to offer, and simulating customer choice decision based on the MNL choice model. We record for each run the achieved revenue, average it over the entire sample and use the resulting value to measure and compare policy performance.

In Figures 6 and 8 we summarized the outcome of our simulation study comparing TISAC with the alternative approaches, the underlying data can be found in Tables 17 and 18. The relative errors of the simulations are at most 0.8% with 99% confidence. We find that the static marginal capacity value estimates do perform quite poorly as expected, and that ADLP and AFF show improved results because they incorporate time-dependent estimates. All approaches are outperformed by TISAC and even TISA2, the latter with the exception of one problem instance in HS-b. In particular, TISA2 works already very well despite only using two marginal values per resource per time step, which indicates that some aggregation of inventory levels to enhance the computational performance will not necessarily severely deteriorate policy performance relative to TISAC.

Table 9 Simulation results for Small Network Example.

α	v_0	DCDLP	LF	DTISAK	LF	TISAK	LF	AFF	LF	%DTISAK − DCDLP	%TISAK − DCDLP	%AFF − DCDLP
0.6	(1,5)	34,045	0.95	33,160	0.94	34,073	0.94	31,720	0.78	-2.60	0.08	-6.83
0.6	(5,10)	30,969	0.92	30,494	0.92	30,667	0.90	30,558	0.93	-1.53	-0.97	-1.33
0.6	(10,20)	27,974	0.88	27,699	0.89	27,966	0.89	26,770	0.87	-0.98	-0.03	-4.30
0.8	(1,5)	41,143	0.93	40,477	0.94	39,086	0.86	37,514	0.88	-1.62	-5.00	-8.82
0.8	(5,10)	37,049	0.89	36,720	0.90	37,036	0.92	35,799	0.89	-0.89	-0.03	-3.37
0.8	(10,20)	32,671	0.82	32,634	0.82	32,473	0.80	32,574	0.81	-0.11	-0.61	-0.30
1.0	(1,5)	47,107	0.91	46,693	0.91	44,876	0.85	43,652	0.83	-0.88	-4.74	-7.33
1.0	(5,10)	41,854	0.85	41,735	0.85	41,897	0.86	41,129	0.84	-0.29	0.10	-1.73
1.0	(10,20)	34,589	0.71	34,589	0.71	34,645	0.72	34,645	0.72	0.00	0.16	0.16
1.2	(1,5)	51,828	0.86	51,658	0.86	51,633	0.86	50,364	0.85	-0.33	-0.38	-2.82
1.2	(5,10)	44,091	0.74	44,091	0.74	44,058	0.74	44,058	0.74	0.00	-0.07	-0.07
1.2	(10,20)	34,969	0.61	34,969	0.61	34,982	0.61	34,982	0.61	0.00	0.04	0.04
1.4	(1,5)	54,308	0.76	54,308	0.76	54,362	0.77	54,360	0.77	-0.00	0.10	0.10
1.4	(5,10)	44,534	0.64	44,534	0.64	44,535	0.64	44,535	0.64	0.00	0.00	0.00
1.4	(10,20)	35,014	0.52	35,014	0.52	35,015	0.52	35,015	0.52	0.00	0.00	0.00

LF: empirical average load factor. %A − B: percentage difference between policy A and B.

We test whether the policies will still perform as strongly when applied to the Small Network Example. CDLP with DP decomposition (DCDLP) sets the benchmark against which we compare the average revenue results from the simulation runs.For all 15 test scenarios, we run 5000 simulations with the policies DCDLP, DTISAK, TISAK and AFF, respectively. Table 9 displays the results, and Table 10 the corresponding relative percentage errors with 95% confidence. TISAK is in all scenarios better than AFF and reaches in most cases the benchmark DCDLP. In only 3

Table 10 Relative percentage errors of simulation results for Small Network Example.

α	v_0	DCDLP	DTISAK	TISAK	AFF
0.6	(1,5)	0.12	0.11	0.15	0.22
0.6	(5,10)	0.14	0.13	0.16	0.17
0.6	(10,20)	0.15	0.14	0.14	0.20
0.8	(1,5)	0.16	0.14	0.21	0.18
0.8	(5,10)	0.17	0.16	0.16	0.22
0.8	(10,20)	0.18	0.18	0.19	0.19
1.0	(1,5)	0.19	0.18	0.21	0.26
1.0	(5,10)	0.19	0.19	0.19	0.21
1.0	(10,20)	0.24	0.24	0.24	0.24
1.2	(1,5)	0.20	0.20	0.19	0.26
1.2	(5,10)	0.25	0.25	0.25	0.25
1.2	(10,20)	0.26	0.26	0.26	0.26
1.4	(1,5)	0.23	0.23	0.24	0.24
1.4	(5,10)	0.27	0.27	0.27	0.27
1.4	(10,20)	0.26	0.26	0.26	0.26

Relative percentage error with 95% confidence of simulation results, sample size 5000.

Table 11 CPU time for Small Network Example.

α	v_0	AFF (h)	TISAK (h)	TISAK/AFF
0.6	(1,5)	0.20	14.06	69.3
0.6	(5,10)	0.16	11.78	71.5
0.6	(10,20)	0.15	10.57	71.0
0.8	(1,5)	0.13	9.73	75.2
0.8	(5,10)	0.11	9.21	80.2
0.8	(10,20)	0.09	6.99	74.2
1.0	(1,5)	0.15	8.11	55.4
1.0	(5,10)	0.12	5.94	48.8
1.0	(10,20)	0.06	3.40	55.2
1.2	(1,5)	0.13	6.76	51.7
1.2	(5,10)	0.08	3.51	42.7
1.2	(10,20)	0.07	3.19	46.2
1.4	(1,5)	0.11	8.09	74.4
1.4	(5,10)	0.07	4.43	67.5
1.4	(10,20)	0.06	4.46	75.0

On average, run time increases with a factor of 63.9.

scenarios is TISAK 1% or more under the benchmark. DTISAK offers some improvement in the worst scenarios of TISAK, but delivers significantly deteriorated results in others. The reason for this behavior is probably an amplification effect of the inexact input resulting from not having solved TISAK to optimality. We can see from the empirical load factors that AFF tends to be too restrictive in the scenarios of its worst performance. This is intuitive because it tends to overestimate the opportunity cost. The improvement in revenue performance of TISAK appears impressive given the relatively coarse inventory aggregation of 1-4 ranges.

Computational Performance

The linear program (**P**) has $(\tau + \tau \sum_i K_i)$ constraints. Computational workload for solving a linear program grows proportionally to the number of constraints to the power of three (Bradley et al. (1977), p. 364), thus considering every inventory level separately on all resources –which corresponds to $K_i = c_i$ for all i– will be expensive. As an example for the grow of computational workload, we observed that solving TISA2 takes about 5 times as long as solving AFF, see Table 14. Let us investigate how the marginal value of capacity actually varies across the inventory: In our numerical experiments, the difference between marginal values of capacity is small if the remaining time to departure is large relative to the capacity. To exemplify this observation, consider the contour plot in Figure 9. For the first 50 time periods, the marginal capacity values are almost constant over large inventory level ranges. Only in the last 30 time periods, the decline becomes more pronounced as it can be seen from the contour lines moving together. This can be intuitively explained by noting that these marginal values depend on the probability that we can sell all the seats up to the corresponding inventory level, and if the number of remaining time periods is large relative to the capacity, the probabilities should not differ very much. Therefore, first solving a large problem with a high level of aggregation and later resolving with refined approximation should be advantageous. Initially, an aggregation might be chosen with inventory ranges of equal length and large enough such that the model is still tractable. Having obtained a solution, we can guide the aggregation in the resolving process by examining the relative differences between resulting marginal values V_{t,i,k_l} and V_{t,i,k_r} of adjacent pairs of inventory ranges (k_l, k_r). If the difference between these values is greater than some specified threshold ϵ, then we could halve both ranges k_l and k_r so that in the next resolving process the change in the slope of the value function can be better represented in the approximation. On the other hand, if $|V_{t,i,k_l} - V_{t,i,k_r}| < \epsilon$, we would conclude that the value function is close to being linear in this area and we do not refine the approximation. In fact, we might even want to merge such two ranges into one to save computational effort. Also, we can exploit the flexibility of our model to approximate different flights with different levels of aggregation; those with low load factor will not need a fine approximation, and can thus be aggregated to enhance computational performance. Such legs could be identified by finding flights i that have $\pi_i = 0$ in the optimal CDLP dual solution.

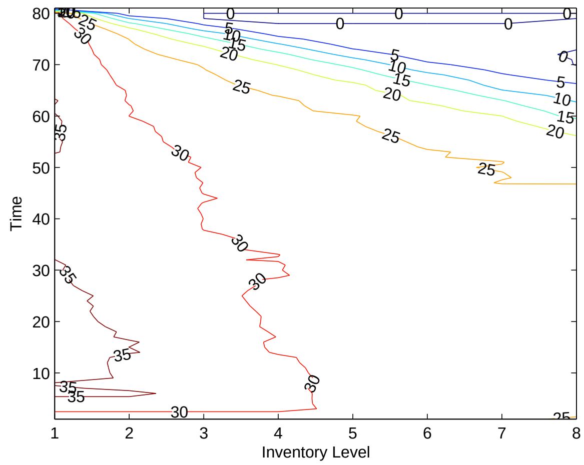
For the Small Network Example, CPU times for solving the respective LPs can be found in Table 11. The run time increases dramatically on average by a factor of 64. This can be attributed to the column generation process and imply that the approach indeed needs high aggregation to make it feasible. In fact, aggregation of time periods appears to be an attractive way to keep the computational burden acceptable. Note that the absolute run times, as always, carry little meaning since they always depend the programming language, the hardware and the skill of the programmer. It can also be reduced by using heuristic for the column pricing as proposed in Meissner and Strauss (2009). While the CDLP can be solved relatively quickly, the DP decomposition is likewise computationally intensive. In particular, its run time increases with the leg capacities, unlike TISAK for a fixed number of ranges per leg.

9. Conclusion and Future Research

In the context of quantity-based network revenue management, we presented a linear programming approach to approximate dynamic programming with nonlinear approximation of the value function with the specific feature that it incorporates both customer choice behavior as well as estimates of marginal capacity values that depend on time and resource inventory level. As a result of the improved approximation, we obtain a better estimate of the opportunity cost, which is reflected in provably tighter upper bounds for any inventory aggregation and improved policy performance as observed in simulation studies. A policy based on the opportunity cost estimates obtained directly from an approximate solution of our linear program using column generation outperforms alternative approaches. The solution of the linear program can be expensive, hence we propose to trade off accuracy with computational workload by aggregating inventory levels at the beginning of the booking horizon and later re-solving with refined inventory level resolution.

More research is needed on the question of how to aggregate inventory without losing too much accuracy. A promising way might be to re-solve the linear program several times over the booking horizon with a process that guides the structure of inventory aggregations towards refinement where the value function exhibits non-linearity and aggregation where it is close to being linear. Such a process could be implemented by examining the difference in marginal values between every pair of adjacent inventory ranges and refining the ranges if this difference is large, or merging them into a single range if not. In addition, aggregation of time steps is also possible and can be used to reduce computational effort by exploiting that typically the value function at the beginning of the booking horizon is close to being linear.

Figure 9 Contour plot of marginal value of capacity for leg d in network HS-b.



Note. Departures at time period $\tau = 81$, capacity of this resource is 8, capacity tightness $\rho = 1.2$.

Table 12 Simulation results for CDLP with dynamic programming decomposition on HS-a.

	τ	c	DP-CDLP	RE	TISAC	RE	$\frac{\text{TISAC}}{\text{DP-CDLP}}$
Low LF ($\rho = 1$)	20	$\hat{c}$	674	0.7	738	0.6	1.09
	40	$2\hat{c}$	1577	0.7	1627	0.8	1.03
	80	$4\hat{c}$	3422	0.6	3450	0.6	1.01
Med LF ($\rho = 1.2$)	20	$\hat{c}$	750	0.7	820	0.6	1.09
	40	$2\hat{c}$	1763	0.6	1816	0.7	1.03
	80	$4\hat{c}$	3859	0.5	3907	0.5	1.01
High LF ($\rho = 1.6$)	20	$\hat{c}$	895	0.7	940	0.7	1.05
	40	$2\hat{c}$	2004	0.5	2090	0.5	1.04
	80	$4\hat{c}$	4285	0.4	4389	0.4	1.02

TISAC was implemented as direct opportunity cost estimate policy. RE is the percentage relative error of the sample mean with 99% confidence. The constant vector $\hat{c}$ is defined as $\hat{c} := [2, 4, 4, 2]$..

Table 13 Simulation results for CDLP with dynamic programming decomposition on HS-b.

	τ	c	DP-CDLP	RE	TISAC	RE	$\frac{\text{TISAC}}{\text{DP-CDLP}}$
Low LF ($\rho=1$)	20	$\hat{c}$	990	0.6	1144	0.7	1.16
	40	$2\hat{c}$	2240	0.6	2378	0.7	1.06
	80	$4\hat{c}$	4771	0.4	4866	0.5	1.02
Med LF ($\rho=1.2$)	20	$\hat{c}$	1212	0.5	1327	0.6	1.09
	40	$2\hat{c}$	2677	0.6	2783	0.7	1.04
	80	$4\hat{c}$	5639	0.5	5648	0.5	1.00
High LF ($\rho=1.6$)	20	$\hat{c}$	1399	0.4	1588	0.5	1.14
	40	$2\hat{c}$	3131	0.4	3321	0.5	1.06
	80	$4\hat{c}$	6666	0.3	6878	0.4	1.03

TISAC was implemented as direct opportunity cost estimate policy. RE is the percentage relative error of the sample mean with 99% confidence. The constant vector $\hat{c}$ is defined as $\hat{c} := [2,4,4,2]$..

Table 14 CPU run time in seconds

	τ	c	AFF (s)	TISA2 (s)	$\frac{\text{TISA2}}{\text{AFF}}$
Low LF ($\rho=1$)	20	$\hat{c}$	4.3	27	6.3
	40	$2\hat{c}$	8.7	41	4.7
	80	$4\hat{c}$	15.6	73	4.7
Med LF ($\rho=1.2$)	20	$\hat{c}$	4.5	24	5.3
	40	$2\hat{c}$	9	49	5.4
	80	$4\hat{c}$	18.8	84	4.5
High LF ($\rho=1.6$)	20	$\hat{c}$	5.1	18	3.5
	40	$2\hat{c}$	9.7	41	4.2
	80	$4\hat{c}$	22.8	115	5.0

CPU run times to solve problem HS-a with **AFF** and **TISA** for the approach with two marginal values per resource per time step. The constant vector $\hat{c}$ is defined as $\hat{c} := [2,4,4,2]$.

Table 15 Upper Bounds for Network HS-a

	τ	c	z_{CDLP}	z_{ADLP}	z_{AFF}	z_{TISA2}	z_{TISAC}
Low LF ($\rho = 1$)	20	$\hat{c}$	925	866	851	775	766
	40	$2\hat{c}$	1850	1808	1803	1788	1661
	80	$4\hat{c}$	3701	3658	3655	3653	3488
Med LF ($\rho = 1.2$)	20	$\hat{c}$	1077	978	962	877	864
	40	$2\hat{c}$	2154	2065	2050	2026	1878
	80	$4\hat{c}$	4307	4234	4219	4214	3953
High LF ($\rho = 1.6$)	20	$\hat{c}$	1200	1102	1086	1008	997
	40	$2\hat{c}$	2400	2333	2321	2299	2153
	80	$4\hat{c}$	4800	4743	4742	4738	4529

Upper bounds on optimal expected revenue from HS-a. The constant vector $\hat{c}$ is defined as $\hat{c} := [2, 4, 4, 2]$.

Table 16 Upper Bounds for Network HS-b

	τ	c	z_{CDLP}	z_{ADLP}	z_{AFF}	z_{TISA2}	z_{TISAC}
Low LF ($\rho = 1$)	20	$\hat{c}$	1293	1273	1250	1243	1235
	40	$2\hat{c}$	2587	2565	2548	2548	2523
	80	$4\hat{c}$	5173	5147	5137	5140	5109
Med LF ($\rho = 1.2$)	20	$\hat{c}$	1495	1472	1447	1440	1430
	40	$2\hat{c}$	2990	2964	2943	2939	2917
	80	$4\hat{c}$	5980	5950	5930	5930	5897
High LF ($\rho = 1.6$)	20	$\hat{c}$	1817	1795	1746	1736	1715
	40	$2\hat{c}$	3633	3609	3580	3573	3537
	80	$4\hat{c}$	7266	7240	7220	7220	7170

Upper bounds on optimal expected revenue from HS-b. The constant vector $\hat{c}$ is defined as $\hat{c} := [2, 4, 4, 2]$.

Table 17 **Simulation results for direct opportunity cost estimate policies on network instances HS-a.**

	τ	c	CDLP	RE	ADLP	RE	AFF	RE	TISA2	RE	TISAC	RE
Low LF ($\rho = 1$)	20	$\hat{c}$	727	0.7	732	0.7	732	0.7	738	0.6	738	0.6
	40	$2\hat{c}$	1594	0.8	1445	0.8	1591	0.8	1607	0.8	1627	0.8
	80	$4\hat{c}$	3364	0.7	2772	0.6	3297	0.6	3398	0.6	3450	0.6
Med LF ($\rho = 1.2$)	20	$\hat{c}$	743	0.6	816	0.6	815	0.6	820	0.6	820	0.6
	40	$2\hat{c}$	1621	0.6	1622	0.7	1575	0.8	1814	0.7	1816	0.7
	80	$4\hat{c}$	3414	0.5	3421	0.5	3693	0.6	3830	0.6	3907	0.5
High LF ($\rho = 1.6$)	20	$\hat{c}$	721	0.6	886	0.8	882	0.8	917	0.8	940	0.7
	40	$2\hat{c}$	1534	0.6	1893	0.6	1981	0.6	2001	0.6	2090	0.5
	80	$4\hat{c}$	3185	0.4	3598	0.4	4213	0.5	4186	0.5	4389	0.4

RE is the percentage relative error of the sample mean with 99% confidence. The constant vector $\hat{c}$ is defined as $\hat{c} := [2, 4, 4, 2]$.

Table 18 **Simulation results for direct opportunity cost estimate policies on network instances HS-b.**

	τ	c	CDLP	RE	ADLP	RE	AFF	RE	TISA2	RE	TISAC	RE
Low LF ($\rho = 1$)	20	$\hat{c}$	1053	0.7	1072	0.7	1146	0.8	1112	0.7	1144	0.7
	40	$2\hat{c}$	2242	0.7	2280	0.7	2294	0.7	2346	0.7	2378	0.7
	80	$4\hat{c}$	4693	0.5	4753	0.5	4760	0.5	4764	0.5	4866	0.5
Med LF ($\rho = 1.2$)	20	$\hat{c}$	1148	0.6	1175	0.6	1222	0.6	1338	0.7	1327	0.6
	40	$2\hat{c}$	2419	0.5	2742	0.8	2495	0.5	2783	0.8	2783	0.7
	80	$4\hat{c}$	5003	0.4	5149	0.5	5033	0.4	5316	0.5	5648	0.5
High LF ($\rho = 1.6$)	20	$\hat{c}$	1260	0.5	1585	0.5	1585	0.5	1586	0.5	1588	0.5
	40	$2\hat{c}$	2531	0.5	2542	0.5	3313	0.5	3315	0.5	3321	0.5
	80	$4\hat{c}$	5021	0.4	5361	0.3	5491	0.3	6832	0.4	6878	0.4

RE is the percentage relative error of the sample mean with 99% confidence. The constant vector $\hat{c}$ is defined as $\hat{c} := [2, 4, 4, 2]$.

References

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

Belobaba, P. P., C. Hopperstad. 1999. Boeing/MIT simulation study: PODS results update. *1999 AGIFORS Reservations and Yield Management Study Group Symposium Proceedings*.

Belobaba, P. P., L. R. Weatherford. 1996. Comparing decision rules that incorporate customer diversion in perishable asset revenue management situations. *Decision Sciences* **27**(2) 343–363.

Bertsekas, D. P., J. N. Tsitsiklis. 1996. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA.

Boyd, E. A., R. Kallesen. 2004. The science of revenue management when passengers purchase the lowest available fare. *Journal of Revenue and Pricing Management* **3** 2.

Boyd, S. P., L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

Bradley, S.P., A.C. Hax, T.L. Magnanti. 1977. *Applied mathematical programming*. Addison-Wesley.

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

Camm, J.D., A.S. Raturi, S. Tsubakitani. 1990. Cutting Big M down to size. *Interfaces* **20** 61–66.

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.

de Farias, D. P., B. Van Roy. 2004. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research* **29** 462–478.

de Farias, D.P., B. Van Roy. 2003. The linear programming approach to approximate dynamic programming. *Operations Research* **51** 850–865.

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.

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. Choice-based network revenue management under weak market segmentation. 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.

Siddappa, S., D. Günther, J. Rosenberger, V. Chen. 2007. Refined experimental design and regression splines method for network revenue management. *Journal of Revenue and Pricing Management* **6** 188–199.

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*.

van Ryzin, G. J. 2005. Models of demand. *Journal of Revenue and Pricing Management* **4** 204–210.

van Ryzin, G. J., G. Vulcano. 2008. Computing virtual nesting controls for network revenue management under customer choice behavior. *Manufacturing & Service Operations Management* **10** 448–467.

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`.

Zhang, D., W. Cooper. 2005. Revenue management for parallel flights with customer choice behavior. *Operations Research* **53** 415–431.