A Dynamic Disaggregation Approach to Approximate Linear Programs for Network Revenue Management*

Thomas W.M. Vossen, Dan Zhang
Leeds School of Business, University of Colorado at Boulder
UCB 419, 995 Regent Dr, Boulder, CO 80309
{Thomas.Vossen,Dan.Zhang}@Colorado.edu

The linear programming approach to approximate dynamic programming has received considerable attention in the recent network revenue management literature. A major challenge of the approach lies in solving the resulting approximate linear programs (ALPs), which often have a huge number of constraints and/or variables. Starting from a recently developed compact affine ALP for network revenue management, we develop a novel dynamic disaggregation algorithm to solve the problem, which combines column and constraint generation and exploits the structure of the underlying problem. We show that the formulation can be further tightened by considering structural properties satisfied by an optimal solution. We prove that the sum of dynamic bid-prices across resources is concave over time. We also give a counterexample to demonstrate that the dynamic bid-prices of individual resources are not concave in general. Numerical experiments demonstrate that dynamic disaggregation is often orders of magnitude faster than existing algorithms in the literature for problem instances with and without choice. In addition, adding the concavity constraints can further speed up the algorithm, often by an order of magnitude, for problem instances with choice.

Key words: network revenue management; choice behavior; dynamic programming; linear programming
This version: Received: September 5, 2012; Revisions received: April 3, 2013, July 23, 2013, January 13, 2014; Accepted: March 30, 2014

1. Introduction

Network revenue management (RM) deals with the revenue maximization problem that arises when matching the availability of different products utilizing a network of resources with stochastic sequential demand over time. The problem can be formulated as a dynamic program, where the state is the vector of available resources and the decision in each period is whether to accept or reject each incoming customer request. Classical formulations of network RM (Gallego and van Ryzin 1997, Talluri and van Ryzin 1998) consider independent demand models where each customer requests a specific product, while more recent developments consider choice-based demand models where customers choose among open fare products (Talluri and van Ryzin 2004, Zhang and Cooper 2005, Gallego et al. 2004, Liu and van Ryzin 2008).

* Authors are listed alphabetically. Both authors contributed equally.

In the canonical airline application of network RM, each resource corresponds to a flight leg while each product corresponds to a fare-itinerary combination. Because airline networks often consist of tens or even hundreds of flight legs, however, the dynamic programming formulation suffers from the well-known "curse of dimensionality." Much of the existing literature therefore considers approximate solution approaches or heuristic control policies.

Recently, the linear programming based approximate dynamic programming (LP-based ADP) has received considerable attention in this stream of literature. Adelman (2007) introduces a solution framework based on an equivalent linear programming formulation of the dynamic program. The central idea is to approximate the value function with linearly weighted basis functions. He implements an affine approximation where the value function is approximated by an affine function of the state (resource vector). The resulting approximate linear program (ALP) has a relatively small number of variables but a huge number of constraints, growing exponentially in the number of resources and the number of products. A column generation procedure was proposed to solve the dual of the ALP, where the subproblems are linear mixed integer programs. The coefficients of each resource in this affine approximation can be interpreted as time-dependent dynamic bid-prices, which produce a control policy superior to the static bid-prices obtained from a widely used deterministic linear program (DLP) (Talluri and van Ryzin 1998, Cooper 2002). In particular, Adelman (2007) compares dynamic bid-price policy with a version of static bid- price policy with frequent DLP resolve and show that it performs much better, even though the latter has strong asymptotic performance guarantee (Jasin and Kumar 2012).

The column generation procedure in Adelman's work poses a potentially significant computational burden because a linear mixed integer program needs to be solved repeatedly to find entering columns. The same drawback is shared by much of the subsequent work in this area. In particular, Zhang and Adelman (2009) generalize Adelman's affine approximation approach to the choice-based network RM. Their solution approach employs a column generation procedure which also requires solving a linear mixed integer program to find an entering column.

We are interested in more efficient solution approaches for the ALPs in Adelman (2007) and Zhang and Adelman (2009) for network RM with and without customer choice, respectively. We start with the compact ALPs introduced in Vossen and Zhang (2013) for network RM both with and without customer choice. The compact ALPs have much smaller number of decision variables and constraints compared with the corresponding formulations in Adelman (2007) and Zhang and Adelman (2009), but produce the same objective values, and therefore can be viewed as proper reductions of the original formulations. The reductions introduced in Vossen and Zhang (2013)

generalize the earlier work of Tong and Topaloglu (2011), who consider the reduction for affine ALP of network RM without choice. We propose a *dynamic disaggregation* method to solve the time-disaggregated DLP. This approach starts with the DLP solution and successively adds both columns and constraints until optimality is reached. We test the method on network revenue management problem instances with up to 40 resources and 2000 periods.

For problem instances that do not involve customer choice, we find that the dynamic disaggregation procedure can be much more efficient than the column generation approach in Adelman (2007) applied to the affine ALP and the constraint generation approach in Tong and Topaloglu (2011) applied to the corresponding reduced program. Our numerical study shows that the dynamic disaggregation approach generally takes a fraction of the time required by Tong and Topaloglu (2011). We attribute the stronger performance of the dynamic disaggregation procedure to exploitation of the underlying structure of the problem, in particular, that dynamic bid-prices tend to stay constant over a significant portion of the booking horizon (Adelman 2007). The procedure also starts at a much stronger initial feasible solution which corresponds to the DLP solution. For problem instances with customer choice, we compare dynamic disaggregation to the column generation procedure from Zhang and Adelman (2009). We show that dynamic disaggregation can be orders of magnitude faster on randomly generated problem instances, echoing the same observation in the no-choice setting.

Aggregation/disaggregation is a powerful idea in solving large-scale linear programs; see Rogers et al. (1991) for a relevant review. The compact ALPs can be viewed as aggregated versions of the corresponding ALPs. While the literature often focuses on aggregation as an approximation method and studies error bounds of such approximations, we show that aggregation can be done for the ALPs without loss of optimality.

Column (or constraint) generation has been a popular method for solving ALPs, including the aforementioned work of Adelman (2007), Zhang and Adelman (2009), Tong and Topaloglu (2011); see also Meissner and Strauss (2012). The dynamic disaggregation approach we propose is distinctive in that it combines column and constraint generation, and effectively is an iterative aggregation/disaggregation procedure. Our solution approach has some precedence in the mathematical programming and dynamic programming literature (Vakhutinsky et al. 1979, Schweitzer et al. 1985), but seems to have received little attention in recent research.

One way to speed up computation of ALPs is tightening the formulation. Indeed, this was considered in the original work of Adelman (2007). He shows that there exist optimal dynamic bid-prices that are monotonically decreasing over time, and reports an average speedup factor of

86 by enforcing these monotonicity constraints. His work provides motivation to look for additional structural properties that can be exploited to tighten ALP formulations. Figure 1 originally appeared in Adelman (2007) as an example of dynamic bid-prices for a network RM problem with four resources. With the exception of a kink for the bid-price of resource 3, the dynamic bid-prices appear to be concave in time. Other researchers also report similar figures in their work (see, e.g., Zhang and Adelman 2009, Kirshner and Nediak 2012).

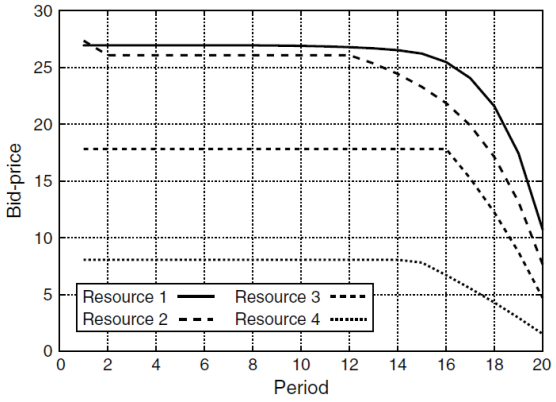

Figure 1 Example of dynamic bid-prices, originally appeared as Figure 1 in Adelman (2007)

If optimal dynamic bid-prices are concave, then we can not only enforce monotonicity constraints, but also enforce concavity constraints requiring the time difference of dynamic bid-prices to be monotone. Unfortunately, we are able to construct a counterexample to show that dynamic bid-prices of individual resources are not concave in general. However, we are able to show a weaker property that the sum of dynamic bid-prices across all resources is indeed concave. Furthermore, employing a quasi-static affine approximation where bid-prices are not time-dependent, we can show that the optimality gap by enforcing concavity of dynamic bid-prices of individual resources is small as the problem grows large. This is fairly intuitive, since it is known that both the quasi-static affine approximation and affine approximation approach the objective value of DLP as the problem size increases.

Enforcing the concavity constraints of individual resources leads to dramatic improvement in computational performance for many problem instances, especially ones with choice. For problem instances without choice, including the ones tested in Section 6.1, adding concavity constraints can sometimes increase the computational time. We note that adding concavity constraints introduces a trade-off in terms of computational efficiency. On the one hand, it tightens the formulation; on the other hand, it also increases the problem size. It is also interesting to point out that for all

problem instances solved to optimality, the objective values with and without concavity constraints are extremely close. For problem instances not solved to optimality, the upper and lower bounds of objective values with and without concavity constraints are also very close. This suggests that enforcing concavity constraint does not lead to much optimality loss.

Our computational results are very encouraging. We are therefore able to solve decent sized problems — problems comparable to ones reported in the literature — within seconds, instead of hours reported in the recent literature. We would like to note that given the development in this area, column generation approach will not be the method of choice. We believe that enforcing additional constraints (such as concavity constraints) is a promising direction to tackling computational challenges, potentially in more general problem setups.

The concavity property exploited in this paper should be contrasted with research that studies structural properties of *optimal* bid-prices (Berman et al. 2012, Xu and Hopp 2009, Akan and Ata 2009). These papers aim at understanding the behavior of bid-prices if they are computed optimally. In particular, they discuss when the optimal bid-prices form martingales, submartingales, and supermartingales. The dynamic bid-prices we consider are monotonically decreasing over time, and therefore (trivially) follow a supermartingale. This behavior does not contradict results in the literature since the dynamic bid-prices are only time-dependent but not capacity-dependent, and therefore are not optimal. Our computational focus should also be contrasted with the theoretical focus of the aforementioned papers. We also would like to point out that much of the existing research considers *first-order* properties of bid-prices, including the aforementioned monotonicity and martingale properties. Our paper is the first one that studies a *second-order* property (concavity) of bid-prices in the literature.

The remainder of the paper is organized as follows. Section 2 starts with some preliminaries. Section 3 develops a dynamic disaggregation algorithm. Section 4 extends our main theoretical results to network RM with customer choice. Section 5 discusses the concavity properties of dynamic bid-prices. Section 6 reports computational results and Section 7 summarizes. Additional proofs and technical derivations are provided in the appendix.

2. Preliminaries

In this section, we formulate the network RM problem, and review the affine functional approximation approach proposed in Adelman (2007). We then briefly discuss the reduction introduced in Tong and Topaloglu (2011) and Vossen and Zhang (2013).

2.1. Dynamic Programming Formulation

For ease of exposition, we use airline terminology throughout the paper. Consider a flight network with m legs and the set of capacities $c = (c_1, \ldots, c_m)$, where c_i is the capacity of leg i. There are n products offered, where a product is a flight itinerary and fare class combination. Let $N = \{1, \ldots, n\}$ be the set of products. The fare for product j is f_j. The consumption matrix is an $(m \times n)$-matrix $A \equiv (a_{ij})$. The entry $a_{ij} \in \{0, 1\}$ represents the amount of resource i required by a class j customer purchasing product j. The i-th row A_i is the incidence vector for leg i, and the j-th column A^j is the incidence vector for product j. There are τ discrete time periods that are counted forward, so period τ is the last period. To simplify notation, we reserve the symbols i, j, and t for legs, products, and time, respectively.

We assume there is at most one customer arrival in each period. The probability of a class-j customer arrival in period t is $\lambda_{t,j}$. Therefore, the probability of no customer arrival in period t is $1 - \sum_j \lambda_{t,j}$. The decision to be made over time is whether to accept or reject each arriving customer in order to maximize total expected revenue. A rejected customer leaves immediately, while an accepted class-j customer consumes resources as specified in A^j.

At time t, let $x = (x_1, \ldots, x_m)$ be the vector of remaining capacity, where x_i is the number of remaining seats on resource i. Let $v_t(x)$ be the value function denoting the maximum expected total revenue from time t onwards, given remaining capacity x. The dynamic programming equations can be written as

$$v_t(x) = \max_{u \in \mathcal{U}(x)} \left\{ \sum_j \lambda_{t,j} [f_j u_j + v_{t+1}(x - A^j u_j)] + \left(1 - \sum_j \lambda_{t,j}\right) v_{t+1}(x) \right\}, \qquad \forall t, x, \qquad (1)$$

where the action space is $\mathcal{U}(x) = \{u \in \{0,1\}^n : A^j u_j \leq x, \forall j\}$. The variable u_j indicates whether product j is accepted. The boundary conditions are $v_{\tau+1}(x) = 0$ for all x.

It is not hard to show that an optimal policy for (1) is given by

$$u_{t,j}(x) = \begin{cases} 1, \text{ if } A^j \leq x, f_j \geq v_{t+1}(x) - v_{t+1}(x - A^j), \\ 0, \text{ otherwise,} \end{cases} \qquad \forall t, x, j.$$

The optimal policy states that if there is enough resources left and the revenue of accepting a customer exceeds the opportunity cost, then the customer should be accepted.

2.2. Affine Approximation and Reduction

Adelman (2007) considers an equivalent linear programming formulation for (1):

$$(\mathbf{LP}) \quad \min_{\{v_t(\cdot)\}_{\forall t}} v_1(c)$$

$$v_t(x) \geq \sum_j \lambda_{t,j}[f_j u_j + v_{t+1}(x - A^j u_j)] + \left(1 - \sum_j \lambda_{t,j}\right) v_{t+1}(x), \qquad \forall t, x \in \mathcal{X}_t, u \in \mathcal{U}(x).$$

In the above, $\mathcal{X}_t$ is the state space in period t and is given by

$$\mathcal{X}_t = \begin{cases} \{c\}, & \text{if } t = 1, \\ \{x \in \mathcal{Z}_+^m : x \leq c\}, & \text{if } t = 2, \ldots, \tau, \end{cases}$$

where $\mathcal{Z}_+$ denotes nonnegative integers.

The number of variables and constraints in (**LP**) increase exponentially in the number of resources m. Therefore, brute-force solution of (**LP**) is at least as difficult as solving the dynamic programming formulation (1) directly. One approach to reducing (**LP**) is approximating the value function $v_t(x)$ by weighted basis functions. Adelman (2007) considers an affine functional approximation

$$v_t(x) \approx \theta_t + \sum_i V_{t,i} x_i, \qquad \forall t, x, \tag{2}$$

where it is assumed that $\theta_{\tau+1} = 0$ and $V_{\tau+1,i} = 0$ for all i. Plugging (2) into (**LP**) leads to the following linear program:

$$\begin{aligned} \textbf{(LP1)} \quad & \min_{\theta, V} \theta_1 + \sum_i V_{1,i} c_i \\ & \theta_t - \theta_{t+1} + \sum_i \left[V_{t,i} x_i - V_{t+1,i} \left(x_i - \sum_j \lambda_{t,j} a_{ij} u_j \right) \right] \geq \sum_j \lambda_{t,j} f_j u_j, \qquad \forall t, x \in \mathcal{X}_t, u \in \mathcal{U}(x). \end{aligned}$$

In order to write the dual of (**LP1**), we introduce the dual variable $p_{t,x,u}$ corresponding to the constraints in (**LP1**). The dual of (**LP1**) is given by

$$\begin{aligned} \textbf{(D1)} \quad & \max_p \sum_{t, x \in \mathcal{X}_t, u \in \mathcal{U}(x)} \left(\sum_j \lambda_{t,j} u_j f_j \right) p_{t,x,u} \\ & \sum_{x \in \mathcal{X}_t, u \in \mathcal{U}(x)} x_i p_{t,x,u} = \begin{cases} c_i, & \text{if } t = 1, \\ \sum_{x \in \mathcal{X}_{t-1}, u \in \mathcal{U}(x)} (x_i - \sum_j \lambda_{t-1,j} a_{ij} u_j) p_{t-1,x,u}, & \text{if } t > 1, \end{cases} \qquad \forall i, t, \\ & \sum_{x \in \mathcal{X}_t, u \in \mathcal{U}(x)} p_{t,x,u} = \begin{cases} 1, & \text{if } t = 1, \\ \sum_{x \in \mathcal{X}_{t-1}, u \in \mathcal{U}(x)} p_{t-1,x,u}, & \text{if } t = 2, \ldots, \tau, \end{cases} \qquad \forall t, \qquad (3) \\ & p \geq 0. \end{aligned}$$

Note that constraint (3) can be reduced to:

$$\sum_{x \in \mathcal{X}_t, u \in \mathcal{U}(x)} p_{t,x,u} = 1, \qquad \forall t.$$

The number of columns in (**D1**) grows exponentially in the number of resources m and the number of products n, and therefore can be huge. However, the number of constraints are only

linearly increasing in the number of periods τ and the number of resources m. Column generation has been proposed to solve (**D1**) in Adelman (2007).

The formulation (**D1**) can be reduced to a much smaller linear program. This reduction was recently considered by Tong and Topaloglu (2011). Vossen and Zhang (2013) provide an alternative proof based on a Dantzig-Wolfe reformulation of the reduced program.

PROPOSITION 1 **(Tong and Topaloglu (2011), Vossen and Zhang (2013))**. *The program (**D1**) can be reduced to the following equivalent linear program:*

(D2)

$$\max_{r,q} \sum_{t,j} \lambda_{t,j} f_j q_{t,j}$$

$$r_{t,i} = \begin{cases} c_i, & \text{if } t = 1, \\ r_{t-1,i} - \sum_j \lambda_{t-1,j} a_{ij} q_{t-1,j}, & \text{if } t > 1, \end{cases} \qquad \forall i, t, \tag{4}$$

$$a_{ij} q_{t,j} \le r_{t,i}, \qquad \forall t, i, j, \tag{5}$$

$$q_{t,j} \le 1, \qquad \forall t, j, \tag{6}$$

$$q \ge 0, r \ge 0. \tag{7}$$

Proposition 1 is the result of some special structure of the ALP resulting from affine approximation; for more details, see Tong and Topaloglu (2011), Vossen and Zhang (2013). Since the solution times of linear programs are often growing in the number of constraints and variables, the more compact formulation (**D2**) can be solved in much shorter time, even if we only used standard solution methods. We would like to take one further step. The next section introduces a specialized algorithm that exploits the special structure of (**D2**). We show that the algorithm is much faster than standard solution method and can beat a recent solution method proposed in Tong and Topaloglu (2011).

3. A Dynamic Disaggregation Approach

The size of the reduced program (**D2**) grows linearly in the number of periods, resources, and products, and is therefore much smaller than (**D1**). Tong and Topaloglu (2011) propose a constraint generation algorithm to solve (**D2**) and show that it significantly speeds up the solution compared with the column generation algorithm applied to (**D1**). Their solution procedure, however, still needs to deal with formulations with a large number of variables, for problems that involve a large number of periods, resources, and products. In this section, we propose an approach that combines constraint generation and column generation, where the overall idea is to iteratively disaggregate the formulation.

After substituting out the $r_{t,i}$ variables and using the change of variables $z_{t,j} = \lambda_{t,j} q_{t,j}$ in (**D2**), we obtain the equivalent formulation

$$
\begin{aligned}
(\mathbf{D3}) \qquad & \max_{z} \sum_{t,j} f_j z_{t,j} \\
& z_{t,j} \le \lambda_{t,j} \left[c_i - \sum_{k=1}^{t-1} \sum_{j'} a_{ij'} z_{k,j'} \right], && \forall t, i, j : a_{ij} = 1, && (8) \\
& z_{t,j} \le \lambda_{t,j}, && \forall t, j, && (9) \\
& z \ge 0.
\end{aligned}
$$

The formulation (**D3**) can be interpreted as follows. The variable $z_{t,j}$ can be interpreted as the accepted product j demand in period t. Constraint (9) states that accepted product j demand cannot exceed $\lambda_{t,j}$, which is total product j demand. Constraint (8) enforces a time-sensitive resource constraint, since $c_i - \sum_{k=1}^{t-1} \sum_{j'} a_{ij'} z_{k,j'}$ represents available resource i at time t.

The main idea of our dynamic disaggregation approach is to iteratively solve a time-aggregated version of (**D3**). Let α be an integer between 1 and τ. Let

$$
\begin{aligned}
Z_{\alpha,j} &= \sum_{t=1}^{\alpha} z_{t,j}, && \forall j, \\
\Lambda_{\alpha,j} &= \sum_{t=1}^{\alpha} \lambda_{t,j}, && \forall j.
\end{aligned}
$$

Consider the following α-aggregated version of (**D3**):

$$
\begin{aligned}
(\mathbf{D3}\text{-}\alpha) \qquad & \max_{Z,z} \sum_{j} f_j Z_{\alpha,j} + \sum_{t=\alpha+1}^{\tau} f_j z_{t,j} \\
& \sum_{j} a_{ij} \left(Z_{\alpha,j} + \sum_{t=\alpha+1}^{\tau} z_{t,j} \right) \le c_i, && \forall i, \\
& z_{t,j} \le \lambda_{t,j} \left[c_i - \sum_{j'} a_{ij'} \left(Z_{\alpha,j'} + \sum_{k=\alpha+1}^{t-1} z_{k,j'} \right) \right], && \forall t = \alpha+1, \ldots, \tau, i, j : a_{ij} = 1, && (10) \\
& Z_{\alpha,j} \le \Lambda_{\alpha,j}, && \forall j, \\
& z_{t,j} \le \lambda_{t,j}, && \forall t = \alpha+1, \ldots, \tau, j, \\
& Z, z \ge 0.
\end{aligned}
$$

In the formulation above, we take $z_{\tau+1,j} = 0$ for all j. Observe that (**D3-**α) is derived from (**D3**) by aggregating variables and constraints for $z_{1,j}, \ldots, z_{\alpha,j}$ for each j. Therefore, (**D3-**α) is a relaxation of (**D3**). When $\alpha = \tau$, (**D3-**α) reduces to

$$
(\mathbf{D3}\text{-}\tau) \qquad \max_{Z} \sum_{j} f_j Z_{\tau,j}
$$

$$\begin{aligned}
&\sum_j a_{ij} Z_{\tau,j} \le c_i, && \forall i,\\
&Z_{\tau,j} \le \Lambda_{\tau,j}, && \forall j,\\
&Z \ge 0.
\end{aligned}$$

The formulation (**D3-**τ) is the well-known DLP formulation for network RM (Talluri and van Ryzin 1998, Cooper 2002).

Next, we state a proposition which is crucial for the validity of our proposed solution procedure.

PROPOSITION 2. *If* $\alpha = 1$, *then (**D3-**α) coincides with (**D3**). If* $\alpha > 1$, *suppose* (Z^*, z^*) *is an optimal solution to (**D3-**α). If*

$$\frac{Z^*_{\alpha,j}}{\Lambda_{\alpha,j}} + \sum_{j'} a_{ij'} \frac{\Lambda_{\alpha-1,j'} Z^*_{\alpha,j'}}{\Lambda_{\alpha,j'}} - c_i \le 0, \quad \forall i, j : a_{ij} = 1, \tag{11}$$

then

$$\tilde{z}_{t,j} = \begin{cases} \lambda_{t,j} \frac{Z^*_{\alpha,j}}{\Lambda_{\alpha,j}}, & \text{if } t = 1, \ldots, \alpha, \Lambda_{\alpha,j} > 0, \\ 0, & \text{if } t = 1, \ldots, \alpha, \Lambda_{\alpha,j} = 0, \\ z^*_{t,j}, & \text{if } t = \alpha+1, \ldots, \tau, \end{cases} \qquad \forall t, j,$$

*is an optimal solution to (**D3**).*

Proof. Since (**D3-**α) is a relaxation of (**D3**) and $\tilde{z}$ gives an objective value for (**D3**) that is the same as the optimal objective in (**D3-**α), $\tilde{z}$ is an optimal solution to (**D3**) if it is feasible. It remains to show that $\tilde{z}$ is a feasible solution to (**D3**).

For each j such that $\Lambda_{\alpha,j} = 0$, an optimal solution to (**D3**) satisfies $z_{t,j} = 0$ for all $t = 1, \ldots, \alpha$. In what follows, we assume $\Lambda_{\alpha,j} > 0$. By construction, $\tilde{z}$ satisfies all constraints in (**D3**) except (8). By the definitions of $\tilde{z}$, (8) holds for $t = \alpha+1, \ldots, \tau$. It remains to verify that

$$\tilde{z}_{t,j} \le \lambda_{t,j} \left[c_i - \sum_{k=1}^{t-1} \sum_{j'} a_{ij'} \tilde{z}_{k,j'} \right], \qquad \forall t = 1, \ldots, \alpha, i, j : a_{ij} = 1.$$

The above inequality is equivalent to

$$\frac{Z^*_{\alpha,j}}{\Lambda_{\alpha,j}} \le c_i - \sum_{k=1}^{t-1} \sum_{j'} a_{ij'} \frac{Z^*_{\alpha,j'}}{\Lambda_{\alpha,j'}}, \quad \forall t = 1, \ldots, \alpha, i, j : a_{ij} = 1.$$

Note that the right hand side above is decreasing in t. Hence, it suffices to verify the inequality for $t = \alpha$, i.e.,

$$\frac{Z^*_{\alpha,j}}{\Lambda_{\alpha,j}} \le c_i - \sum_{j'} a_{ij'} \frac{\Lambda_{\alpha-1,j'} Z^*_{\alpha,j'}}{\Lambda_{\alpha,j'}}, \quad \forall i, j : a_{ij} = 1,$$

which holds by (11). This completes the proof. ■

The dynamic disaggregation approach can be described as follows.

1. Initialize: Let $\alpha = \tau$.

2. Solve Aggregate LP: If $\alpha = 1$, stop. Otherwise, solve (**D3-**α), and denote the optimal solution (Z^*, z^*).

3. Test Violation: Let

$$\nu = \max_{i,j:a_{ij}=1} \frac{Z^*_{\alpha,j}}{\Lambda_{\alpha,j}} + \sum_{j'} a_{ij'} \frac{\Lambda_{\alpha-1,j'} Z^*_{\alpha,j'}}{\Lambda_{\alpha,j'}} - c_i.$$

4. Update: If $\nu \le 0$, terminate. Otherwise, let $\alpha = \alpha - 1$, and go to Step 2.

We did not introduce a stopping criterion that allows for arbitrary optimality tolerance. We ignore the stopping criterion here, because the algorithm typically solves decent sized problems within seconds. However, it is not hard to provide a stopping criterion. In Section 4, we introduce such a stopping criterion for essentially the same algorithm when there is customer choice behavior involved.

An optimal solution (Z, z) to (**D3-**α) can be used to constructed an allocation policy to different resources. However, a "bid-price" type policy cannot be constructed from the solution directly. This is a potential drawback since there is considerable interest in constructing bid-price policies from V (Adelman 2007). Here, we show that an optimal solution (θ, V) to (**LP1**) can be recovered from the dual information of (**D3-**α). A proof is given in Appendix A.

PROPOSITION 3. *Associate dual variables* $(\beta, \gamma, \phi, \eta)$ *to the constraints in (**D3-**α). Let* $(\beta^*, \gamma^*, \phi^*, \eta^*)$ *be the dual values at optimality. Let*

$$V^*_{t,i} = \begin{cases} \beta^*_i + \sum_{k=\alpha+1}^{\tau} \sum_j a_{ij} \lambda_{k,j} \gamma^*_{kij}, & \text{if } t = 1, \dots, \alpha, \\ \beta^*_i + \sum_{k=t}^{\tau} \sum_j a_{ij} \lambda_{k,j} \gamma^*_{kij}, & \text{if } t = \alpha+1, \dots, \tau, \end{cases} \qquad \forall t, i, \tag{12}$$

$$\theta^*_t = \begin{cases} \sum_j \left(\sum_{k=t}^{\alpha} \lambda_{k,j} \phi^*_j + \sum_{k=\alpha+1}^{\tau} \lambda_{k,j} \eta^*_{k,j} \right), & \text{if } t = 1, \dots, \alpha, \\ \sum_j \sum_{k=t}^{\tau} \lambda_{k,j} \eta^*_{k,j}, & \text{if } t = \alpha+1, \dots, \tau, \end{cases} \qquad \forall t. \tag{13}$$

Then (V^*, θ^*) *is an optimal solution to (**LP1**).*

4. Dynamic Disaggregation for Choice-Based Network Revenue Management

In this section, we extend our earlier results to choice-based network RM. An early formulation of the problem is given by Gallego et al. (2004). Our formulation closely follows Liu and van Ryzin (2008). The basic setup is the same as the network RM problem introduced before. However, it is assumed that there is a customer arrival with probability λ in each period and each arriving customer chooses product $j \in S$ with probability $P_j(S)$ when the set of open products is $S \subseteq N$, where $N = \{1, \dots, n\}$ denotes the set of all products.[1] The decision in each period is the offer set

[1] Our formulation here assumes stationary arrival rates and choice probabilities for notational simplicity. The model can be extended to non-homogeneous arrival rates and choice probabilities by dividing the booking horizon into segments with stationary parameters; see also discussion in Liu and van Ryzin (2008). All of our theoretical results hold for the generalized model subject to minor modifications.

S.

With slight abuse of notation, we still use $v_t(x)$ to denote the value function. The dynamic programming equations can be written as

$$v_t(x) = \max_{S \subseteq N(x)} \left\{ \lambda \sum_j P_j(S)[f_j + v_{t+1}(x - A^j)] + \left(1 - \lambda \sum_j P_j(S)\right) v_{t+1}(x) \right\}, \qquad \forall t, x, \qquad (14)$$

where $N(x) = \{j \in N : x \geq A^j\}$ is the set of products that can be offered given resource vector x. The boundary conditions are $v_{\tau+1}(x) = 0$ for all x.

Zhang and Adelman (2009) generalize the affine functional approximation approach of Adelman (2007) to network RM with customer choice. Using the affine approximation (2) in the linear programming formulation for (14), we obtain

$$\begin{aligned} (\textbf{CLP1}) \qquad & \min_{\theta, V} \theta_1 + \sum_i V_{1,i} c_i \\ & \theta_t - \theta_{t+1} + \sum_i \left(V_{t,i} x_i - V_{t+1,i}(x_i - \lambda Q_i(S))\right) \geq \lambda R(S), \quad \forall t, x, S \subseteq N(x). \end{aligned}$$

In the above $R(S) = \sum_{j \in S} f_j P_j(S)$ is the revenue rate of offer set $S \subseteq N$, and $Q_i(S) = \sum_{j \in S} a_{ij} P_j(S)$ is the resource consumption rate of offer set $S \subseteq N$ for each resource i. The corresponding dual program is given by

$$\begin{aligned} \textbf{(CD1)} \qquad & \max_y \sum_{t, x \in \mathcal{X}_t, S \subseteq N(x)} \lambda R(S) y_{t,x,S} \\ & \sum_{x \in \mathcal{X}_t, S \subseteq N(x)} x_i y_{t,x,S} = \begin{cases} c_i, & \text{if } t = 1, \\ \sum_{x \in \mathcal{X}_{t-1}, S \subseteq N(x)} (x_i - \lambda Q_i(S)) y_{t-1,x,S}, & \text{if } t > 1, \end{cases} \qquad \forall t, i \\ & \sum_{x \in \mathcal{X}_t, S \subseteq N(x)} y_{t,x,S} = 1, \qquad \forall t, \\ & y \geq 0. \end{aligned}$$

The program (**CD1**) has a large number of variables but relatively few constraints, so it can be solved via column generation (Zhang and Adelman 2009). Recently, Vossen and Zhang (2013) show that (**CD1**) can be reduced to a more compact linear program.

PROPOSITION 4 **(Vossen and Zhang (2013))**. *The program (**CD1**) can be reduced to the following equivalent linear program:*

$$\begin{aligned} \textbf{(CD2)} \qquad & \max_{r,h} \sum_{t, S \subseteq N} \lambda R(S) h_{t,S} \\ & r_{t,i} = \begin{cases} c_i, & \text{if } t = 1, \\ r_{t-1,i} - \sum_{S \subseteq N} \lambda Q_i(S) h_{t-1,S}, & \text{if } t > 1, \end{cases} \qquad \forall t, i, \\ & \sum_{S \subseteq N : i \in I(S)} h_{t,S} \leq r_{t,i}, \qquad \forall t, i, \end{aligned}$$

$$\sum_{S\subseteq N} h_{t,S} = 1, \qquad \forall t, \tag{15}$$
$$r, h \geq 0.$$

In the above, $I(S) = \{i : \sum_{j\in S} a_{ij} \geq 1\}$ *is the set of resources used by the products in* S*.*

In the reminder of this section, we develop a dynamic disaggregation approach to (**CD2**). First, we note that the dual of (**CD2**) can be solved by the constraint generation procedure described below. Associating variables (θ, W, V) to the constraints in (**CD2**), the dual program of (**CD2**) can be written as

(**CLP2**)
$$\begin{aligned}
&\min_{\theta,W,V} \theta_1 + \sum_i V_{1,i} c_i \\
&-W_{t,i} + V_{t,i} - V_{t+1,i} \geq 0, \qquad \forall t, i, \\
&\theta_t - \theta_{t+1} + \sum_{i\in I(S)} W_{t,i} + \sum_{i\in I(S)} \lambda Q_i(S) V_{t+1,i} \geq \lambda R(S), \qquad \forall t, S, \\
&W_{t,i} \geq 0, \qquad \forall t, i.
\end{aligned}$$

Note that when writing the dual above, we have replaced the constraint (15) in (**CD2**) with the equivalent constraint

$$\sum_{S\subseteq N} h_{t,S} = \begin{cases} 1, & \text{if } t = 1, \\ \sum_{S\subseteq N} h_{t-1,S}, & \text{if } t > 1, \end{cases} \qquad \forall t.$$

We also used the fact that $Q_i(S) = 0$ for $i \notin I(S)$.

For a feasible solution (θ, W, V) to (**CLP2**), a new feasible solution (θ, W', V) with the same objective value can be constructed by taking $W'_{t,i} = V_{t,i} - V_{t+1,i}$ for all t and i. Therefore, the inequality in the first constraint in (**CLP2**) can be replaced by an equality. Consequently, we can substitute out W and rewrite (**CLP2**) as

(**CLP3**)
$$\begin{aligned}
&\min_{\theta,V} \theta_1 + \sum_i V_{1,i} c_i \\
&V_{t,i} - V_{t+1,i} \geq 0, \qquad \forall t, i, \\
&\theta_t - \theta_{t+1} + \sum_{i\in I(S)} (V_{t,i} - V_{t+1,i}(1 - \lambda Q_i(S))) \geq \lambda R(S), \qquad \forall t, S.
\end{aligned}$$

Our result so far implies that (**CLP3**) is equivalent to (**CLP1**), but (**CLP3**) has much fewer constraints.

As an aside, we note that (**CLP3**) immediately implies that there exists an optimal solution where $V_{t,i}$ is monotonically decreasing in t for each i. This result was first pointed out by Adelman (2007) for affine approximations to network RM without choice. This monotonicity result was extended to the choice setting by Zhang and Adelman (2009). Both Adelman (2007) and Zhang and Adelman (2009) use constructive proof for the monotonicity, whereas here the monotonicity

is a direct consequence of the reformulation. Clearly, the monotonicity of V in the choice setting immediately implies the monotonicity of V in the no-choice setting, since network RM without choice can be cast as a special case of the model with choice.

The program (**CLP3**) can be solved by a constraint generation algorithm, where we solve the following constraint generation subproblem in each period t:

$$\max_{S\subseteq N} \lambda R(S) - \sum_{i\in I(S)} (V_{t,i} - V_{t+1,i}(1-\lambda Q_i(S))) - \theta_t + \theta_{t+1}.$$

The problem above can be solved as a mixed integer linear program for multinomial logit choice model; see Zhang and Adelman (2009).

Although (**CD2**) can be solved by the procedure described above, we show that it can be solved by a dynamic disaggregation approach, extending the results in Section 3 to the choice setting as well. After substituting out r, (**CD2**) can be written as

$$\begin{aligned}
(\textbf{CD3}) \qquad & \max_{h} \sum_{t,S\subseteq N} \lambda R(S) h_{t,S} \\
& \sum_{k=1}^{t-1} \sum_{S\subseteq N} \lambda Q_i(S) h_{k,S} + \sum_{S\subseteq N: i\in I(S)} h_{t,S} \le c_i, && \forall t, i, \qquad (16)\\
& \sum_{S\subseteq N} h_{t,S} = 1, && \forall t, \\
& h \ge 0.
\end{aligned}$$

The formulation (**CD3**) resembles an alternative linear programming formulation for choice-based RM given in Kunnumkal and Topaloglu (2008). The difference is that they replaced the constraint (16) with

$$\sum_{k=1}^{t-1} \sum_{S\subseteq N} \lambda Q_i(S) h_{k,S} + \sum_{S\subseteq N} \mathbb{I}\{j\in S\}\, a_{ij} h_{t,S} \le c_i, \qquad \forall t, i, j, \qquad (17)$$

where $\mathbb{I}\{\}$ is the indicator function. Since

$$\sum_{S\subseteq N} \mathbb{I}\{j\in S\}\, a_{ij} h_{t,S} \le \sum_{S\subseteq N: i\in I(S)} h_{t,S}, \qquad \forall j,$$

the constraint (17) is weaker than (16). Hence, the alternative formulation in Kunnumkal and Topaloglu (2008) is also weaker than (**CD3**). Indeed, Kunnumkal and Topaloglu show that the objective value from their formulation is not necessarily a tighter bound than the bound from CDLP. On the other hand, since (**CD3**) is equivalent to (**CD1**), (**CD3**) produces a tighter bound than CDLP; see Zhang and Adelman (2009).

Let α be an integer between 1 and τ, and define

$$H_{\alpha,S} = \sum_{t=1}^{\alpha} h_{t,S}, \qquad \forall S.$$

An α-aggregated version of (**CD3**) is given by

$$\begin{aligned}
(\textbf{CD3-}\alpha) \quad & \max_{H,h} \sum_{S\subseteq N} \lambda R(S) H_{\alpha,S} + \sum_{t=\alpha+1}^{\tau} \sum_{S\subseteq N} \lambda R(S) h_{t,S} \\
& \sum_{S\subseteq N} \lambda Q_i(S) H_{\alpha,S} + \sum_{k=\alpha+1}^{t-1} \sum_{S\subseteq N} \lambda Q_i(S) h_{k,S} + \sum_{S\subseteq N: i\in I(S)} h_{t,S} \le c_i, \quad \forall t = \alpha+1,\dots,\tau+1, i, \\
& \sum_{S\subseteq N} H_{\alpha,S} = \alpha, \\
& \sum_{S\subseteq N} h_{t,S} = 1, \qquad \forall t = \alpha+1,\dots,\tau, \\
& H, h \ge 0.
\end{aligned} \tag{18}$$

In the above, we take $h_{\tau+1,S} = 0$ for all S. When $\alpha = \tau$, (**CD3-**α) becomes

$$\begin{aligned}
(\textbf{CD3-}\tau) \qquad & \max_{H} \sum_{S\subseteq N} \lambda R(S) H_{\tau,S} \\
& \sum_{S\subseteq N} \lambda Q_i(S) H_{\tau,S} \le c_i, \qquad \forall i, \\
& \sum_{S\subseteq N} H_{\tau,S} = \tau, \\
& H \ge 0.
\end{aligned}$$

Notice that (**CD3-**τ) is the choice-based deterministic linear program (CDLP) for network RM (Gallego et al. 2004, Liu and van Ryzin 2008).

The formulation (**CD3-**α) can be solved via column generation. Associating dual variables $W_{t,i}, \theta_\alpha$, and θ_t to the constraints in (**CD3-**α), we can solve one column generation subproblem for each $t = \alpha,\dots,\tau$. For $t = \alpha$, the column generation subproblem is given by

$$\max_{S\subseteq N} \lambda R(S) - \sum_i \lambda Q_i(S) \left(\sum_{t=\alpha+1}^{\tau} W_{t,i} \right) - \theta_\alpha. \tag{19}$$

For each $t = \alpha+1,\dots,\tau$, the column generation subproblem is given by

$$\max_{S\subseteq N} \lambda R(S) - \sum_{i\in I(S)} W_{t,i} - \sum_i \lambda Q_i(S) \left(\sum_{k=t+1}^{\tau} W_{k,i} \right) - \theta_t. \tag{20}$$

We have the following proposition.

PROPOSITION 5. *If $\alpha = 1$, then (**CD3-**α) coincides with (**CD3**). If $\alpha > 1$, suppose (H^*, h^*) is an optimal solution to (**CD3-**α). If*

$$\sum_{S\subseteq N} \lambda Q_i(S) \frac{(\alpha-1) H^*_{\alpha,S}}{\alpha} + \sum_{S\subseteq N: i\in I(S)} \frac{H^*_{\alpha,S}}{\alpha} \le c_i, \qquad \forall i, \tag{21}$$

Then

$$\tilde{h}_{t,S} = \begin{cases} \frac{H^*_{\alpha,S}}{\alpha}, \text{ if } t = 1, \ldots, \alpha, \\ h^*_{t,S}, \text{ if } t = \alpha+1, \ldots, \tau, \end{cases} \qquad \forall t, S,$$

*is an optimal solution to (**CD3**).*

Proof. Since (**CD3-**α) is a relaxation of (**CD3**) and $\tilde{h}$ gives an objective value for (**CD3**) that is the same as the optimal objective in (**CD3-**α), $\tilde{h}$ is an optimal solution to (**CD3**) if it is feasible. It remains to show that if (21) holds, then $\tilde{h}$ is a feasible solution to (**CD3**).

By construction, $\tilde{h}$ satisfies all constraints in (**CD3**) except (16) for $t = 1, \ldots, \alpha$. Therefore, we need to verify that

$$\sum_{k=1}^{t-1} \sum_{S \subseteq N} \lambda Q_i(S) \tilde{h}_{k,S} + \sum_{S \subseteq N : i \in I(S)} \tilde{h}_{t,S} \leq c_i, \qquad \forall t = 1, \ldots, \alpha, i.$$

Using the definition of $\tilde{h}$, the above inequality can be written as

$$\sum_{k=1}^{t-1} \sum_{S \subseteq N} \lambda Q_i(S) \frac{H^*_{\alpha,S}}{\alpha} + \sum_{S \subseteq N : i \in I(S)} \frac{H^*_{\alpha,S}}{\alpha} \leq c_i, \qquad \forall t = 1, \ldots, \alpha, i.$$

Note that the left hand side above is increasing in t. Hence, it suffices to verify the inequality for $t = \alpha$, which holds by (21). This completes the proof. ■

The dynamic disaggregation approach can be described as follows.

1. Initialize: Let $\alpha = \tau$.

2. Solve Aggregate LP: If $\alpha = 1$, stop. Otherwise, solve (**CD3-**α), and denote the optimal solution (H^*, h^*).

3. Test Violation: Let

$$\nu = \max_i \sum_{S \subseteq N} \lambda Q_i(S) \frac{(\alpha-1) H^*_{\alpha,S}}{\alpha} + \sum_{S \subseteq N : i \in I(S)} \frac{H^*_{\alpha,S}}{\alpha} - c_i.$$

4. Update: If $\nu \leq 0$, terminate. Otherwise, let $\alpha = \alpha - 1$, and go to Step 2.

The solution procedure above does not avail itself to a stopping criterion. In our numerical experiments, we construct a lower bound to (**CD3**) while we disaggregate (**CD3-**α) to (**CD3-**$(\alpha-1)$). [2] This result is summarized in the following proposition.

[2] We also consider several alternative lower bounds. One lower bound is given by solving restricted version of (**CD3**) that involves only product set S that appeared in solutions to (**CD3-**α). This lower bound, however, requires solving much larger fully disaggregated programs, while the lower bound given in Proposition 6 only involves solving a partially disaggregated program.

PROPOSITION 6. *For each $\alpha = 2, \ldots, \tau$, the objective value of (**CD3**-$(\alpha-1)$) with the additional constraint*

$$H_{\alpha-1,S} = (\alpha-1)h_{\alpha,S}, \qquad \forall S, \tag{22}$$

*gives a lower bound for (**CD3**).*

Proof. Let (H, h) be an optimal solution to (**CD3**-$(\alpha-1)$) with the additional constraint (22). Let $\hat{h}_{t,S} = h_{\alpha,S}$ for $t = 1, \ldots, \alpha$ and $\hat{h}_{t,S} = h_{t,S}$ for $t = \alpha+1, \ldots, \tau$. The additional constraint (22) ensures that $\hat{h}$ satisfies the resource constraint (16) in (**CD3**). Hence, it is easy to verify that $\hat{h}$ is feasible for (**CD3**). Note that the objective value for (**CD3**) at the solution $\hat{h}$ is the same as the objective value of (**CD3**-$(\alpha-1)$) with the additional constraint (22). This completes the proof. ■

5. Concavity Properties of Dynamic Bid-Prices

In this section, we discuss strong structural properties of dynamic bid-prices for the independent demand model. We make an additional assumption that the arrivals over time are homogeneous; i.e., the arrival rate $\lambda_{t,j}$ for class-j customer does not dependent on time t. Consequently, we use λ_j instead of $\lambda_{t,j}$. The homogeneity assumption is somewhat strong. However, it is common in practice to divide the booking horizon into time segments with homogeneous arrival rates, because it is difficult to forecast demand separately for each time interval. Our theoretical results can be easily shown to hold within each time segment with homogeneous arrival rates.

The primal formulation of (**D2**) can be written as:

(P2)
$$\begin{aligned}
&\min_{V,U,\theta} \sum_{t,j} \theta_{t,j} + \sum_i c_i V_{1,i} \\
&V_{t,i} - V_{t+1,i} - \sum_j a_{ij} U_{tij} \geq 0 && \forall i, t, && (23)\\
&\theta_{t,j} + \sum_i a_{ij} U_{tij} + \lambda_j \sum_i a_{ij} V_{t+1,i} \geq \lambda_j f_j && \forall t, j, && (24)\\
&V, U, \theta \geq 0.
\end{aligned}$$

Using the substitution, $W_{t,i} = V_{t,i} - V_{t+1,i}$, we can reformulate the primal as:

(AP2)
$$\begin{aligned}
&\min_{W,U,\theta} \sum_{t,j} \theta_{t,j} + \sum_i c_i \left(\sum_t W_{t,i} \right) \\
&W_{t,i} - \sum_j a_{ij} U_{tij} = 0 && \forall i, t, && (25)\\
&\theta_{t,j} + \sum_i a_{ij} U_{tij} + \lambda_j \sum_i a_{ij} \left(\sum_{k=t+1}^{\tau} W_{k,i} \right) \geq \lambda_j f_j && \forall t, j, \\
&W, U, \theta \geq 0.
\end{aligned}$$

In the above, note that the inequality in (23) is replaced with equality. This can be done without loss of generality. To see this, suppose $W_{t,i} > \sum_j a_{ij} U_{tij}$ for some t, i in an optimal solution. We

can increase U_{tij} to reach equality in (23) without affecting the objective value and feasibility of constraint (24).

Substituting out W in (**AP2**), we obtain

$$\textbf{(AP3)} \qquad z_{\mathrm{AP3}} = \min_{U,\theta} \sum_{t,j} \theta_{t,j} + \sum_i c_i \left(\sum_{t,j} a_{ij} U_{tij} \right)$$

$$\theta_{t,j} + \sum_i a_{ij} U_{tij} + \lambda_j \sum_i a_{ij} \left(\sum_{k=t+1}^{\tau} \sum_{j'} a_{ij'} U_{kij'} \right) \geq \lambda_j f_j \qquad \forall t, j, \tag{26}$$

$$U, \theta \geq 0.$$

The dual of (**AP3**) can be formulated as

$$\textbf{(AD3)} \qquad \max_{q} \sum_{t,j} \lambda_j f_j q_{t,j}$$

$$q_{t,j} + \sum_{j'} \sum_{k=1}^{t-1} \lambda_{j'} a_{ij'} q_{k,j'} \leq c_i, \qquad \forall t, i, j : a_{ij} = 1, \tag{27}$$

$$q_{t,j} \leq 1, \qquad \forall t, j, \tag{28}$$

$$q \geq 0.$$

Proposition 7. *There exists an optimal solution* (θ^*, U^*) *to (**AP3**) where* $\sum_i a_{ij} U^*_{tij} \leq \sum_i a_{ij} U^*_{t+1,i,j}$ *for all* $t = 1, \ldots, \tau - 1, j$.

Proof. Let q^* denote an optimal solution to (**AD3**). Define $\hat{t}_i = \min\{t : c_i - \sum_{k=1}^{t-1} \sum_{j'} \lambda_{j'} q^*_{k,j'} < 1\}$ for each resource i, and $\hat{t} = \min_i \hat{t}_i$. By the definition of $\hat{t}_i$, constraint (27) is not tight before time $\hat{t}_i$ if $a_{ij} = 1$, and constraint (28) is not tight after time $\hat{t}$. From complementary slackness conditions, there exists an optimal solution (θ^*, U^*) to (**AP3**) such that

$$\theta^*_{t,j} = 0, \qquad \forall t \geq \hat{t}, j, \tag{29}$$

$$U^*_{tij} = 0, \qquad \forall t < \hat{t}_i, i, j. \tag{30}$$

Since U^* is nonnegative, it suffices to show that $\sum_i a_{ij} U^*_{tij}$ is monotone increasing in t for j and $t \geq \hat{t}$.

Using (29)–(30), constraint (26) can be written as

$$\theta^*_{t,j} + \lambda_j \sum_i a_{ij} \left(\sum_{k=t+1}^{\tau} \sum_{j'} a_{ij'} U^*_{kij'} \right) \geq \lambda_j f_j, \qquad \forall t = 1, \ldots, \hat{t} - 1, j, \tag{31}$$

$$\sum_i a_{ij} U^*_{tij} + \lambda_j \sum_i a_{ij} \left(\sum_{k=t+1}^{\tau} \sum_{j'} a_{ij'} U^*_{kij'} \right) \geq \lambda_j f_j, \qquad \forall t = \hat{t}, \ldots, \tau, j. \tag{32}$$

We next show that U^* can be adjusted to satisfy $\sum_i a_{ij}U^*_{tij} \le \sum_i a_{ij}U^*_{t+1,i,j}$ for $t = \hat{t}, \ldots, \tau$. To this end, suppose U^* is such that $\sum_i a_{ij'}U^*_{t'ij'} > \sum_i a_{ij'}U^*_{t'+1,i,j'}$ for some j' and $t' = \hat{t}, \ldots, \tau - 1$. Let U' equal U^* except that $U'_{t'ij'} = U^*_{t'+1,i,j'}$ and $U'_{t'+1,i,j'} = U^*_{t',i,j'}$ for all i. Clearly, (θ^*, U') and (θ^*, U^*) produces the same objective value in (**AP3**). It remains to show that (θ^*, U') is feasible. When U^* is replaced by U', it can be easily verified that (31) holds, and (32) holds for $t < t'$ and $t > t' + 1$.

Now, we show that (32) holds for $t = t'$ and $t = t' + 1$ with U^* replaced by U'. This entails showing that for each j,

$$\sum_i a_{ij}U'_{t'ij} + \lambda_j \sum_i a_{ij} \left(\sum_{k=t'+1}^{\tau} \sum_{j''} a_{ij''}U'_{kij''} \right) \ge \lambda_j f_j, \tag{33}$$

$$\sum_i a_{ij}U'_{t'+1,i,j} + \lambda_j \sum_i a_{ij} \left(\sum_{k=t'+2}^{\tau} \sum_{j''} a_{ij''}U'_{kij''} \right) \ge \lambda_j f_j. \tag{34}$$

For $j = j'$, the above inequalities become

$$\sum_i a_{ij'}U'_{t'ij'} + \lambda_{j'} \sum_i a_{ij'} \left(\sum_{k=t'+1}^{\tau} \sum_{j''} a_{ij''}U'_{kij''} \right) \ge \lambda_{j'} f_{j'}, \tag{35}$$

$$\sum_i a_{ij'}U'_{t'+1,i,j'} + \lambda_{j'} \sum_i a_{ij'} \left(\sum_{k=t'+2}^{\tau} \sum_{j''} a_{ij''}U'_{kij''} \right) \ge \lambda_{j'} f_{j'}. \tag{36}$$

To show (36), note that

$$\begin{aligned}
&\sum_i a_{ij'}U'_{t'+1,i,j'} + \lambda_{j'} \sum_i a_{ij'} \left(\sum_{k=t'+2}^{\tau} \sum_{j''} a_{ij''}U'_{kij''} \right) \\
&= \sum_i a_{ij'}U^*_{t',i,j'} + \lambda_{j'} \sum_i a_{ij'} \left(\sum_{k=t'+2}^{\tau} \sum_{j''} a_{ij''}U^*_{kij''} \right) \\
&\ge \sum_i a_{ij'}U^*_{t'+1,i,j'} + \lambda_{j'} \sum_i a_{ij'} \left(\sum_{k=t'+2}^{\tau} \sum_{j''} a_{ij''}U^*_{kij''} \right) \\
&\ge \lambda_{j'} f_{j'}.
\end{aligned}$$

In the above, the last inequality holds from (32) with $t = t' + 1$. To show (35), note that

$$\begin{aligned}
&\sum_i a_{ij'}U'_{t'ij'} + \lambda_{j'} \sum_i a_{ij'} \left(\sum_{k=t'+1}^{\tau} \sum_{j''} a_{ij''}U'_{kij''} \right) \\
&= \sum_i a_{ij'}U^*_{t'+1,i,j'} + \lambda_{j'} \sum_i a_{ij'} \left(\sum_{k=t'+2}^{\tau} \sum_{j''} a_{ij''}U^*_{kij''} \right) + \lambda_{j'} \sum_i a_{ij'} \sum_{j''} a_{ij''}U^*_{t',i,j''}
\end{aligned}$$

$$\geq \sum_i a_{ij'} U^*_{t'+1,i,j'} + \lambda_{j'} \sum_i a_{ij'} \left(\sum_{k=t'+2}^{\tau} \sum_{j''} a_{ij''} U^*_{kij''} \right)$$
$$\geq \lambda_{j'} f_{j'}.$$

In the above, the last inequality again follows from (32) with $t = t' + 1$. Similarly, we can show that (33) and (34) hold for $j \neq j'$. This completes the proof. ■

Proposition 7 immediately implies the following result, since $\sum_i W_{t,i} = \sum_i \sum_j a_{ij} U_{tij}$.

COROLLARY 1 **(Concavity of the Sum of Dynamic Bid-Prices)**. *There exists an optimal solution* (θ^*, W^*, U^*) *to (**AP2**) where* $\sum_i W^*_{t,i}$ *is monotonically increasing over time.*

Another special case is when each product uses exactly one resource. In this case, we can show that U^*_{tij} is monotone, implying that the dynamic bid-prices for each resource are concave.

COROLLARY 2 **(Monotonicity of U when each product uses exactly one resource)**. *There exists an optimal solution* (θ^*, U^*) *to (**AP3**) where* $U^*_{tij} \leq U^*_{t+1,i,j}$ *for all* $t = 1, \ldots, \tau - 1$ *and product* j *that uses only resource* i.

It is natural to ask whether dynamic bid-prices for each individual resource are concave. Unfortunately, the following counterexample shows that in general this is not true.

EXAMPLE 1 (COUNTEREXAMPLE TO CONCAVITY OF INDIVIDUAL DYNAMIC BID-PRICES). Consider a two-leg network with three products. Product 1 uses both resources, product 2 uses resource 1, and product 3 uses resource 2. The fares for the three products are \$200, \$100, and \$100, respectively. Three are three periods, each with arrival rates for the three products 0.2, 0.1, and 0.5, respectively. The optimal objective value of (**AP3**) for this problem is \$153.8. On the other hand, when W is forced to be monotonically increasing (i.e., bid-prices are forced to be concave), the objective value is \$155.2. This indicates that the optimal dynamic bid-prices are not concave.

Table 1 shows optimal solutions to the counterexample with and without the concavity constraints. Observe that in the optimal solution without concavity constraints, W^* is not monotone. When concavity constraints are enforced, W^* becomes monotone. However, the objective value increases from 153.8 to 155.2, implying that there is no alternative solution that gives concave dynamic bid-prices.

Example 1 is somewhat disappointing, as it showed the nonconcavity of dynamic bid-prices in general. It is natural to ask whether this nonconcavity will persist if the problem is scaled up. To this end, we solve a linearly scaled version of Example 1 where both the number of periods and capacity are scaled up linearly. We note this type of scaling is widely used in the RM literature

Period	w/o concavity			w/ concavity		
	V^*	W^*	U^*	V^*	W^*	U^*
1	(34,113.2)	(0,13.2)	$\begin{bmatrix} 0 & 0 & 0 \\ 13.2 & 0 & 0 \end{bmatrix}$	(20,114)	(0,0)	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
2	(34,100)	(24,10)	$\begin{bmatrix} 15 & 9 & 0 \\ 5 & 0 & 5 \end{bmatrix}$	(20,114)	(10,24)	$\begin{bmatrix} 1 & 9 & 0 \\ 19 & 0 & 5 \end{bmatrix}$
3	(10,90)	(10,90)	$\begin{bmatrix} 0 & 10 & 0 \\ 40 & 0 & 50 \end{bmatrix}$	(10,90)	(10,90)	$\begin{bmatrix} 0 & 10 & 0 \\ 40 & 0 & 50 \end{bmatrix}$

Table 1 Optimal solutions to Example 1 with and without concavity constraints

(Gallego and van Ryzin 1994, 1997, Talluri and van Ryzin 1998, Cooper 2002). With a scale-up factor of k, the total number of periods τ is replaced by $k\tau$, and the capacity vector c is replaced by kc. Table 2 shows the optimal objective values when Example 1 is linearly scaled up by a factor of k with and without concavity constraints. For $k \geq 4$, the two versions produce the same objective value. Therefore, it appears that enforcing concavity constraints is without loss of optimality in this example when k is sufficiently large.

k	w/o concavity	w/ concavity
1	153.8	155.2
2	347.2	350
3	540.6	542
4	734	734
5	925	925

Table 2 Optimal objective values when Example 1 is linearly scaled up by a factor of k with and without concavity constraints

How general is the observation made in this small experiment? The answer is that it is quite general. To this end, let z'_{AP3} be the objective function value of the formulation (**AP3**) with the additional concavity constraint $W_{t,i} \leq W_{t+1,i}$ for all $t = 1, \ldots, \tau - 1$ and i. Also, let z_{quasi} be the objective value of the quasi-static approximation $v_t(x) = \theta_t + \sum_i V_i x_i$ for all t and x. Note that quasi-static approximation requires V stays constant over time, and therefore is equivalent to the affine approximation with the additional restriction $W_{t,i} = 0$ for all t. Adelman (2007) shows that the quasi-static approximation produces a looser bound on the dynamic programming objective value than the affine approximation, which nevertheless is tighter than the objective value from DLP, which we denote by z_{LP}. Our discussion here implies that the following relationships among different bounds hold: $v_1(c) \leq z_{\text{AP3}} \leq z'_{\text{AP3}} \leq z_{\text{quasi}} \leq z_{\text{LP}}$.

It is known that the bound z_{LP} is asymptotically tight as both the number of periods and capacity are scaled up linearly; see Talluri and van Ryzin (1998) and Cooper (2002). Hence z_{AP3}, z'_{AP3}, and

z_{quasi} are asymptotically tight as well. Since optimal V from the quasi-static approximation satisfies concavity, this suggest that enforcing concavity constraints on V does not lead to big optimality loss when the problems are large. The asymptotic argument we give above is somewhat loose, as it does not directly relate the gap in bounds to the size of the problems. In our numerical section, we show that for many problem instances from the recent RM literature, the objective values with and without concavity constraints are the same, therefore providing an empirical support for our argument.

We point out that the argument in the previous paragraph applies equally well to the choice setting. However, a generalization of Proposition 7 to the choice case is not trivial and is not established in the current paper. Nevertheless, our numerical results in Section 6 shows that enforcing concavity constraints can dramatically improve computational performance in the choice setting and hence may still be desirable.

Another relevant question is what happens when arrival probabilities are non-stationary. Example 2 shows that stationarity is a necessary condition for Proposition 7. In fact, Proposition 7 does not hold even for the single-resource case when arrival probabilities are non-stationary.

Example 2 (Example with non-stationary arrival probabilities). Consider a single-leg problem with two products. The fares are \$100 and \$50, respectively. There are four periods and a single unit of capacity. The arrival rate for each product is 0.4 in the first two periods and 0.1 in the last two periods. The optimal dynamic bid-price vector is $[79.24, 65.40, 27.00, 15.00]^T$ and hence is not concave in time. The objective value is 79.24. In contrast, when concavity constraint is enforced, the optimal dynamic bid-price vector is $[81.538, 69.231, 46.154, 23.077]^T$ with corresponding objective value 81.538.

6. Computational Results

In this section, we report the results of a computational study. Our primary focus is to compare the performance of our dynamic disaggregation approach (**DD**) with solution methods from the literature. We consider test instances for network RM problems with or without choice. We consider two sets of test instances for problems without choice. The first set of test instances without choice is taken from Topaloglu (2009). The second set is randomly generated. The benchmark methods considered are the column generation approach (**CLG**) from Adelman (2007) applied to (**D1**) and the constraint generation approach (**CNG**) proposed in Tong and Topaloglu (2011) applied to the reduced program (**D2**). The test instances with choice are randomly generated. In this case, the benchmark method is the column generation approach (**CLG**) from Zhang and Adelman (2009)

applied to (**CD1**). Tests were performed on a Intel Quad Core Q9650 3.00GHZ computer running Windows 7 Professional 64-bit. The code were implemented using Visual C++ 2010 and CPLEX 12.3.

6.1. Computational Results for Network RM without Choice

The problem instances from Topaloglu (2009) consider hub-and-spoke networks; see Section 6 of Topaloglu (2009) for detailed descriptions of the these instances. Following Topaloglu (2009), each problem instance is indexed by (τ, N, ρ, κ), where τ is the total number of time periods, N is the number of spokes out of the hub, ρ is the load factor, and κ is the revenue ratio between the high fare and the low fare for each itinerary. Table 3 reports computational times for the three solution methods. We also reported the objective values for each problem instance. These values match the results reported in Topaloglu (2009).

Echoing the results reported in Tong and Topaloglu (2011), we found that **CNG** can be much faster than **CLG**. Our primary focus here is to compare the performance between **DD** and **CNG**. The results in the last two columns of Table 3 show that **DD** can be an order of magnitude faster than **CNG**. All problem instances are solved by **DD** within one second. Since a major drawback of LP-based ADP approaches is the solution time, our results show promises for practical applications.

There are several reasons for why **DD** is much faster than **CLG** and **CNG**. First, **DD** takes advantage of the special structure of the time-dependent bid prices, which can remain constant on a significant portion of the time horizon (Adelman 2007). As a result, **DD** works with much smaller problem formulations. Table 4 reports the initial and final columns and rows for the three approaches. Since **DD** starts with the deterministic linear programming formulation, the initial columns and rows are rather smaller. Often times, the algorithm does not need to disaggregate many columns to reach optimality. Therefore, the final columns and rows are often fairly small as well. In contrast, even though **CNG** works with a reduced formulation, it still need to populate an initial solution for each time, resource, and product. For this reason, the initial number of columns and rows can even be bigger than that of **CLG**. However, substantially fewer rows need to be generated in the solution process compared to **CLG**, leading to a considerable speedup.

Second, **CLG** starts with an initial feasible solution corresponding to "do nothing" (rejecting all demand), as it is trivial to establish the feasibility of such a solution. However, it is reasonable to expect that, in many cases, this initial solution is rather weak. In contrast, **DD** does not need to start with an initial feasible solution. Rather, the procedure starts with the DLP solution, which, although may not be feasible, can be much closer to optimal than the "do nothing" solution. We

note that the initial solution used in **CNG** also corresponds to the DLP solution; however, the solutions are generated from a much larger initial formulation.

Finally, **DD** appears to be better in dealing with multiple optimal solutions. Since the objective functions of ALP and the reduced program only depend on the aggregate acceptance of each product, there might be many alternative optimal solutions to them. This can hamper convergence, as a solution algorithm can oscillate too much among paths to different optimal solutions. Adelman (2007) showed that adding monotonicity of bid-prices over time can significantly speedup the solution of ALP, which can be viewed as one way to alleviate, but not completely eliminate, the issue. **DD**, on the other hand, actively deals with the multiplicity by looking at aggregate acceptance, while maintaining the resource constraints over time. Our numerical study shows that **DD** can solve many problem instances with much fewer iterations.

DD significantly expands the set of problem instances that can be solved within a reasonable time framework. To substantiate this point, we solve a set of randomly generated hub-and-spoke instances with 40 non-hub locations. There are 2000 periods and 880 products in total. Half of the non-hub locations have direct flights to the hub and the other half of non-hub locations have direct flights from the hub. These direct flights can be used to offer through itineraries going from one non-hub location to another. Each customer segment refers to itineraries with the same origin and destination. Each segment contains two products. We assume the arrival probability in each period is 0.9. The fares for local itineraries are generated from Poisson distributions with mean 100 and 300. The fares for through itineraries are 0.8 times the sum of the fares of the corresponding local itineraries.

Table 5 reports computational results for this set of problem instances. For the 20 randomly generated instances, the maximum solution time is 96 seconds. Note that we have chosen to solve the problem instances to optimality. The solution time can be further reduced if we impose an optimality tolerance. We also report the initial and final columns and rows across the 20 problem instances. Understandably, problems with longer solution times have larger numbers of final columns and rows, indicating more iterations of dynamic disaggregation. The solutions times reported here should be contrasted with existing results in the literature on similar problem instances (e.g., Adelman (2007)), where on average it takes hours to solve the problems to within 5% optimality.

6.2. Computational Results for Choice-based Network Revenue Management

We report computational results for randomly generated hub-and-spoke instances that involve customer choice. Half of the non-hub locations have direct flights to the hub and the other half of non-hub locations have direct flights from the hub. These direct flights can be used to offer through

itineraries going from one non-hub location to another. Each customer segment refers to itineraries with the same origin and destination. Each segment contains two products. Customer choice follows a multinomial logit (MNL) model with disjoint consideration sets. In MNL choice model, each product as well as the no-purchase option is given a choice weight. The choice weights for the two classes are generated from Poisson distributions with means 50 and 200, respectively. The no-choice weights are given by one plus a Poisson random variable with mean 10. We assume the arrival probability in each period is 0.9. The fares are also generated from Poisson distributions. For the first two customer segments, the fares for the two classes are given by Poisson distributions with means 30 and 10, respectively. For all other segments, the fares are given by Poisson distributions with means 300 and 100, respectively.

Because demand depends on the offer set, the notion of load factor is not immediately clear for network choice problems. We adopt the notation of nominal load factor from Zhang and Adelman (2009). In particular, given choice probability P, let $S^* \in \arg\max_{S\subseteq N} \sum_{j\in S} P_j(S) f_j$ be a revenue-maximizing set of open products when there is ample capacity of each resource. We call

$$\rho = \frac{\lambda \sum_{t=1}^{\tau} \sum_{j\in S^*} \sum_{i=1}^{m} a_{ij} P_j(S^*)}{\sum_{i=1}^{m} c_i} \tag{37}$$

the (nominal) load factor. In the no-choice setting $S^* = N$ and the load factor defined in (37) is the same as the one commonly used in the literature; see, e.g., Adelman (2007).

Table 6 reports computational results on a set of randomly generated problem instances. Each instance is indexed by a tuple; for example (100,8,8,48) means there are 100 periods, 8 resources, 8 segments, and 48 products. We report the nominal load factor for each problem instances in the second column. Since problems with choice can take considerably longer to solve, we solve the instances to 2% optimality.[3] We report the upper and lower bounds from **DD** when it reaches 2% optimality. All problem instances are solved within 7 minutes by **DD**. However, it can take much longer for **CLG** to solve the problems. For some larger instances, we stopped **CLG** at 7200 seconds if it had not converged by then. Overall, our results indicate that **DD** can lead to an order of magnitude speedup compared with **CLG**.

Table 7 reports computational results for choice-based instances when concavity constraints are enforced. We solve the problems to within 2% of optimality. We also terminate the procedure when 2% optimality is not reached within two hours (7200 seconds). Since the problems are not solved to

[3] We also tried other optimality gaps, such as 5% and 1%. At 5% optimality gap, **DD** converges very quickly, partially because it starts with a very strong initial solution corresponding to the one from CDLP (Liu and van Ryzin 2008), while it still takes quite long for **CLG** to converge. Note that 5% optimality gap were used in the computational study of Zhang and Adelman (2009). At 1% optimality gap, **DD** can still converge reasonably fast, while it takes **CLG** a long time to converge even for some of the small instances.

optimality, we report both upper and lower bounds when the solution procedure terminates. The bounds given by different algorithms are slightly different but very close. We choose to report the bounds from **DD**. Notice that adding concavity constraints does not lead to substantially different bounds. In fact, the magnitude of bound differences is much smaller than the optimality tolerance.

We also report the solution times with concavity constraints. **CLG** does not see improvement from adding concavity constraints. This is understandable, since the gain from tightening the problem formulation can be outweighed by the time increase due to the problem size increase in each iteration of column generation. However, adding concavity constraints leads to orders of magnitude improvement in the solution times for **DD**, achieving a speedup factor of more than 100 in some problem instances. Upon further investigation, this is mainly caused by substantially less time disaggregation steps in the **DD** algorithm. That is, by enforcing concavity constraints, an approximately optimal solution can be reached by a solution where the dynamic bid-prices stays constant for longer periods of time. Since the solution time of **DD** is approximately linear in the number of disaggregation steps, this drives dramatic reduction in solution times. Since solution time is one of the major concerns for applying choice-based models, our result can be crucial for practical applications.

7. Summary and Future Directions

We develop a dynamic disaggregation approach to solving the compact formulation of affine approximation for network revenue management with and without choice. In computational experiments, this approach usually takes a fraction of the time compared to other approaches proposed in the literature. Dynamic aggregation-disaggregation approaches for solving large-scale linear programs have received considerable attention in the early literature of linear programming, but have not been actively considered in the last two decades, partly due to the rapid development in computational power and commercial solvers. Solving huge linear programs in ADP, however, shows the value and necessity of such approaches even when we have access to very powerful computational facilities by today's standard. We believe there is still much to be done to explore such approaches for other ADP applications.

LP-based ADP views dynamic programs as equivalent linear programs and applies parametric approximation of the dynamic programming value functions. The resulting ALPs, however, often have significant degeneracy and many redundant constraints. One approach to tackling this issue is to add additional constraints in order to tighten the ALP formulations. Adelman (2007) shows that adding monotonicity constraints to an ALP formulation results in dramatic computational speedup for an affine approximation of the network revenue management problem without choice. Since

then, the monotonicity property has been exploited by almost all follow-up research. This paper further shows that dynamic bid-prices from the affine approximation satisfy stronger, second-order, structural properties. In particular, we show that the sum of dynamic bid-prices across resources is concave over time. We also give an example to show that dynamic bid-prices for individual resources may not be concave. Nevertheless, we argue that enforcing concavity constraints will not incur significant loss of optimality when the problems are large. We conduct a numerical study for network revenue management problems with and without choice. We show that adding concavity constraints leads to significant speedup in computations while incurring little or no optimality loss for network RM instances with choice. Our results suggest that, as a practical matter, it can be desirable to enforce concavity constraints.

Even though we focused on affine approximation, it is natural to consider more powerful approximation architectures, such as piecewise linear approximations. For approximation architectures that are refined enough, the behavior of bid-prices should be quite close to optimal bid-prices. If that is indeed the case, it is possible to enforce properties of optimal bid-prices on the approximation. Akan and Ata (2009) show that an ϵ-optimal bid-price process follows a martingale. Does it make sense to enforce this martingale property in a strong functional approximation, such as piecewise linear approximation?

As a final remark, we comment that adding additional constraints to tighten a linear programming formulation is a well-known technique in the mathematical programming community. Our research indicates that such techniques can be very powerful for the LP-based ADP. It is our sincere hope that more work would be done in this area.

Acknowledgment

We would like to thank Huseyin Topaloglu for the network RM data set provided on his website at `http://people.orie.cornell.edu/huseyin/research/rm_datasets/rm_datasets.html`.

References

Adelman, D. 2007. Dynamic bid-prices in revenue management. *Operations Research* **55**(4) 647–661.

Akan, M., B. Ata. 2009. Bid-price controls for network revenue management: Martingale characterization of optimal bid prices. *Mathematics of Operations Research* **34**(4) 912–936.

Berman, O., M. Hu, Z. Pang. 2012. On bid-price trends in revenue management. Working paper, Rotman School of Management, University of Toronto.

Cooper, W. L. 2002. Asymptotic behavior of an allocation policy for revenue management. *Operations Research* **50**(4) 720–727.

Gallego, G., G. Iyengar, R. Phillips, A. Dubey. 2004. Managing flexible products on a network. CORC Technical Report Tr-2004-01, IEOR Department, Columbia University.

Gallego, G., G. J. van Ryzin. 1994. Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Science* **40**(8) 999–1020.

Gallego, G., G. J. van Ryzin. 1997. A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research* **45**(1) 24–41.

Jasin, S., S. Kumar. 2012. A re-solving heuristic with bounded revenue loss for network revenue management with customer choice. *Mathematics of Operations Research* **37**(2) 313–345.

Kirshner, S., M. Nediak. 2012. Scalable dynamic bid prices for network revenue management in continuous time. Working paper, School of Business, Queens University.

Kunnumkal, S., H. Topaloglu. 2008. A refined deterministic linear program for the network revenue management problem with customer choice behavior. *Naval Research Logistics* **55**(6) 563–580.

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

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

Rogers, D., R. Plante, R. Wong, J. Evans. 1991. Aggregation and disaggregation techniques and methodology in optimization. *Operations Research* **39**(4) 553–582.

Schweitzer, P., M. Puterman, K. Kindle. 1985. Iterative aggregation-disaggregation procedures for discounted semi-markov reward processes. *Operations Research* **33**(3) 589–605.

Talluri, K., G. J. van Ryzin. 1998. An analysis of bid-price controls for network revenue management. *Management Science* **44**(11) 1577–1593.

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

Tong, C., H. Topaloglu. 2011. On approximate linear programming approach for network revenue management problems. Forthcoming, *INFORMS Journal on Computing*.

Topaloglu, H. 2009. Using Lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. *Operations Research* **57**(3) 637–649.

Vakhutinsky, I., L. Dudkin, A. Ryvkin. 1979. Iterative aggregation – a new approach to the solution of large-scale problems. *Econometrica* **47** 821–841.

Vossen, T., D. Zhang. 2013. Reductions of approximate linear programs for network revenue management. Working paper, Leeds School of Business, University of Colorado.

Xu, X., W. Hopp. 2009. Price trends in a dynamic pricing model with heterogeneous customers: A martingale perspective. *Operations Research* **57**(5) 1298–1302.

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

Zhang, D., W. L. Cooper. 2005. Revenue management for parallel flights with customer-choice behavior. *Operations Research* **53**(3) 415–431.

Instance	Obj. Value	**CLG** for (**D1**)	**CNG** for (**D2**)	**DD** for (**D3**)
(600,4,1.0,4)	32212.6	126.4540	0.2964	0.0156
(600,4,1.0,8)	51875.6	97.0950	0.2496	0.0000
(600,4,1.2,4)	29618.1	139.8860	0.2340	0.0156
(600,4,1.2,8)	49279.1	148.6380	0.2496	0.0000
(600,4,1.6,4)	26082.2	123.5840	0.2028	0.0000
(600,4,1.6,8)	45742.1	139.8700	0.1716	0.0000
(600,5,1.0,4)	33153.0	211.2720	0.3276	0.0156
(600,5,1.0,8)	53134.4	216.6230	0.3120	0.0000
(600,5,1.2,4)	31772.9	175.7820	0.3120	0.0156
(600,5,1.2,8)	51716.7	204.2680	0.2496	0.0156
(600,5,1.6,4)	28022.5	374.9950	0.4680	0.0156
(600,5,1.6,8)	47938.8	428.2700	0.2964	0.0156
(600,6,1.0,4)	26722.3	768.6950	0.8424	0.0312
(600,6,1.0,8)	42703.5	677.7460	0.4992	0.0156
(600,6,1.2,4)	24878.2	612.6160	0.7800	0.1248
(600,6,1.2,8)	40834.0	656.6710	0.4836	0.0156
(600,6,1.6,4)	21892.8	900.0790	0.5928	0.1404
(600,6,1.6,8)	37841.7	948.0650	0.4524	0.0312
(600,8,1.0,4)	23997.9	845.6190	2.3244	0.1404
(600,8,1.0,8)	38216.5	905.3670	1.5132	0.0468
(600,8,1.2,4)	22381.7	1854.7700	2.4180	0.3432
(600,8,1.2,8)	36580.0	1609.3200	1.0608	0.0780
(600,8,1.6,4)	19760.8	3063.8400	3.3540	0.9828
(600,8,1.6,8)	33942.5	2349.5500	1.1700	0.2028

Table 3 Computational times for hub-and-spoke instances without choice

Instance	**CLG** for (**D1**)		**CNG** for (**D2**)		**DD** for (**D3**)	
	Init. cols and rows	Final cols and rows	Init. cols and rows	Final cols and rows	Init. cols and rows	Final cols and rows
(600,4,1.0,4)	(5400,5400)	(16205,5400)	(21989,4800)	(21989,5392)	(41,8)	(201,264)
(600,4,1.0,8)	(5400,5400)	(17980,5400)	(21989,4800)	(21989,5392)	(41,8)	(121,136)
(600,4,1.2,4)	(5400,5400)	(22793,5400)	(21989,4800)	(21989,5400)	(41,8)	(201,264)
(600,4,1.2,8)	(5400,5400)	(23981,5400)	(21989,4800)	(21989,5390)	(41,8)	(121,136)
(600,4,1.6,4)	(5400,5400)	(23389,5400)	(21989,4800)	(21989,5354)	(41,8)	(201,264)
(600,4,1.6,8)	(5400,5400)	(22788,5400)	(21989,4800)	(21989,5333)	(41,8)	(121,136)
(600,5,1.0,4)	(6600,6600)	(23386,6600)	(31645,6000)	(31645,7147)	(61,10)	(241,310)
(600,5,1.0,8)	(6600,6600)	(22191,6600)	(31645,6000)	(31645,7147)	(61,10)	(121,110)
(600,5,1.2,4)	(6600,6600)	(26999,6600)	(31645,6000)	(31645,6835)	(61,10)	(241,310)
(600,5,1.2,8)	(6600,6600)	(25793,6600)	(31645,6000)	(31645,6830)	(61,10)	(181,210)
(600,5,1.6,4)	(6600,6600)	(33572,6600)	(31645,6000)	(31645,6971)	(61,10)	(301,410)
(600,5,1.6,8)	(6600,6600)	(32385,6600)	(31645,6000)	(31645,6936)	(61,10)	(181,210)
(600,6,1.0,4)	(7800,7800)	(40602,7800)	(42813,7200)	(42813,9346)	(85,12)	(421,588)
(600,6,1.0,8)	(7800,7800)	(46761,7800)	(42813,7200)	(42813,9344)	(85,12)	(253,300)
(600,6,1.2,4)	(7800,7800)	(43153,7800)	(42813,7200)	(42813,9186)	(85,12)	(673,1020)
(600,6,1.2,8)	(7800,7800)	(40175,7800)	(42813,7200)	(42813,9156)	(85,12)	(337,444)
(600,6,1.6,4)	(7800,7800)	(50341,7800)	(42813,7200)	(42813,9327)	(85,12)	(673,1020)
(600,6,1.6,8)	(7800,7800)	(45573,7800)	(42813,7200)	(42813,9207)	(85,12)	(337,444)
(600,8,1.0,4)	(10200,10200)	(39023,10200)	(70527,9600)	(70527,14486)	(145,16)	(865,1296)
(600,8,1.0,8)	(10200,10200)	(45002,10200)	(70527,9600)	(70527,14484)	(145,16)	(577,784)
(600,8,1.2,4)	(10200,10200)	(60604,10200)	(70527,9600)	(70527,14752)	(145,16)	(1009,1552)
(600,8,1.2,8)	(10200,10200)	(66591,10200)	(70527,9600)	(70527,14700)	(145,16)	(577,784)
(600,8,1.6,4)	(10200,10200)	(78025,10200)	(70527,9600)	(70527,14476)	(145,16)	(1297,2064)
(600,8,1.6,8)	(10200,10200)	(77373,10200)	(70527,9600)	(70527,14290)	(145,16)	(721,1040)

Table 4 Initial and final columns and rows of different solution approaches for hub-and-spoke instances without choice

Instance	Obj. value	Time	Init. cols and rows	Final cols and rows
1	630004	2.43	(881,40)	(5281,8440)
2	640055	70.54	(881,40)	(16721,30280)
3	641885	88.31	(881,40)	(17601,31960)
4	642560	0.76	(881,40)	(3521,5080)
5	630853	0.14	(881,40)	(1761,1720)
6	642013	0.11	(881,40)	(1761,1720)
7	652182	43.35	(881,40)	(14081,25240)
8	639378	47.80	(881,40)	(14961,26920)
9	640448	0.11	(881,40)	(1761,1720)
10	634847	33.54	(881,40)	(13201,23560)
11	641562	0.11	(881,40)	(1761,1720)
12	649383	10.72	(881,40)	(8801,15160)
13	636210	0.14	(881,40)	(1761,1720)
14	649139	2.45	(881,40)	(5281,8440)
15	630684	0.12	(881,40)	(1761,1720)
16	642369	96.00	(881,40)	(18481,33640)
17	647211	0.11	(881,40)	(1761,1720)
18	648966	33.06	(881,40)	(12321,21880)
19	636055	26.04	(881,40)	(11441,20200)
20	656112	0.16	(881,40)	(1761,1720)

Table 5 Computational performance of **DD** for 40 location hub-and-spoke network instances

Instance	Load factor	UB	LB	**CLG** for (**CD1**)	**DD** for (**CD3**)
(100,8,8,48)	1.40	14448.6	14205.7	20.4205	0.5928
(100,8,8,48)	1.45	12761.7	12511.1	14.9293	0.7956
(100,8,8,48)	1.41	14472.3	14233.1	16.2085	0.8112
(100,8,8,48)	1.38	14716.0	14459.7	17.4409	1.2480
(100,8,8,48)	1.39	14164.6	13912.9	15.3817	0.5304
(200,16,24,160)	1.44	34784.3	34127.6	440.8900	21.0445
(200,16,24,160)	1.41	35391.5	34726.1	490.3110	44.1171
(200,16,24,160)	1.43	34635.5	33943.7	492.4800	22.5109
(200,16,24,160)	1.45	34955.3	34277.6	447.8480	44.1483
(200,16,24,160)	1.43	33827.3	33211.5	474.3680	24.0866
(400,24,48,336)	1.42	74085.2	72673.6	*	160.6810
(400,24,48,336)	1.44	73199.6	71819.8	*	134.0200
(400,24,48,336)	1.44	72604.2	71258.6	*	224.4390
(400,24,48,336)	1.44	72317.9	70900.3	*	195.1880
(400,24,48,336)	1.43	74592.9	73188.2	*	325.8550
(800,32,80,576)	1.41	152223.0	149270.0	*	254.4220
(800,32,80,576)	1.38	153100.0	150164.0	*	191.3200
(800,32,80,576)	1.39	151556.0	148645.0	*	136.8440
(800,32,80,576)	1.43	148569.0	145844.0	*	455.5070
(800,32,80,576)	1.44	148674.0	145744.0	*	410.7820

Table 6 Computational times for hub-and-spoke instances with choice [* indicates a 2% gap was not reached within 7200 seconds]

Instance	UB	LB	**CLG** time	**DD** time	Diff. in bounds		Speedup factor	
					UB	LB	**CLG**	**DD**
(100,48,8,8)	14463.4	14201.0	27.0974	0.1404	14.8	-4.7	0.75	4.22
(100,48,8,8)	12766.6	12522.1	21.8869	0.2028	4.9	11.0	0.68	3.92
(100,48,8,8)	14456.5	14196.5	23.8838	0.1872	-15.8	-36.6	0.68	4.33
(100,48,8,8)	14728.9	14468.1	23.6186	0.3900	12.9	8.4	0.74	3.20
(100,48,8,8)	14191.0	13949.5	27.1130	0.3120	26.4	36.6	0.57	1.70
(200,160,16,24)	34858.7	34236.1	1609.2300	1.1856	74.4	108.5	0.27	17.75
(200,160,16,24)	35483.0	34776.6	2699.4400	0.8424	91.5	50.5	0.18	52.37
(200,160,16,24)	34681.5	34005.0	1253.7800	0.8892	46.0	61.3	0.39	25.32
(200,160,16,24)	35035.9	34408.6	1918.1700	3.6816	80.6	131.0	0.23	11.99
(200,160,16,24)	33900.0	33323.3	1629.4300	1.2636	72.7	111.8	0.29	19.06
(400,336,24,48)	74253.2	73077.1	*	2.4960	168.0	403.5	*	64.37
(400,336,24,48)	73401.8	72172.0	*	1.7316	202.2	352.2	*	77.40
(400,336,24,48)	72706.2	71547.2	*	2.8236	102.0	288.6	*	79.49
(400,336,24,48)	72386.4	71220.7	*	2.2152	68.5	320.4	*	88.11
(400,336,24,48)	74678.5	73498.2	*	2.4960	85.6	310.0	*	130.55
(800,576,32,80)	152509.0	150163.0	*	5.8968	286.0	893.0	*	43.15
(800,576,32,80)	153424.0	151177.0	*	3.6036	324.0	1013.0	*	53.09
(800,576,32,80)	151936.0	149639.0	*	2.8860	380.0	994.0	*	47.42
(800,576,32,80)	148940.0	146606.0	*	3.2448	371.0	762.0	*	140.38
(800,576,32,80)	148911.0	146627.0	*	6.5988	237.0	883.0	*	62.25

Table 7 Effect of enforcing concavity constraints for network revenue management problems with choice; "*" indicates that 2% optimality is not reached within 2 hours (7200 seconds)

Appendix A: Additional Proofs

Proof for Proposition 3

Associate dual variables $(\beta, \gamma, \phi, \eta)$ to the constraints in (**D3-α**). The dual of (**D3-α**) can be written as

$$
\begin{aligned}
(\textbf{D3-}\alpha\textbf{-Dual}) \quad & \min_{\beta,\gamma,\phi,\eta} \sum_i c_i \left(\beta_i + \sum_{t=\alpha+1}^{\tau} \sum_j a_{ij} \lambda_{t,j} \gamma_{tij} \right) + \sum_j \left(\Lambda_{\alpha,j} \phi_j + \sum_{t=\alpha+1}^{\tau} \lambda_{t,j} \eta_{t,j} \right) \\
& \phi_j + \sum_i a_{ij} \left(\beta_i + \sum_{t=\alpha+1}^{\tau} \sum_{j'} a_{ij'} \lambda_{t,j'} \gamma_{tij'} \right) \geq f_j, \qquad \forall j, \qquad (38) \\
& \eta_{t,j} + \sum_i a_{ij} \left(\beta_i + \gamma_{tij} + \sum_{k=t+1}^{\tau} \sum_{j'} a_{ij'} \lambda_{k,j'} \gamma_{kij'} \right) \geq f_j, \qquad \forall j, t = \alpha+1, \ldots, \tau, \qquad (39) \\
& \beta, \gamma, \phi, \eta \geq 0.
\end{aligned}
$$

By the definition of (V^*, θ^*) , the objective value for (**LP1**) is the same as the objective value of (**D3-α-Dual**), and hence is also the same as the objective value of (**D3-α**). Therefore, it suffices to show that (V^*, θ^*) is feasible for (**LP1**); i.e.,

$$
\theta^*_t - \theta^*_{t+1} + \sum_i \left[V^*_{t,i} x_i - V^*_{t+1,i} \left(x_i - \sum_j \lambda_{t,j} a_{ij} u_j \right) \right] \geq \sum_j \lambda_{t,j} f_j u_j, \qquad \forall t, x \in \mathcal{X}_t, u \in \mathcal{U}(x).
$$

The above inequality can be rewritten as

$$
\theta^*_t - \theta^*_{t+1} + \sum_i \left[(V^*_{t,i} - V^*_{t+1,i}) x_i - V^*_{t+1,i} \sum_j \lambda_{t,j} a_{ij} u_j \right] - \sum_j \lambda_{t,j} f_j u_j \geq 0, \qquad \forall t, x \in \mathcal{X}_t, u \in \mathcal{U}(x). \qquad (40)
$$

From (12)–(13),

$$
V^*_{t,i} - V^*_{t+1,i} = \begin{cases} 0, & \text{if } t = 1, \ldots, \alpha, \\ \sum_j a_{ij} \lambda_{t,j} \gamma^*_{tij}, & \text{if } t = \alpha+1, \ldots, \tau-1, \\ \beta^*_i + \sum_j a_{ij} \lambda_{\tau,j} \gamma^*_{\tau ij}, & \text{if } t = \tau, \end{cases} \qquad \forall t, i, \qquad (41)
$$

$$
\theta^*_t - \theta^*_{t+1} = \begin{cases} \sum_j \lambda_{t,j} \phi^*_j, & \text{if } t = 1, \ldots, \alpha, \\ \sum_j \lambda_{t,j} \eta^*_{t,j}, & \text{if } t = \alpha+1, \ldots, \tau, \end{cases} \qquad \forall t. \qquad (42)
$$

Next, we check that (40) holds using the definition of (V^*, θ^*) and (41)–(42).

Fix $x \in \mathcal{X}_t, u \in \mathcal{U}_t$. If $t = 1, \ldots, \alpha$,

$$
\begin{aligned}
& \theta^*_t - \theta^*_{t+1} + \sum_i \left[(V^*_{t,i} - V^*_{t+1,i}) x_i - V^*_{t+1,i} \sum_j \lambda_{t,j} a_{ij} u_j \right] - \sum_j \lambda_{t,j} f_j u_j \\
= & \sum_j \lambda_{t,j} \phi^*_j + \sum_i \left(\beta_i + \sum_{k=\alpha+1}^{\tau} \sum_{j'} a_{ij'} \lambda_{k,j'} \gamma^*_{kij'} \right) \sum_j \lambda_{t,j} a_{ij} u_j - \sum_j \lambda_{t,j} f_j u_j
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_j \lambda_{t,j}\phi_j^*(1-u_j) + \sum_j \lambda_{t,j}u_j \left[\phi_j^* + \sum_i a_{ij}\left(\beta_i + \sum_{k=\alpha+1}^{\tau}\sum_{j'} a_{ij'}\lambda_{k,j'}\gamma_{kij'}^*\right) - f_j\right] \\
&\geq 0.
\end{aligned}
$$

In the above, the last inequality follows since $1-u_j \geq 0$ for all j and the term in the square bracket is nonnegative from (38).

If $t = \alpha+1, \ldots, \tau-1$,

$$
\begin{aligned}
&\theta_t^* - \theta_{t+1}^* + \sum_i \left[(V_{t,i}^* - V_{t+1,i}^*)x_i - V_{t+1,i}^* \sum_j \lambda_{t,j}a_{ij}u_j\right] - \sum_j \lambda_{t,j}f_j u_j \\
&= \sum_j \lambda_{t,j}\eta_{t,j}^* + \sum_i \left[\sum_j a_{ij}\lambda_{t,j}\gamma_{tij}^* x_i + \left(\beta_i^* + \sum_{k=t+1}^{\tau}\sum_{j'} a_{ij'}\lambda_{k,j'}\gamma_{kij'}^*\right)\sum_j \lambda_{t,j}a_{ij}u_j\right] - \sum_j \lambda_{t,j}f_j u_j \\
&= \sum_j \lambda_{t,j}\eta_{t,j}^*(1-u_j) + \sum_i \sum_j a_{ij}\lambda_{t,j}\gamma_{tij}^*(x_i - u_j) \\
&\quad + \sum_j \lambda_{t,j}u_j \left[\eta_{t,j}^* + \sum_i a_{ij} + \left(\beta_i^* + \gamma_{tij}^* + \sum_{k=t+1}^{\tau}\sum_{j'} a_{ij'}\lambda_{k,j'}\gamma_{kij'}^*\right) - f_j\right] \\
&\geq 0.
\end{aligned}
$$

In the above, the last inequality follows since $1-u_j \geq 0$ for all j, $a_{ij}(x_i - u_j) \geq 0$ for all i, j, and the term in the square bracket is nonnegative from (39).

If $t = \tau$,

$$
\begin{aligned}
&\theta_t^* - \theta_{t+1}^* + \sum_i \left[(V_{t,i}^* - V_{t+1,i}^*)x_i - V_{t+1,i}^* \sum_j \lambda_{t,j}a_{ij}u_j\right] - \sum_j \lambda_{t,j}f_j u_j \\
&= \sum_j \lambda_{\tau,j}\eta_{\tau,j}^* + \sum_i \left(\beta_i^* + \sum_j a_{ij}\lambda_{\tau,j}\gamma_{\tau ij}^*\right)x_i - \sum_j \lambda_{t,j}f_j u_j \\
&= \sum_j \lambda_{\tau,j}\eta_{\tau,j}^*(1-u_j) + \sum_i \beta_i^* \left(x_i - \sum_j a_{ij}\lambda_{\tau,j}u_j\right) + \sum_i \sum_j a_{ij}\lambda_{\tau,j}\gamma_{\tau ij}^*(x_i - u_j) \\
&\quad + \sum_j \lambda_{\tau,j}u_j \left[\eta_{\tau,j}^* + \sum_i a_{ij} + \left(\beta_i^* + \gamma_{\tau ij}^*\right) - f_j\right] \\
&\geq 0.
\end{aligned}
$$

In the above, the last inequality can be similarly established. In particular, the term in the square bracket is nonnegative from (39). This verifies (40) and completes the proof.