Reductions of 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. We show that the ALPs can be dramatically reduced in size for both affine and separable piecewise linear approximations to network revenue management problems, under both independent and discrete choice models of demand. Our key result is the equivalence between each ALP and a corresponding reduced program, which is more compact in size and admits an intuitive probabilistic interpretation. For the affine approximation to network revenue management under an independent demand model, we recover an equivalence result known in the literature, but provide an alternative proof. Our other equivalence results are new. We test the numerical performance of solving the reduced programs directly using off-the-shelf commercial solvers on a set of test instances taken from the literature.

This version: September 27, 2015

1. Introduction

Network revenue management (NRM) problems can be broadly viewed as sequential decision making problems under uncertainty and are often formulated as dynamic programs (Gallego and van Ryzin 1997, Talluri and van Ryzin 1998). In the canonical airline application of NRM, the state of the system is the vector of remaining resources, where each resource corresponds to a flight leg. Since practical airline NRM problems usually involve a large number of flight legs, the dynamic programming formulation suffers from the well-known "curse of dimensionality." Dealing with this curse of dimensionality through approximations and heuristic control policies has been the focus of much of the research in NRM over the last two decades (Talluri and van Ryzin 1998, Bertsimas and Popescu 2003).

The seminal paper of Adelman (2007) introduces a solution framework based on equivalent linear programming formulations of the corresponding dynamic programs. His work builds on the stream of literature on linear programming-based approximate dynamic programming (LP-based ADP)

* Authors are listed alphabetically. Both authors contributed equally.

(Schweitzer and Seidmann 1985, de Farias and Van Roy 2003, de Farias and Van Roy 2004). The central idea is to approximate the value function with linearly weighted basis functions. Adelman (2007) illustrates his idea by implementing a functional approximation where the value function is approximated by an affine function of the state (resource) vector. The resulting approximate linear programs (ALPs) have a relatively small number of variables but a huge number of constraints, which grow exponentially in the number of resources and the number of products. He shows that the affine approximation produces an upper bound on total expected revenue, which is tighter than the upper bound from the widely used deterministic linear program (DLP) (Talluri and van Ryzin 1998, Cooper 2002). The coefficients of each resource in the affine approximation can be interpreted as time-dependent bid-prices, that is, per-unit values attached to each resource in each period. A bid-price control accepts a customer request if the revenue earned exceeds the total value of the resources consumed. Adelman (2007) shows that the dynamic bid-price control policy is superior to the static bid-price policies obtained using DLP, even when the latter is frequently resolved. Zhang and Adelman (2009) study an extension to NRM with discrete choice models of customer demand (Talluri and van Ryzin 2004, Gallego et al. 2004, Zhang and Cooper 2005, Liu and van Ryzin 2008).

Adelman's work inspired the development of stronger functional approximations, whose corresponding ALPs yields tighter upper bounds than the affine approximation. Even though tighter bounds do not guarantee stronger heuristic policies, numerical studies have found positive correlations between the two (Talluri 2008). A powerful and intuitively appealing functional approximation is the separable piecewise linear approximation, where the basis functions are separable by resource. Separable piecewise linear approximations have been used in many applications; see, e.g., Bertsekas and Tsitsiklis (1996) and Powell (2007). For NRM, this approximation has recently been used by Farias and Van Roy (2007) and Meissner and Strauss (2012). Instead of a time-dependent bid-price for each resource from the affine approximation, the separable piecewise linear approximation leads to bid-prices that depend on both time and resource levels. Naturally, the resulting ALPs are much larger in comparison to the ALPs from the affine approximation, and therefore also tend to be harder to solve.

Both the affine and the separable piecewise linear approximations lead to large-scale linear programs that are computationally challenging, even with powerful modern linear programming tools. For example, Table 6 in Meissner and Strauss (2012) shows that solving a small NRM problem with separable piecewise linear approximation can take more than 10 hours. Two standard approaches to tackling these computational challenges are column generation (Adelman 2007, Zhang and Adelman 2009, Meissner and Strauss 2012) and constraint sampling (de Farias and Van Roy 2004, Farias and Van Roy 2007). The fundamental idea in both approaches is to successively solve smaller versions of the ALPs, because brute force solutions are computationally intractable.

In this paper, we show that the ALPs under affine and separable piecewise linear approximations for NRM can be reduced to linear programs that are much smaller in size, for both independent and discrete choice models of demand. Under the independent demand model, the reduced programs grow linearly in the numbers of resources and products. However, under the discrete choice model, the reduced programs for both affine and separable piecewise linear approximations grow linearly in the number of resources, but still grow exponentially in the number of products. Hence, the reduction breaks the "curse of dimensionality" in the state space, but not in the action space. On an intuitive level, this is due to the fact that products cannot be tracked independently when customers choose among them, while they can be tracked independently when each arriving customer requests a specific product.

Our analytical framework for deriving the main results has its roots in the Dantzig-Wolfe decomposition principle (Dantzig and Wolfe 1960, 1961). Essentially, we show that the ALPs under consideration are equivalent to the Dantzig-Wolfe reformulations of some smaller linear programs, which we call the reduced programs. In the classical Dantzig-Wolfe decomposition, a linear program is "expanded" by considering the explicit representation of the polyhedron defined by a subset of the constraints. The development in our paper is quite different, in that the starting point is an ALP that corresponds to an expanded formulation. To construct a reduced formulation, the question is then whether and how we can recover constraints whose expansion results in the ALP. We rely on the structure of the ALP's column generation subproblem to generate these constraints.

An important concept in dynamic programming, and in approximate dynamic programming by extension, is the notion of state-action pairs and their corresponding probabilities (Puterman 1994). In particular, the decision variables in ALPs correspond to state-action pairs, and their values can be interpreted as approximate state-action probabilities. This correspondence plays an important role in our theoretical development. To show that the Dantzig-Wolfe reformulations of some reduced programs can be directly related to the ALPs, we need to show that the columns in the reformulations can be properly labeled by state-action pairs. This requirement entails showing that the underlying polyhedra have integer extreme points. We show that the required integrality property holds for all cases considered in the paper.

Several authors consider compact representations for ALPs in the literature. Farias and Van Roy (2007) consider a relaxed ALP (called rALP), which is shown to be equivalent to the original ALP under the affine approximation for NRM with independent demand. For the separable piecewise linear approximation, rALP provides a feasible solution to the original ALP. They use numerical experiments to demonstrate the solution quality of rALP. Unlike the reductions considered in this paper, rALP formulations are still exponential in size for NRM under the independent demand model. More recently, Tong and Topaloglu (2011) show great promise, both theoretically and computationally, in reducing ALPs from affine approximations under the independent demand model. They prove that the affine ALP in Adelman (2007) can be reduced to a more compact linear program, which grows linearly, rather than exponentially, in model primitives (number of resources, number of products, etc.). Via numerical experiments, they illustrate that the reduced program can be solved orders of magnitude faster than the original ALP. The significant reduction in computation times is not unexpected given the dramatic reduction of program size. The reduction proof in Tong and Topaloglu (2011) relates the dual of the affine ALP to a network flow problem. For the affine ALP with independent demand, our reduced program coincides with the one proposed in Tong and Topaloglu (2011), though we offer an alternative framework to prove the result. Neither Farias and Van Roy (2007) nor Tong and Topaloglu (2011) consider NRM with a discrete choice model of customer demand.

The reduced programs admit an intuitive interpretation. For the separable piecewise linear approximation, the decision variables can be interpreted as *marginal* state-action probabilities, the dynamics of which are tracked by the constraints. This should be contrasted with the full equivalent linear programming formulation of the dynamic programming formulation, which tracks the *joint* probabilities of state-action pairs, and the affine ALP, which tracks expectations.

The ALP reductions introduced in this paper open up the possibility to develop specialized algorithms based on the more compact problem representations. This is precisely the approach taken in Vossen and Zhang (2014). They show that the reduced programs for affine ALPs for both discrete choice and independent demand models can be efficiently solved by a dynamic aggregation/disaggregation procedure. However, their work does not consider separable piecewise linear approximations. Our numerical experiments suggest that the reduced programs for separable piecewise linear approximations can be solved much faster than existing column generation approaches proposed in the literature. This result is not unexpected, because the reduced programs are much smaller in size.

Lagrangian relaxation is considered a distinctive approach from the LP-based ADP. For a class of infinite horizon stochastic dynamic programs, Adelman and Mersereau (2008) compare the bounds and policy performance of Lagrangian relaxation and LP-based ADP. Topaloglu (2009) proposes a clever Lagrangian relaxation approach for NRM with independent demand, which relaxes the requirement that a customer traveling multiple legs has to be accepted on all legs in the itinerary. He introduces a subgradient approach to solve the relaxation and reports encouraging bounds and policy performance from the approach. Kunnumkal and Talluri (2011) show that Lagrangian relaxation and separable piecewise linear approximation are equivalent in that they produce the same upper bound.

The reduced programs for separable piecewise linear approximation, both with and without choice, can be used as a starting point to derive a Lagrangian relaxation. For the independent demand case, relaxing a linking constraint in the reduced formulation recovers the Lagrangian relaxation in Topaloglu (2009). The equivalence result established in Kunnumkal and Talluri (2011) is immediate since the reduced program is a linear program. A Lagrangian relaxation in the choice case can also be derived following a similar approach. However, the number of linking constraints relaxed in that setting is exponential in the number of products. Hence the number of corresponding Lagrangian multipliers is also exponential in the number of products (Kunnumkal and Talluri 2014).

The equivalence result in Kunnumkal and Talluri (2011) establishes the subgradient algorithm proposed in Topaloglu (2009) as an alternative approach to solving the separable piecewise linear approximation for NRM with independent demand. Topaloglu (2009) shows the efficacy of the approach in an extensive numerical study. More recently, Kunnumkal and Talluri (2011) observe in numerical tests that the subgradient algorithm is much faster than solving the separable piecewise linear ALP using a column generation approach. In our numerical study, we solve the reduced programs directly on the test instances taken from Topaloglu (2009). We show that solving the reduced programs leads to tighter upper bounds on total expected revenues. We conjecture that this is caused by the fact that the subgradient algorithm used in Topaloglu (2009) can terminate prematurely, since it relies on an ad-hoc stopping criterion. On the other hand, both upper and lower bounds on the objective values can be constructed when solving the reduced programs directly. Hence, a solution can be obtained for any desired optimality tolerance. Our implementation uses the default interior point solver in CPLEX and therefore requires minimal custom coding and algorithm tuning.

The remainder of the paper is organized as follows. Section 2 formulates the NRM problem and introduces some technical preliminaries. Section 3 introduces a general framework for reducing the size of multi-stage linear programs. Sections 4 and 5 consider reductions for affine and separable piecewise linear approximations, respectively. Section 6 reports numerical results and Section 7 concludes. An appendix at the end contains a summary of notation and additional proofs.

2. Model Formulation and Preliminaries

For ease of exposition, we use airline terminology throughout the paper. In particular, we have a set of flight legs that can be used to serve customers who arrive over time. The time horizon is finite and discrete, and at the start of each time period we need to decide which itinerary-fare combinations (products) to offer to the customers. Customers review the offered products and purchase at most one of them. The overall objective is to determine which products to offer so as to maximize total expected revenue.

To be more precise, we consider a flight network that consists of legs in the set $\mathcal{I}=\{1,\ldots,I\}$, where I is the number of legs. The capacity of each leg is given by the vector $\mathbf{c}=(c_1,\ldots,c_I)$, where c_i is the capacity of leg i. The products that are offered belong to the set $\mathcal{J}=\{1,\ldots,J\}$, where J is the number of products and the fare component of product j is denoted f_j. Products can differ in the legs they use, which is expressed using a consumption matrix. The consumption matrix is an $(I\times J)$-matrix $\mathbf{A}\equiv(a_{ij})$, where the entry $a_{ij}\in\{0,1\}$ represents whether leg i is required by product j. We use $\mathbf{a}^j$ to represent the j-th column of $\mathbf{A}$, which is the incidence vector for product j, and $\mathcal{I}_j=\{i\in\mathcal{I}:a_{ij}=1\}$ to represent the set of legs used by product j. The time periods belong to the set $\mathcal{T}=\{1,\ldots,T\}$ and time periods are counted forward, so period T corresponds to the last period in the horizon. To simplify notation, we reserve the symbols i, j, and t for legs, products, and time, and omit writing the corresponding index sets.

In each period t, there is one customer arrival with probability ρ_t, and no customer arrival with probability $1-\rho_t$. We consider different versions of the problem, both with discrete choice and independent demand models of customer behavior (Liu and van Ryzin 2008). When accommodating customer choice behavior, we assume that the arriving customer chooses product j with probability $P_{t,j}(\mathbf{u})$, where $\mathbf{u}\in\mathcal{U}\equiv\{0,1\}^J$ corresponds to the characteristic vector of the set of products currently being offered; with slight abuse of terminology, we sometimes refer to this vector as an offer set in the remainder of this paper. Under an independent demand model, we assume that each arriving customer belongs to a customer class requesting a specific product. In other words, a class-j customer will only request product j, and we use $\lambda_{t,j}$ for the probability that an arriving customer requests product j during period t. Note that the latter can be viewed as a special case of the discrete choice model where

$$P_{t,j}(\mathbf{u})=\frac{\lambda_{t,j}}{\rho_t}u_j,\qquad \forall t,j,\mathbf{u}\in\mathcal{U}. \tag{1}$$

The resulting problem has been studied in a number of papers in the recent years (Gallego et al. 2004, Liu and van Ryzin 2008), and can be cast as a finite-horizon discrete-time Markov decision process. The state in this MDP corresponds to an I-dimensional vector $\mathbf{x}$ that specifies the remaining capacity on each of the flight legs at the beginning of a period, and we require $\mathbf{x} \in \mathcal{X} \equiv \{\mathbf{x} \in \mathbb{Z}_+^I : \mathbf{x} \le \mathbf{c}\}$. A product can only be offered when there is enough capacity remaining on each of the legs in its itinerary. For any state $\mathbf{x} \in \mathcal{X}$, we define its feasible offer sets as $\mathcal{U}(\mathbf{x}) = \{\mathbf{u} \in \mathcal{U} : \mathbf{a}^j u_j \le \mathbf{x} \ \forall j\}$. These represent the action space in the MDP, that is, in each state we select exactly one offer set. Now, let $v_t(\mathbf{x})$ denote the total expected revenue over periods $t, \ldots, T$ starting from state $\mathbf{x}$ at the beginning of period t. The optimality equations are given by

$$\begin{aligned} v_t(\mathbf{x}) &= \max_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \left\{ \sum_j \rho_t P_{t,j}(\mathbf{u})[f_j + v_{t+1}(\mathbf{x} - \mathbf{a}^j)] + \left(1 - \rho_t \sum_j P_{t,j}(\mathbf{u})\right) v_{t+1}(\mathbf{x}) \right\}, \qquad \forall t, \mathbf{x} \in \mathcal{X}, \\ &= \max_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} \left\{ \sum_j \rho_t P_{t,j}(\mathbf{u}) \left[f_j - \left(v_{t+1}(\mathbf{x}) - v_{t+1}(\mathbf{x} - \mathbf{a}^j)\right)\right] + v_{t+1}(\mathbf{x}) \right\}, \qquad \forall t, \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{2}$$

where the boundary conditions are $v_{\tau+1}(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$.

Solving (2) is computationally challenging due to the state space explosion. In addition, optimization over $\mathbf{u}$ for a given state can be difficult for general choice probabilities. In this paper, we consider LP-based ADP approaches to solving the problem (Adelman 2007, Zhang and Adelman 2009).

2.1. Approximate Dynamic Programming

Following Adelman (2007), we start with the equivalent linear programming formulation for (2). Using decision variables $v_t(\mathbf{x})$ for all $t, \mathbf{x} \in \mathcal{X}$, and defining the set of feasible state-action pairs as $\mathcal{S} = \{(\mathbf{x}, \mathbf{u}) \in \mathcal{X} \times \mathcal{U} : \mathbf{u} \in \mathcal{U}(\mathbf{x})\}$ yields the following formulation:

$$\begin{aligned} (\mathbf{P}) \ &\min_{\{v_t(\cdot)\}_{\forall t}} \ v_1(\mathbf{c}) \\ &\text{s.t.} \quad v_t(\mathbf{x}) \ge \sum_j \rho_t P_{t,j}(\mathbf{u}) \left[f_j - \left(v_{t+1}(\mathbf{x}) - v_{t+1}(\mathbf{x} - \mathbf{a}^j)\right)\right] + v_{t+1}(\mathbf{x}), \quad \forall t, (\mathbf{x}, \mathbf{u}) \in \mathcal{S}. \end{aligned}$$

Its dual equals

$$\begin{aligned} (\mathbf{D}) \ &\max_{\mathbf{p}} \sum_{t, (\mathbf{x}, \mathbf{u}) \in \mathcal{S}} \left(\sum_j \rho_t P_{t,j}(\mathbf{u}) f_j \right) p_{t,\mathbf{x},\mathbf{u}} \\ &\text{s.t.} \sum_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} p_{t,\mathbf{x},\mathbf{u}} \ = \begin{cases} \mathbb{1}\{\mathbf{x} = \mathbf{c}\}, & \text{if } t = 1, \\ \displaystyle\sum_{\mathbf{u} \in \mathcal{U}(\mathbf{x})} p_{t-1,\mathbf{x},\mathbf{u}} - \sum_{\mathbf{u} \in \mathcal{U}(\mathbf{x}), j} \rho_{t-1} P_{t-1,j}(\mathbf{u}) \left(p_{t-1,\mathbf{x},\mathbf{u}} - p_{t-1,\mathbf{x}+\mathbf{a}^j,\mathbf{u}}\right), & \text{if } t > 1, \end{cases} \\ &\hspace{28em} \forall t, \mathbf{x} \in \mathcal{X}, \quad (3) \\ &\qquad\quad \mathbf{p} \ \ge \mathbf{0}. \end{aligned}$$

In the above, $\mathbb{1}\{\cdot\}$ is the indicator function. We also assume $p_{t,\mathbf{x},\mathbf{u}} = 0$ if $(\mathbf{x}, \mathbf{u}) \notin \mathcal{S}$, to simplify the presentation of constraint (3). The decision variables in the dual formulation can be interpreted as state-action probabilities; that is, $p_{t,\mathbf{x},\mathbf{u}}$ equals the fraction of time we reach state $\mathbf{x}$ at the beginning of period t and offer the set of products corresponding to $\mathbf{u}$ during the period. Thus, the term $\sum_{\mathbf{u}\in\mathcal{U}(\mathbf{x})} p_{t,\mathbf{x},\mathbf{u}}$ can be interpreted as the probability of reaching state $\mathbf{x}$ at the beginning of period t, and constraints (3) can be interpreted as flow balance constraints that ensure the evolution of the state distribution is maintained correctly over time.

In and of itself, however, the formulation as a linear program does little to improve tractability: the number of variables and constraints in ($\mathbf{P}$) increases exponentially in the number of legs I and the number of products J, so brute-force solution of ($\mathbf{P}$) is as difficult as solving the dynamic programming formulation (2) directly. To achieve tractability, it is common to resort to approximately solving ($\mathbf{P}$). Typically, this is done by representing the value function $v_t(\mathbf{x})$ by a collection of weighted basis functions. Consider a set of basis functions $\phi_b : \mathcal{X} \to \mathbb{R}$ for $b \in \mathcal{B}$, where $\mathcal{B}$ is some index set, and take

$$v_t(\mathbf{x}) \approx \theta_t + \sum_{b\in\mathcal{B}} V_{t,b}\phi_b(\mathbf{x}), \quad \forall t, \mathbf{x} \in \mathcal{X}, \tag{4}$$

where $V_{t,b}$ is a parameter that weighs basis function $\phi_b(\cdot)$ at time t, and θ_t is a constant offset. While θ_t corresponds to a constant basis function $\phi_\emptyset(\mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}$, we include it here for ease of exposition in later sections. Substituting (4) into ($\mathbf{P}$) yields an optimization problem over the parameters θ_t and $V_{t,b}$, and leads to the following linear program:

$$\begin{aligned}
&(\mathbf{P})^\phi \\
&\min_{\theta,V}\ \theta_1 + \sum_{b\in\mathcal{B}} V_{1,b}\phi_b(\mathbf{c}) \\
&\text{s.t.}\quad \theta_t - \theta_{t+1} + \sum_{b\in\mathcal{B}} (V_{t,b} - V_{t+1,b})\phi_b(\mathbf{x}) + \sum_{b\in\mathcal{B},j} \rho_t P_{t,j}(\mathbf{u}) V_{t+1,b}\left(\phi_b(\mathbf{x}) - \phi_b(\mathbf{x}-\mathbf{a}^j)\right) \geq \sum_j \rho_t P_{t,j}(\mathbf{u}) f_j \\
&\hspace{30em} \forall t, (\mathbf{x},\mathbf{u}) \in \mathcal{S}.
\end{aligned}$$

Its dual equals

$$\begin{aligned}
(\mathbf{D})^\phi\ \max_{\mathbf{p}} & \sum_{t,(\mathbf{x},\mathbf{u})\in\mathcal{S}} \left(\sum_j \rho_t P_{t,j}(\mathbf{u}) f_j \right) p_{t,\mathbf{x},\mathbf{u}} \\
\text{s.t.} \sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} \phi_b(\mathbf{x}) p_{t,\mathbf{x},\mathbf{u}} &= \begin{cases} \phi_b(\mathbf{c}), & \text{if } t = 1, \\ \displaystyle\sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} \phi_b(\mathbf{x}) p_{t-1,\mathbf{x},\mathbf{u}} - \\ \displaystyle\sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S},j} \rho_{t-1} P_{t-1,j}(\mathbf{u}) \left(\phi_b(\mathbf{x}) - \phi_b(\mathbf{x}-\mathbf{a}^j)\right) p_{t-1,\mathbf{x},\mathbf{u}}, & \text{if } t > 1, \end{cases} \\
& \hspace{20em} \forall t, b \in \mathcal{B}, \quad (5) \\
\sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} p_{t,\mathbf{x},\mathbf{u}} &= \begin{cases} 1, & \text{if } t = 1, \\ \displaystyle\sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} p_{t-1,\mathbf{x},\mathbf{u}}, & \text{if } t > 1, \end{cases} \qquad \forall t, \quad (6)
\end{aligned}$$

$$\mathbf{p} \;\geq \mathbf{0}.$$

Note that constraint (6) can be simplified to

$$\sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} p_{t,\mathbf{x},\mathbf{u}} \;=\; 1 \qquad \forall t. \tag{7}$$

The decision variables $p_{t,\mathbf{x},\mathbf{u}}$ in the dual formulation $(\mathbf{D})^{\phi}$ can be interpreted as *approximate* state-action probabilities. The probabilities are approximate since they may not be the same as those corresponding to an optimal policy. We refer to Adelman (2007) for further discussion. The term $\sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} \phi_b(\mathbf{x}) p_{t,\mathbf{x},\mathbf{u}}$ in (5) can be interpreted as the expected value of the basis functions at the beginning of period t. Thus, whereas the flow balance constraints in $(\mathbf{D})$ track the evolution of the state distribution, the flow balance constraints in $(\mathbf{D})^{\phi}$ only require that the expectations of the basis functions with respect to the state distribution are maintained over time.

In comparison to $(\mathbf{D})$, the dual problem $(\mathbf{D})^{\phi}$ has relatively few constraints when a moderate number of basis functions are used, though the number of decision variables is still exponential in both the number of legs and the number of products. This suggests a column generation procedure (Desrosiers and Lübbecke 2005) to solve the resulting problem, which uses only a small subset of the decision variables and adds more variables only when needed. This approach has received considerable attention in the NRM literature (Adelman 2007, Zhang and Adelman 2009, Meissner and Strauss 2012).

While the approximation (4) allows many potential approximation alternatives, in the remainder of this paper we focus on two specific approximation architectures: the *affine* approximation and the *separable piecewise linear* approximation. The affine approximation is given by

$$v_t(\mathbf{x}) \approx \theta_t + \sum_i V_{t,i} x_i, \qquad \forall t, \mathbf{x} \in \mathcal{X}, \tag{8}$$

while the separable piecewise linear approximation is given by

$$v_t(\mathbf{x}) \approx \theta_t + \sum_i \sum_{k=1}^{c_i} V_{t,i,k} \mathbb{1}\{x_i \geq k\}, \qquad \forall t, \mathbf{x} \in \mathcal{X}. \tag{9}$$

In light of our previous discussion, we emphasize their underlying interpretation. Because the affine approximation uses basis functions $\phi_i(\mathbf{x}) = x_i$ for all i, it ensures that the expected number of remaining units for each leg is enforced consistently over time. The separable piecewise linear approximation, on the other hand, uses basis functions $\phi_{i,k}(\mathbf{x}) = \mathbb{1}\{x_i \geq k\}$ for all $i, k = 1, \ldots, c_i$ and therefore tracks the probability that the remaining capacity of leg i is at least k. Thus, approximation (9) enforces time consistency on the marginal distributions of capacity remaining for each leg.

Both approximations have been studied in connection with the NRM problem (Adelman 2007, Zhang and Adelman 2009, Farias and Van Roy 2007, Meissner and Strauss 2012), and the heuristic policies from these approximations are known to give strong revenue performance. Approximation (9), in particular, has also been a popular approximation architecture in the approximate dynamic programming literature in other contexts (Bertsekas and Tsitsiklis 1996, Powell 2007).

3. General Framework

Before proceeding to the main results in the following sections, we discuss the framework that we use to derive these results. Our approach to constructing reduced formulations has its basis in the Dantzig-Wolfe decomposition principle. We start with a brief review of Dantzig-Wolfe decomposition, applying the setup introduced in Chapter 6 of Bertsimas and Tsitsiklis (1997) to a general multi-stage linear programming problem. We then introduce two lemmas that relate multi-stage linear programs to the Dantzig-Wolfe reformulations of some more compact linear programs. These lemmas are central to the reductions of ALPs in Sections 4 and 5.

Consider a general multi-stage linear programming formulation

$$\begin{aligned} (\mathbf{LP}) \ \max \ & \sum_t \mathbf{c}_t^T \mathbf{w}_t \\ \text{s.t.} \ & \sum_t \mathbf{D}_t \mathbf{w}_t = \mathbf{d}, \\ & \mathbf{w}_t \in \mathcal{P}, \qquad \forall t, \end{aligned}$$

where $\mathbf{c}_t$ is a K-vector, $\mathbf{d}$ is an L-vector, $\mathbf{D}_t$ is an $(L \times K)$ matrix, and $\mathcal{P} = \{\mathbf{w} \in \mathbb{R}_+^K : \mathbf{B}\mathbf{w} \le \mathbf{b}\}$ is a bounded polyhedron. To relate our framework to the ALPs we consider later in the paper, we assume that the time index t runs from 1 to T and restrict $\mathcal{P}$ to be independent of t. Our analysis also applies when $\mathcal{P}$ depends on t.

Let $\mathcal{E}$ be the index set of the extreme points of $\mathcal{P}$; that is, the set of extreme points of $\mathcal{P}$ can be written as $\{\mathbf{w}^e \in \mathcal{P} : e \in \mathcal{E}\}$. The Dantzig-Wolfe decomposition principle (Dantzig and Wolfe 1960, 1961) states that $(\mathbf{LP})$ is equivalent to

$$\begin{aligned} (\mathbf{MP}) \ \max \ & \sum_{t, e \in \mathcal{E}} \mu_{t,e} \mathbf{c}_t^T \mathbf{w}^e \\ \text{s.t.} \ & \sum_{t, e \in \mathcal{E}} \mu_{t,e} \mathbf{D}_t \mathbf{w}^e = \mathbf{d}, && (10) \\ & \sum_{e \in \mathcal{E}} \mu_{t,e} = 1, \qquad \forall t, && (11) \\ & \mu_{t,e} \ge 0, \qquad \forall t, e \in \mathcal{E}. \end{aligned}$$

The formulation $(\mathbf{MP})$ is often called the *master problem* and contains a column for each extreme point of $\mathcal{P}$ in each period. Direct solution methods for $(\mathbf{MP})$ are usually intractable. Instead, $(\mathbf{MP})$

is often solved using a column generation procedure. This procedure maintains a *restricted master problem* (**RMP**), which works with a small subset $\bigcup_t \overline{\mathcal{E}}_t$ of the columns in the master problem with $\overline{\mathcal{E}}_t \subseteq \mathcal{E}$ for each t. Let (π, ϕ) be the dual variables corresponding to (10) and (11) in (**RMP**). A column generation subproblem is used to determine the columns with the highest reduced cost. For a given t, the column generation subproblem equals

$$(\mathbf{CG})_t \quad \varphi_t = \max \ (\mathbf{c}_t^T - \pi^T \mathbf{D}_t)\mathbf{w}_t - \phi_t \\ \text{s.t.} \quad \mathbf{w}_t \in \mathcal{P}.$$

If the reduced cost $\varphi_t \leq 0$ for all t, the solution to (**RMP**) is optimal to (**MP**) as well. Otherwise, we can add columns with positive reduced costs to (**RMP**) based on the solution to the column generation subproblems, and continue by re-optimizing (**RMP**).

The Dantzig-Wolfe decomposition discussed above originates from a compact formulation (**LP**), which is reformulated to a master problem (**MP**) with a huge number of columns. Usually the formulation aims at taking advantage of some special structure in the polyhedron $\mathcal{P}$ that allows the column generation subproblems $(\mathbf{CG})_t$ to be solved efficiently. In the remainder of this section we take a different approach, in that our starting point is a large scale multi-stage linear program. In subsequent sections, these large scale multi-stage linear programs correspond to ALPs that result from the LP-based ADP approach. The ALPs have a huge number of columns but a moderate number of constraints (corresponding to the basis functions), with a multi-stage structure that is similar to (**MP**). The question we try to resolve is whether it is possible to construct equivalent, but more compact, formulations for the ALPs by applying the Dantzig-Wolfe decomposition principle. That is, we are interested in relating the ALPs to Dantzig-Wolfe reformulations of some more compact linear programs. We use the structure of the column generation subproblems to evaluate when and under what conditions we can construct compact formulations whose Dantzig-Wolfe reformulations are equivalent to the ALPs.

To explain our approach, consider a generic multi-stage linear program:

$$\begin{aligned} (\mathbf{MP}') \ \max \ & \sum_{t,e \in \mathcal{E}'} \mu_{t,e} c'_{t,e} \\ \text{s.t.} \ & \sum_{t,e \in \mathcal{E}'} \mu_{t,e} \mathbf{d}'_{t,e} = \mathbf{d}, && && (12) \\ & \sum_{e \in \mathcal{E}'} \mu_{t,e} = 1, && \forall t, && (13) \\ & \mu_{t,e} \geq 0, && \forall t, e \in \mathcal{E}', \end{aligned}$$

where $c'_{t,e}$ is a scalar, $\mathbf{d}'_{t,e}$ is an L-vector for all t, e, and $\mathcal{E}'$ is an index set of columns for each t. An implicit assumption is that $\mathcal{E}'$ is huge. We also emphasize that, in contrast to (**MP**), the parameters $c'_{t,e}$ and $\mathbf{d}'_{t,e}$ are not expressed relative to the extreme points of an underlying polyhedron.

Nevertheless, a column generation procedure can be used to solve ($\mathbf{MP'}$) by successively solving a restricted master problem ($\mathbf{RMP'}$), which contains selected columns in ($\mathbf{MP'}$). Let (π, ϕ) be the dual variables corresponding to (12) and (13) in ($\mathbf{RMP'}$). The column generation subproblem for each t is given by

$$(\mathbf{CG'})_t \ \max_{e \in \mathcal{E}'} c'_{t,e} - \pi^T \mathbf{d}'_{t,e} - \phi_t.$$

While $(\mathbf{CG'})_t$ is posed as an *oracle* subproblem that requires enumeration of all $e \in \mathcal{E}'$, there is often an underlying structure that enables a more efficient solution procedure. Specifically, suppose that we can formulate $(\mathbf{CG'})_t$ as an equivalent linear program

$$\begin{aligned} (\mathbf{RCG})_t \ \max \ & (\hat{\mathbf{c}}_t^T - \pi^T \hat{\mathbf{D}}_t)\mathbf{w}_t - \phi_t \\ \text{s.t.} \quad & \mathbf{w}_t \in \hat{\mathcal{P}}, \end{aligned}$$

where $\hat{\mathcal{P}} = \{\mathbf{w} \in \mathbb{R}_+^k : \hat{\mathbf{B}}\mathbf{w} \le \hat{\mathbf{b}}\}$ is a bounded polyhedron, $\hat{\mathbf{c}}_t$ is a K-vector, and $\hat{\mathbf{D}}_t$ is an $(L \times K)$ matrix. Our central thesis is that we can construct a compact representation of ($\mathbf{MP'}$) whenever $(\mathbf{CG'})_t$ and $(\mathbf{RCG})_t$ are equivalent.

LEMMA 1. *Let π and ϕ be* any *L-vector and T-vector, respectively. If for each t,*

(i) For any $e \in \mathcal{E}'$, there exists an $\mathbf{w}_t \in \hat{\mathcal{P}}$ such that $c'_{t,e} = \hat{\mathbf{c}}_t^T \mathbf{w}_t$ and $\mathbf{d}'_{t,e} = \hat{\mathbf{D}}_t \mathbf{w}_t$, and

(ii) For each extreme point $\mathbf{w}_t$ of $\hat{\mathcal{P}}$, there exists an $e \in \mathcal{E}'$ such that $c'_{t,e} = \hat{\mathbf{c}}_t^T \mathbf{w}_t$ and $\mathbf{d}'_{t,e} = \hat{\mathbf{D}}_t \mathbf{w}_t$,

then the linear program ($\boldsymbol{MP'}$) is equivalent to

$$\begin{aligned} (\boldsymbol{LP'}) \ \max \ & \sum_t \hat{\mathbf{c}}_t^T \mathbf{w}_t \\ \text{s.t.} \ & \sum_t \hat{\mathbf{D}}_t \mathbf{w}_t = \mathbf{d}, \\ & \mathbf{w}_t \in \hat{\mathcal{P}}, \qquad \forall t. \end{aligned}$$

Proof. To establish this result, we first verify that ($\mathbf{MP'}$) is equivalent to the formulation

$$\begin{aligned} (\mathbf{MP''}) \ \max \ & \sum_{t, e \in \hat{\mathcal{E}}} \mu_{t,e} \hat{\mathbf{c}}_t^T \mathbf{w}^e \\ \text{s.t.} \ & \sum_{t, e \in \hat{\mathcal{E}}} \mu_{t,e} \hat{\mathbf{D}}_t \mathbf{w}^e = \mathbf{d}, \\ & \sum_{e \in \mathcal{E}} \mu_{t,e} = 1, \qquad \forall t, \\ & \mu_{t,e} \ge 0, \qquad \forall t, e \in \hat{\mathcal{E}}, \end{aligned}$$

where the set $\{\mathbf{w}^e : e \in \hat{\mathcal{E}}\}$ corresponds to the extreme points of $\hat{\mathcal{P}}$. Condition (i) guarantees that any solution to ($\mathbf{MP'}$) has a corresponding solution (with the same objective value) to ($\mathbf{MP''}$), while condition (ii) guarantees that any solution to $(\mathbf{MP}'')$ gives a solution to $(\mathbf{MP}')$. Then, the Dantzig-Wolfe decomposition principle states that $(\mathbf{MP}'')$ is equivalent to $(\mathbf{LP}')$. ■

Lemma 1 provides sufficient conditions to constructing a reduced formulation. In certain settings, however, these conditions can be overly restrictive. In particular, it can be difficult to establish condition (ii) for *any* dual vector π and *all* extreme points. Therefore, we also consider a weaker notion of equivalence between linear programs. We say two linear programs are *weakly equivalent* if their optimal solutions intersect; that is, there exists an optimal solution in one that is feasible in the other, and vice versa. Note that weakly equivalent linear programs yield the same optimal solution value. This definition of weak equivalence is related to a similar notion considered in Tardella (1990). Our main result is established in the following lemma.

LEMMA 2. *The linear program* $(\boldsymbol{MP'})$ *is* weakly equivalent *to the linear program* $(\boldsymbol{LP'})$ *if the following conditions hold for any t:*

(i) For any $e \in \mathcal{E}'$, there exists an $\mathbf{w}_t \in \hat{\mathcal{P}}$ such that $c'_{t,e} = \hat{\mathbf{c}}_t^T \mathbf{w}_t$ and $\mathbf{d}'_{t,e} = \hat{\mathbf{D}}_t \mathbf{w}_t$,

(ii) There exists an optimal solution (π, ϕ) to the dual of $(\boldsymbol{LP'})$ such that a set of linear inequalities $\pi^T \mathbf{G} \leq \mathbf{g}$ holds, and

(iii) For any π such that $\pi^T \mathbf{G} \leq \mathbf{g}$, there exists an optimal solution $\mathbf{w}_t$ to $(\boldsymbol{RCG})_t$ such that $c'_{t,e} = \hat{\mathbf{c}}_t^T \mathbf{w}_t$ and $\mathbf{d}'_{t,e} = \hat{\mathbf{D}}_t \mathbf{w}_t$ for some $e \in \mathcal{E}'$.

Proof. We first show that $(\mathbf{MP}')$ is weakly equivalent to the formulation $(\mathbf{MP}'')$. Condition (i) guarantees that there exists an optimal solution to $(\mathbf{MP}')$ that has a corresponding solution (with the same objective value) to $(\mathbf{MP}'')$ (in fact, this will hold for any solution).

To establish that there exists an optimal solution to $(\mathbf{MP}'')$ that has a corresponding feasible solution to $(\mathbf{MP}')$, we use conditions (ii) and (iii). In particular, assume that we solve $(\mathbf{MP}'')$ using the column generation procedure outlined before. The column generation subproblem equals $(\mathbf{RCG})_t$, where we can assume (or impose) $\pi^T \mathbf{G} \leq \mathbf{g}$ due to condition (ii). Condition (iii) then guarantees that we can only add columns during the procedure that will have a corresponding column in $(\mathbf{MP}')$. Thus, the procedure will terminate with an optimal solution that has a corresponding solution to $(\mathbf{MP}')$.

Finally, the Dantzig-Wolfe decomposition principle states that $(\mathbf{MP}'')$ is equivalent to $(\mathbf{LP}')$. ■

The main difference between Lemmas 1 and 2 is that a set of inequalities $\pi^T \mathbf{G} \leq \mathbf{g}$ is required in Lemma 2 for an optimal solution (π, ϕ) to the dual of $(\mathbf{LP}')$. Such restrictions are commonly called *dual optimal inequalities* (Ben Amor et al. 2006) in the literature, and have been used to improve convergence of column generation procedures. We emphasize that the dual optimal inequalities are imposed on the dual of $(\mathbf{LP}')$, which is the *reduced program*. This is often quite convenient, as establishing the dual optimal inequalities can be much easier for the reduced program ($\mathbf{LP}'$) than for the full program ($\mathbf{MP}'$).

In the next two sections, we apply this framework to the ALP dual programs for NRM that result from applying the affine and separable piecewise linear approximations. In Section 4, we apply Lemma 1 to establish the reduced programs for affine approximations to NRM with independent or discrete choice models of demand. Section 5 applies Lemma 2 to establish the reduced programs for separable piecewise linear approximations. Since Lemma 2 requires dual optimal inequalities, we show how such inequalities can be established in the context of separable piecewise linear approximations for NRM.

4. Affine Approximations

In this section, we consider ALPs that result from the affine approximation (8), and show that Lemma 1 can be used to construct reduced formulations for both the independent and discrete choice models of demand.

4.1. Independent Demand Models

Substituting (8) into $(\mathbf{D})^{\phi}$ and using (1) yields the dual program

$$
\begin{aligned}
(\mathbf{D})^{A,I} \quad \max_{\mathbf{p}} \quad & \sum_{t,(\mathbf{x},\mathbf{u})\in\mathcal{S}} \left(\sum_j \lambda_{t,j} f_j u_j\right) p_{t,\mathbf{x},\mathbf{u}} \\
\text{s.t.} \quad & \sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} x_i p_{t,\mathbf{x},\mathbf{u}} = \begin{cases} c_i, & \text{if } t=1, \\ \displaystyle\sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} x_i p_{t-1,\mathbf{x},\mathbf{u}} - \sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S},j} \lambda_{t-1,j} a_{ij} u_j p_{t-1,\mathbf{x},\mathbf{u}}, & \text{if } t>1, \end{cases} \quad \forall t,i, \quad (14) \\
& \sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} p_{t,\mathbf{x},\mathbf{u}} = 1, \qquad \forall t, \quad (15) \\
& \mathbf{p} \geq \mathbf{0}.
\end{aligned}
$$

Adelman (2007) proposes a column generation procedure to solve $(\mathbf{D})^{A,I}$. Suppose we have an optimal dual solution (V,θ) to a restricted master problem that only contains a subset of the columns in $(\mathbf{D})^{A,I}$. For each t, the column generation subproblem is given by

$$
(\mathbf{CG})_t^{A,I} \quad \max_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} \sum_j \lambda_{t,j} f_j u_j - \sum_i V_{t,i} x_i - \sum_i V_{t+1,i}\left(\sum_j \lambda_{t,j} a_{ij} u_j - x_i\right) - \theta_t.
$$

Adelman (2007) solves the column generation subproblem $(\mathbf{CG})_t^{A,I}$ as an integer program. Our objective however is to construct a reduced formulation that is equivalent to $(\mathbf{D})^{A,I}$ by applying Lemma 1, which requires that we formulate the subproblems as linear programs. To that end, we consider the linear programming relaxation of $(\mathbf{CG})_t^{A,I}$ and show that the integrality restrictions are redundant. We obtain the linear programming relaxation of $(\mathbf{CG})_t^{A,I}$ using decision variables

$\mathbf{r} \in \mathbb{R}_+^I$ and $\mathbf{q} \in \mathbb{R}_+^J$, with $\mathbf{r}$ a continuous relaxation of $\mathbf{x}$ and $\mathbf{q}$ a continuous relaxation of $\mathbf{u}$. The resulting formulation is given by

$$
\begin{aligned}
(\mathbf{RCG})_t^{A,I} \quad \max_{\mathbf{r},\mathbf{q}} \quad & \sum_j \lambda_{t,j} f_j q_j - \sum_i V_{t,i} r_i - \sum_i V_{t+1,i} \left(\sum_j \lambda_{t,j} a_{ij} q_j - r_i \right) - \theta_t \\
\text{s.t.} \quad & 0 \le q_j \le 1, && \forall j, && (16) \\
& 0 \le r_i \le c_i, && \forall i, && (17) \\
& q_j \le r_i, && \forall i,j : a_{ij} = 1. && (18)
\end{aligned}
$$

Here, constraints (16) and (17) are simple bounds and constraint (18) follows from the restriction that products can only be offered if sufficient resources are available.

Lemma 3. *The polyhedron* $\mathcal{Q}^{A,I} = \{(\mathbf{r},\mathbf{q}) \in \mathbb{R}_+^{I+J} : (16),(17),(18)\}$ *is integral.*

Proof. Note that the constraints in $\mathcal{Q}^{A,I}$ exhibit "dual network structure," which means that each contributes a row with coefficients of 0, except for at most one $+1$, and at most one -1. As a result, the constraint matrix is totally unimodular (Wolsey 1998) and $\mathcal{Q}^{A,I}$ has integer extreme points. ■

In the following, we check that conditions (i) and (ii) in Lemma 1 are satisfied. According to Lemma 1, $(\mathbf{D})^{A,I}$ is then equivalent to the reduced formulation

$$
\begin{aligned}
(\mathbf{D\text{-}R})^{A,I} \quad \max_{\{\mathbf{r}_t,\mathbf{q}_t\}_{\forall t}} \quad & \sum_{t,j} \lambda_{t,j} f_j q_{t,j} \\
\text{s.t.} \quad & r_{t,i} = \begin{cases} c_i, & \text{if } t = 1, \\ r_{t-1,i} - \sum_j \lambda_{t-1,j} a_{ij} u_j, & \text{if } t > 1, \end{cases} && \forall t,i, && (19) \\
& q_{t,j} \le r_{t,i}, && \forall t,i,j : a_{ij} = 1, && (20) \\
& 0 \le q_{t,j} \le 1, && \forall t,j. && (21)
\end{aligned}
$$

Note that constraint (17) is redundant in the reduced formulation.

To illustrate the connection with Lemma 1, we note that every element e in the index set of columns in $\mathcal{E}'$ in (**MP**') corresponds to a state-action pair $(\mathbf{x},\mathbf{u}) \in \mathcal{S}$ in $(\mathbf{D})^{A,I}$. Thus, $c'_{t,e} = \sum_j \lambda_{t,j} f_j u_j$ when e corresponds to $(\mathbf{x},\mathbf{u}) \in \mathcal{S}$. In $(\mathbf{MP}')$, $d'_{t,e}$ is a vector of dimension $T \times I$. When e corresponds to $(\mathbf{x},\mathbf{u}) \in \mathcal{S}$, the (t,i)-th element of this vector equals x_i, and $(t+1,i)$-th element equals $\sum_j \lambda_{t,j} a_{ij} u_j - x_i$. All other elements equal 0. Problem $(\mathbf{RCG})_t$ corresponds to $(\mathbf{RCG})_t^{A,I}$, with $\hat{\mathbf{c}}_t$ a vector of dimension $I+J$ and $\hat{\mathbf{B}}_t$ a matrix with TI rows and $I+J$ columns. The i-th element in $\hat{\mathbf{c}}_t$ equals 0 for all i, while the $(I+j)$-th element in $\hat{c}_t$ equals $\lambda_{t,j} f_j$ for all j. In the matrix $\hat{\mathbf{B}}_t$, the element in row (t,i) and column i equals 1 and the element in row $(t+1,i)$ and column i equals -1. An element in row $(t+1,i)$ and column $I+j$ equals $\lambda_{t,j} a_{ij}$. All other elements in $\hat{\mathbf{B}}_t$ equal 0.

For fixed t, Condition (i) in Lemma 1 therefore requires that for every $(\mathbf{x},\mathbf{u})\in\mathcal{S}$, there exists an $(\mathbf{r},\mathbf{q})\in\mathcal{Q}^{A,I}$ such that

$$\sum_j \lambda_{t,j} f_j u_j = \sum_j \lambda_{t,j} f_j q_j, \tag{22}$$

$$x_i = r_i, \qquad \forall i, \tag{23}$$

$$\sum_j \lambda_{t,j} a_{ij} u_j - x_i = \sum_j \lambda_{t,j} a_{ij} q_j - r_i, \qquad \forall i. \tag{24}$$

Note that we obtain (22)–(24) by equating the terms corresponding to each individual dual variable in $(\mathbf{CG})_t^{A,I}$ and $(\mathbf{RCG})_t^{A,I}$, as well as equating the terms that do not involve a dual variable. In particular, (23) follows by equating terms for $V_{t,i}$, while (24) follows by equating terms for $V_{t+1,i}$. This yields a convenient way to ensure that columns in the Dantzig-Wolfe reformulation of $(\mathbf{D\text{-}R})^{A,I}$ has a corresponding individual column in $(\mathbf{D})^{A,I}$, and vice versa. Equations (22)–(24) are verified by letting $r_i = x_i$ for all i and $q_j = u_j$ for all j and observing that $(\mathbf{r},\mathbf{q})\in\mathcal{Q}^{A,I}$.

Condition (ii) in Lemma 1 requires that for every extreme point $(\mathbf{r},\mathbf{q})$ of $\mathcal{Q}^{A,I}$, there exists a state-action pair $(\mathbf{x},\mathbf{u})\in\mathcal{S}$ such that (22)–(24) are satisfied. Because every extreme point of $\mathcal{Q}^{A,I}$ is integer according to Lemma 3, we can confirm this by defining $x_i = r_i$ for all i and $u_j = q_j$ for all j. Note that the constraints in $\mathcal{Q}^{A,I}$ ensure that $(\mathbf{x},\mathbf{u})\in\mathcal{S}$.

As a result, we verify that the conditions in Lemma 1 are satisfied, and we summarize the main result of this section in the following proposition.

Proposition 1 **(Tong and Topaloglu (2011))**. *The linear programs $(\boldsymbol{D})^{A,I}$ and $(\boldsymbol{D\text{-}R})^{A,I}$ are equivalent.*

Proposition 1 relies on the connection between $(\mathbf{D})^{A,I}$ and the Dantzig-Wolfe reformulation of $(\mathbf{D\text{-}R})^{A,I}$, which is made explicit in Lemma 1. The column generation subproblem to solve $(\mathbf{D})^{A,I}$ is $(\mathbf{CG})_t^{A,I}$, while the column generation subproblem to solve the Dantzig-Wolfe reformulation of $(\mathbf{D\text{-}R})^{A,I}$ is $(\mathbf{RCG})_t^{A,I}$. Equations (22)–(24) ensure that the two column generation subproblems are equivalent when $(\mathbf{CG})_t^{A,I}$ has integer extreme points, which Lemma 3 confirms.

Proposition 1 was first shown in Tong and Topaloglu (2011), though their proof is markedly different from ours. However, our main goal here is to highlight a general framework for constructing reduced formulations that can also be applied in other settings.

4.2. Customer Choice Models

Substituting (8) into $(\mathbf{D})^{\phi}$ yields the dual program

$$(\mathbf{D})^{A,C} \ \max_{\mathbf{p}} \sum_{t,(\mathbf{x},\mathbf{u})\in\mathcal{S}} R_t(\mathbf{u}) p_{t,\mathbf{x},\mathbf{u}}$$

$$\text{s.t.} \quad \sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} x_i p_{t,\mathbf{x},\mathbf{u}} = \begin{cases} c_i, & \text{if } t=1, \\ \sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} x_i p_{t-1,\mathbf{x},\mathbf{u}} - \sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} Q_{t-1,i}(\mathbf{u}) p_{t-1,\mathbf{x},\mathbf{u}}, & \text{if } t>1, \end{cases} \quad \forall t,i, \tag{25}$$

$$\sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} p_{t,\mathbf{x},\mathbf{u}} = 1, \quad \forall t, \tag{26}$$

$$\mathbf{p} \geq \mathbf{0},$$

where $R_t(\mathbf{u}) = \rho_t \sum_j f_j P_{t,j}(\mathbf{u})$ and $Q_{t,i}(\mathbf{u}) = \rho_t \sum_j a_{ij} P_{t,j}(\mathbf{u})$ are the revenue and consumption rate of leg i during period t for the offer set given by $\mathbf{u}$, respectively.

Zhang and Adelman (2009) discuss a column generation procedure to solve $(\mathbf{D})^{A,C}$, and use an integer program to solve the column generation subproblems in the special case of a multinomial logit (MNL) choice model with disjoint consideration sets (Liu and van Ryzin 2008). Here, we focus on reduced formulations that can be obtained with a general discrete choice model. In this case, the column generation subproblem for a given t equals

$$(\mathbf{CG})_t^{A,C} \quad \max_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} R_t(\mathbf{u}) - \sum_i V_{t,i} x_i - \sum_i V_{t+1,i} \left(Q_{t-1,i}(\mathbf{u}) - x_i\right) - \theta_t,$$

where (V,θ) again refers to an optimal dual solution to the restricted master problem.

To construct a reduced formulation, we begin by formulating $(\mathbf{CG})_t^{A,C}$ as a linear program:

$$(\mathbf{RCG})_t^{A,C} \quad \max_{\mathbf{r},\mathbf{h}} \sum_{\mathbf{u}\in\mathcal{U}} R_t(\mathbf{u}) h_{\mathbf{u}} - \sum_i V_{t,i} r_i - \sum_i V_{t+1,i} \left(\sum_{\mathbf{u}\in\mathcal{U}} Q_{t-1,i}(\mathbf{u}) h_{\mathbf{u}} - r_i \right) - \theta_t$$

$$\text{s.t.} \quad \sum_{\mathbf{u}\in\mathcal{U}} h_{\mathbf{u}} = 1, \tag{27}$$

$$0 \leq r_i \leq c_i, \quad \forall i, \tag{28}$$

$$\sum_{\mathbf{u}\in\mathcal{U}: i\in\mathcal{I}(\mathbf{u})} h_{\mathbf{u}} \leq r_i, \quad \forall i, \tag{29}$$

$$h_{\mathbf{u}} \geq 0, \quad \forall \mathbf{u}\in\mathcal{U}.$$

Here, $\mathcal{I}(\mathbf{u}) = \bigcup_{j:u_j=1} \mathcal{I}_j$ corresponds to the set of legs used in an offer set given by $\mathbf{u}\in\mathcal{U}$. Constraint (27) states that exactly one set of products is offered, and constraint (28) provides a simple bound on the resource levels. To motivate constraint (29), we first note that $h_{\mathbf{u}} \leq r_i$ for all $\mathbf{u}, i \in \mathcal{I}(\mathbf{u})$ because products can only be offered when sufficient resources are available. Since constraint (28) states that exactly one set of products is offered, this can be strengthened to obtain (29). This strengthening is critical, as it guarantees that the constraints in $(\mathbf{RCG})_t^{A,C}$ yield integer extreme points.

Lemma 4. *The polyhedron* $\mathcal{Q}^{A,C} = \{(\mathbf{r},\mathbf{h}) \in \mathbb{R}_+^{I+2^J} : (27),(28),(29)\}$ *is integral.*

Proof. The proof follows by contradiction. Suppose $(\tilde{\mathbf{r}}, \tilde{\mathbf{h}})$ is a fractional extreme point of $\mathcal{Q}^{A,C}$. As a first step we observe that either $\tilde{r}_i = c_i$ or $\tilde{r}_i = \sum_{\mathbf{u}\in\mathcal{U}: i\in\mathcal{I}(\mathbf{u})} \tilde{h}_{\mathbf{u}}$, since $(\tilde{\mathbf{r}}, \tilde{\mathbf{h}})$ is an extreme point. Let $\tilde{\mathcal{I}} = \{i : \tilde{r}_i = c_i\}$. In addition, let $\tilde{\mathcal{U}} = \{\mathbf{u} \in \mathcal{U} : 0 < \tilde{h}_{\mathbf{u}} < 1\}$ and $L = |\tilde{\mathcal{U}}|$. Then we must have $L > 1$, for otherwise $(\tilde{\mathbf{r}}, \tilde{\mathbf{h}})$ is integral by our previous observation. Without loss of generality, let $\tilde{\mathcal{U}} = \{\mathbf{u}^l : l = 1, \ldots, L\}$ and let $\omega^l = \tilde{h}_{\mathbf{u}^l}$ for $l = 1, \ldots, L$. Next, we define solutions $(\hat{\mathbf{r}}^l, \hat{\mathbf{h}}^l)$ for $l = 1, \ldots, L$ by letting

$$\hat{h}^l_{\mathbf{u}} = \begin{cases} 1, \text{ if } \mathbf{u} = \mathbf{u}^l, \\ 0, \text{ otherwise}, \end{cases} \qquad \forall \mathbf{u} \in \mathcal{U},$$

$$\hat{r}^l_i = \begin{cases} c_i, \text{ if } i \in \tilde{\mathcal{I}}, \\ 1, \text{ if } i \in \mathcal{I}(\mathbf{u}^l) \setminus \tilde{\mathcal{I}}, \\ 0, \text{ otherwise}, \end{cases} \qquad \forall i.$$

By construction, each solution $(\hat{\mathbf{r}}^l, \hat{\mathbf{h}}^l)$ is integral and satisfies the constraints in $\mathcal{Q}^{A,C}$. We can verify that $(\tilde{\mathbf{r}}, \tilde{\mathbf{h}}) = \sum_{l=1}^{L} \omega^l \left(\hat{\mathbf{r}}^l, \hat{\mathbf{h}}^l\right)$, which implies that $(\tilde{\mathbf{r}}, \tilde{\mathbf{h}})$ cannot be an extreme point of $\mathcal{Q}^{A,C}$. ■

In the following, we check that conditions (i) and (ii) in Lemma 1 are satisfied; this shows that $(\mathbf{D})^{A,C}$ is equivalent to the reduced formulation

$$(\mathbf{D\text{-}R})^{A,C} \quad \max_{\{\mathbf{r}_t, \mathbf{h}_t\}\forall t} \sum_{t, \mathbf{u} \in \mathcal{U}} R_t(\mathbf{u}) h_{t,\mathbf{u}}$$

$$\text{s.t.} \quad r_{t,i} = \begin{cases} c_i, & \text{if } t = 1, \\ r_{t-1,i} - \sum_{\mathbf{u} \in \mathcal{U}} Q_{t-1,i}(\mathbf{u}) h_{t-1,\mathbf{u}}, & \text{if } t > 1, \end{cases} \qquad \forall t, i, \tag{30}$$

$$\sum_{\mathbf{u} \in \mathcal{U}} h_{t,\mathbf{u}} = 1, \qquad \forall t, \tag{31}$$

$$\sum_{\mathbf{u} \in \mathcal{U} : i \in \mathcal{I}(\mathbf{u})} h_{t,\mathbf{u}} \le r_{t,i}, \qquad \forall t, i, \tag{32}$$

$$h_{t,\mathbf{u}} \ge 0, \qquad \forall t, \mathbf{u} \in \mathcal{U}.$$

For fixed t, condition (i) states that for every $(\mathbf{x}, \mathbf{u}) \in \mathcal{X}$, there exists an $(\mathbf{r}, \mathbf{h}) \in \mathcal{Q}^{A,C}$ such that

$$R_t(\mathbf{u}) = \sum_{\mathbf{u}' \in \mathcal{U}} R_t(\mathbf{u}') h_{\mathbf{u}'}, \tag{33}$$

$$x_i = r_i, \qquad \forall i, \tag{34}$$

$$Q_{t-1,i}(\mathbf{u}) - x_i = \sum_{\mathbf{u}' \in \mathcal{U}} Q_{t-1,i}(\mathbf{u}') h_{\mathbf{u}'} - r_i, \qquad \forall i. \tag{35}$$

Again, we obtain (33)–(35) by equating the terms corresponding to each individual dual variable in $(\mathbf{CG})_t^{A,C}$ and $(\mathbf{RCG})_t^{A,C}$, as well as equating the terms that do not involve a dual variable. For every $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}$, we can verify (33)–(35) by letting $r_i = x_i$ for all i and $h_{\mathbf{u}'} = \mathbb{1}\{\mathbf{u}' = \mathbf{u}\}$ for all $\mathbf{u}' \in \mathcal{U}$, and observing that $(\mathbf{r}, \mathbf{h}) \in \mathcal{Q}^{A,C}$. Condition (ii) in Lemma 1 requires that for every extreme point $(\mathbf{r}, \mathbf{h})$ of $\mathcal{Q}^{A,C}$, there exists a state-action pair $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}$ such that (33)–(35) are satisfied. Because every extreme point of $\mathcal{Q}^{A,C}$ is integer, we can confirm this by defining $x_i = r_i$ for all i and choosing $\mathbf{u}$ such that $h_{\mathbf{u}} = 1$.

Consequently, the conditions in Lemma 1 are satisfied, and its application leads to the following proposition.

Proposition 2. *The linear programs $(\boldsymbol{D})^{A,C}$ and $(\boldsymbol{D}\text{-}\boldsymbol{R})^{A,C}$ are equivalent.*

We note that constraint (28) is redundant in $(\mathbf{D}\text{-}\mathbf{R})^{A,C}$ and therefore is removed. The reduced program $(\mathbf{D}\text{-}\mathbf{R})^{A,C}$ is still exponential over the product space, though it is no longer exponential over the state space. Hence, the reduction result in Proposition 2 breaks the "curse of dimensionality" in the state space, but not in the action space. This should be contrasted to the reduced program $(\mathbf{D}\text{-}\mathbf{R})^{A,I}$ for the independent demand case, where the size of the reduced program is polynomial in the number of legs and the number of products. On an intuitive level, this is due to the fact that products cannot be tracked independently when customers choose among them, while they can be tracked independently when each arriving customer requests a specific product.

5. Separable Piecewise Linear Approximations

We now turn to the ALPs that result from applying the separable piecewise linear approximation (9) for both the independent and discrete choice models of demand. In contrast to the affine approximations considered in Section 4, constructing equivalent reduced formulations turns out to be more involved and requires the use of Lemma 2, which imposes a set of dual optimal inequalities.

5.1. Independent Demand Models

Substituting (1) and (9) into $(\mathbf{D})^{\phi}$ yields the dual program

$$
\begin{aligned}
(\mathbf{D})^{S,I}\ \max_{\mathbf{p}} \quad & \sum_{t,(\mathbf{x},\mathbf{u})\in\mathcal{S}} \left(\sum_j \lambda_{t,j} f_j u_j\right) p_{t,\mathbf{x},\mathbf{u}} \\
\text{s.t.} \quad & \sum_{\substack{(\mathbf{x},\mathbf{u})\in\mathcal{S}:\\ x_i\ge k}} p_{t,\mathbf{x},\mathbf{u}} = \begin{cases} 1, & \text{if } t=1,\\ \displaystyle\sum_{\substack{(\mathbf{x},\mathbf{u})\in\mathcal{S}:\\ x_i\ge k}} p_{t-1,\mathbf{x},\mathbf{u}} - \sum_{\substack{(\mathbf{x},\mathbf{u})\in\mathcal{S},j:\\ x_i=k}} \lambda_{t-1,j} a_{ij} u_j p_{t-1,\mathbf{x},\mathbf{u}}, & \text{if } t>1, \end{cases} \\
& \qquad \forall t,i,k, \qquad (36) \\
& \sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} p_{t,\mathbf{x},\mathbf{u}} = 1, \qquad \forall t, \qquad (37) \\
& \mathbf{p} \ge \mathbf{0}.
\end{aligned}
$$

For notational convenience, we omit the index set $\{1,\ldots,c_i\}$ for k when i is given. Suppose (V,θ) is an optimal dual solution to a restricted master problem, and fix t. Then, the column generation subproblem equals

$$
(\mathbf{CG})_t^{S,I}
$$

$$
\max_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} \sum_j \lambda_{t,j} f_j u_j - \sum_{i,k} V_{t,i,k} \mathbb{1}\{x_i\ge k\} - \sum_{i,k} V_{t+1,i,k}\left(\sum_j \lambda_{t,j} a_{ij} u_j \mathbb{1}\{x_i=k\} - \mathbb{1}\{x_i\ge k\}\right) - \theta_t
$$

$$= \max_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} \sum_j \lambda_{t,j}\left[f_j - \sum_{i,k} a_{ij}\left(V_{t+1,i,k} - V_{t+1,i,k-1}\right)\mathbb{1}\{x_i \geq k\}\right]u_j - \sum_i (V_{t,i,k} - V_{t+1,i,k})\mathbb{1}\{x_i \geq k\} - \theta_t.$$

Here, we assume boundary conditions $V_{T+1,i,k} = 0$ for all t,i,k and $V_{t,i,0} = 0$ for all t,i.

$(\mathbf{CG})_t^{S,I}$ is more complicated than its counterpart $(\mathbf{CG})_t^{A,I}$ for the affine approximation, due to the non-linear terms $\mathbb{1}\{x_i \geq k\}u_j$. However, we can linearize the column generation subproblem by introducing additional decision variables $\mathbf{y}$ and $\mathbf{z}$, such that $y_{i,k} = 1$ iff $\mathbb{1}\{x_i \geq k\}$ for all i,k, and $z_{j,i,k} = 1$ iff $\mathbb{1}\{x_i \geq k\}$ and $u_j = 1$ for all j,i,k. Given these decision variables, we can formulate $(\mathbf{CG})_t^{S,I}$ as an integer programming problem:

$$
\begin{aligned}
(\mathbf{CG}')_t^{S,I} \quad \max_{\mathbf{q},\mathbf{y},\mathbf{z}} \quad & \sum_j \lambda_{t,j} f_j q_j - \sum_{i,k}(V_{t,i,k} - V_{t+1,i,k})y_{i,k} - \sum_j \sum_{i,k} \lambda_{t,j} a_{ij}(V_{t+1,i,k} - V_{t+1,i,k-1})z_{j,i,k} - \theta_t \\
\text{s.t.} \quad & y_{i,k+1} \leq y_{i,k}, && \forall i,k, \quad (38)\\
& q_j = z_{j,i,1}, && \forall j,i : a_{ij} = 1, \quad (39)\\
& z_{j,i,k+1} \leq z_{j,i,k}, && \forall j,i,k : a_{ij} = 1, \quad (40)\\
& z_{j,i,k} \leq y_{i,k}, && \forall j,i,k : a_{ij} = 1, \quad (41)\\
& q_j + y_{i,k} \leq 1 + z_{j,i,k}, && \forall j,i,k : a_{ij} = 1, \quad (42)\\
& \mathbf{q},\mathbf{y},\mathbf{z} \text{ binary.} && (43)
\end{aligned}
$$

For notational convenience, we adopt the convention that $y_{i,c_i+1} = 0$ for all i, and $z_{j,i,c_i+1} = 0$ for all j,i. Constraint (38) enforces the definition of the variables $y_{i,k}$, while constraint (39) captures the restriction that products can only be offered if sufficient resources are available. Constraints (40)-(42) enforce the definition of $z_{j,i,k}$. We note that there are alternative ways to express these constraints.

As before, we aim to apply our general framework to construct a reduced formulation for $(\mathbf{D})^{S,I}$. To that end, we need to formulate $(\mathbf{CG}')^{S,I}$ as a linear program. For the affine approximations in Section 4, simply relaxing the integrality constraints is sufficient because the subproblem polyhedra are integral. Here, however, the linear programming relaxation of $(\mathbf{CG}')_t^{S,I}$ may result in a fractional optimal solution. As a result, Lemma 1 cannot be applied and we resort to Lemma 2 to construct a weakly equivalent reduced formulation.

To apply Lemma 2, we need to identify restrictions on the dual variables V that allow the subproblem to be relaxed. Our key observation is that, absent constraint (42), the resulting constraint matrix is totally unimodular (by the same reasoning used in the proof of Lemma 3). But, constraint (42) is redundant if the decision variables $z_{j,i,k}$ have non-negative objective function coefficients for all j, i and $k > 1$. To see why this is the case, suppose that $q_j = 1$, $y_{i,k} = 1$ and $z_{j,i,k} = 0$ for some j, i and k (i.e., constraint (42) is violated). Then, the non-negative objective function coefficients allow us to set $z_{j,i,k}$ equal to 1 without loss of optimality. Note that constraint (42) is redundant for $k=1$, due to the presence of constraint (39). As a result, $(\mathbf{CG}')_t^{S,I}$ can be solved as a linear program if $V_{t,i,k} \leq V_{t,i,k-1}$ for all t,i,k, because this will guarantee that the objective function coefficients for the decision variables $z_{j,i,k}$ are non-negative.

We comment that the restriction $V_{t,i,k} \leq V_{t,i,k-1}$ for all t,i,k is sufficient to construct a reduced formulation. However, $(\mathbf{CG}')_t^{S,I}$ can be simplified further if, in addition, $V_{t+1,i,k} \leq V_{t,i,k}$ for all t,i,k. If this restriction holds, the decision variables $y_{t,i,k}$ will have non-positive objective function coefficients and constraint (38) becomes redundant. In particular, we can let $y_{t,i,k}$ be equal to $\min_{j:a_{ij}=1} z_{j,i,k}$ without loss of optimality, because constraint (40) guarantees that constraint (38) is satisfied.

The preceding discussion sets up the application of Lemma 2. In particular, suppose that the dual restrictions $V_{t,i,k} \leq V_{t,i,k-1}$ and $V_{t+1,i,k} \leq V_{t,i,k}$ are satisfied for all t,i,k. Then, $(\mathbf{CG})_t^{S,I}$ can be solved using the linear program

$$
\begin{aligned}
(\mathbf{RCG})_t^{S,I} \quad \max_{\mathbf{q},\mathbf{y},\mathbf{z}} \quad & \sum_j \lambda_{t,j} f_j q_j - \sum_{i,k} V_{t,i,k} y_{i,k} - \sum_{i,k} V_{t+1,i,k} \left(\sum_j \lambda_{t,j} a_{ij} (z_{j,i,k} - z_{j,i,k+1}) - y_{i,k} \right) - \theta_t \\
\text{s.t.} \quad q_j &= z_{j,i,1}, && \forall j,i : a_{ij}=1, \quad (44)\\
z_{j,i,k+1} &\leq z_{j,i,k}, && \forall j,i,k : a_{ij}=1, \quad (45)\\
z_{j,i,k} &\leq y_{i,k}, && \forall j,i,k : a_{ij}=1, \quad (46)\\
y_{i,k} &\leq 1, && \forall i,k : a_{ij}=1. \quad (47)
\end{aligned}
$$

Note that the non-negativity constraints as well as the upper bounds on q_j and $z_{j,i,k}$ are redundant, because we assume $z_{j,i,c_i+1}=0$ for all j,i.

Given $(\mathbf{RCG})_t^{S,I}$, we check that conditions (i)-(iii) in Lemma 2 are satisfied. Then, Lemma 2 states that $(\mathbf{D})^{S,I}$ is *weakly equivalent* to the reduced formulation

$$
\begin{aligned}
(\textbf{D-R})^{S,I} \quad \max_{\mathbf{q},\mathbf{y},\mathbf{z}} \quad & \sum_{t,j} \lambda_{t,j} f_j q_{t,j} \\
\text{s.t.} \quad y_{t,i,k} &= \begin{cases} 1, & \text{if } t=1, \\ y_{t-1,i,k} - \sum_{j:a_{ij}=1} \lambda_{t-1,j} (z_{t-1,j,i,k} - z_{t-1,j,i,k+1}), & \text{if } t>1, \end{cases} && \forall t,i,k,\\
q_{t,j} &= z_{t,j,i,1}, && t,j,i : a_{ij}=1, \quad (48)\\
z_{t,j,i,k+1} &\leq z_{t,j,i,k}, && \forall t,j,i,k : a_{ij}=1, \quad (49)\\
z_{t,j,i,k} &\leq y_{t,i,k}, && \forall t,j,i,k : a_{ij}=1. \quad (50)
\end{aligned}
$$

Observe that the upper bounds (47) are redundant in the reduced formulation. Recall that weak equivalence implies that the optimal solutions of $(\mathbf{D})^{S,I}$ and $(\textbf{D-R})^{S,I}$ intersect. In particular, the conditions in Lemma 2 guarantee that there exists an optimal solution to the Dantzig-Wolfe reformulation of $(\mathbf{D\text{-}R})^{S,I}$ that is feasible to $(\mathbf{D})^{S,I}$.

To verify that the conditions in Lemma 2 are satisfied, we first introduce the following lemma.

LEMMA 5. *There exists an optimal solution* $(V^*, \beta^*, \gamma^*, \delta^*)$ *to the dual of* $(\boldsymbol{D\text{-}R})^{S,I}$ *such that* $V^*_{t,i,k} \le V^*_{t,i,k-1}$ *and* $V^*_{t+1,i,k} \le V^*_{t,i,k}$ *for all* t,i,k.

Proof. See Appendix. ■

Lemma 5 confirms the dual optimal inequalities imposed in condition (ii) of Lemma 2. For a given period t, condition (i) in Lemma 2 requires that for every $(\mathbf{x},\mathbf{u}) \in \mathcal{S}$, there exists a $(\mathbf{q},\mathbf{y},\mathbf{z})$ that satisfies constraints (44)-(47) such that

$$\sum_j \lambda_{t,j} f_j u_j = \sum_j \lambda_{t,j} f_j q_j, \tag{51}$$

$$\mathbb{1}\{x_i \ge k\} = y_{i,k}, \qquad \forall i,k, \tag{52}$$

$$\sum_{j:a_{ij}=1} \lambda_{t,j} u_j \mathbb{1}\{x_i = k\} - \mathbb{1}\{x_i \ge k\} = \sum_{j:a_{ij}=1} \lambda_{t,j}(z_{j,i,k} - z_{j,i,k+1}) - y_{i,k}, \qquad \forall i,k. \tag{53}$$

We confirm this by letting $q_j = u_j$, $y_{i,k} = \mathbb{1}\{x_i \ge k\}$, and $z_{j,i,k} = \mathbb{1}\{x_i \ge k\}u_j$ for all j,i,k.

Finally, fix t and suppose that $V_{t,i,k} \le V_{t,i,k-1}$ and $V_{t+1,i,k} \le V_{t,i,k}$ for all i,k. Then, condition (iii) in Lemma 2 requires that there exists an optimal solution $(\mathbf{q}^*,\mathbf{y}^*,\mathbf{z}^*)$ to $(\mathbf{RCG})_t^{S,I}$ such that (51)–(53) are satisfied for some $(\mathbf{x},\mathbf{u}) \in \mathcal{S}$. Let $(\mathbf{q},\mathbf{y},\mathbf{z})$ be any optimal solution to $(\mathbf{RCG})_t^{S,I}$. By our previous discussion, we can let $q^*_j = q_j$ for all j, $y^*_{i,k} = \min_{j:a_{ij}=1} z_{j,i,k}$, and $z^*_{j,i,k} = \min(y^*_{i,k}, q^*_j)$ without loss of optimality. Because the resulting solution will be integral, we can define $x_i = \sum_k y^*_{i,k}$ for all i and $u_j = q^*_j$ for all j. Because $z^*_{j,i,k} = \mathbb{1}\{x_i \ge k\}u_j$ under this definition, it follows that (51)–(53) are satisfied.

To conclude, we verify that the conditions in Lemma 2 are satisfied, and summarize the main result of this section in the following proposition.

PROPOSITION 3. *The linear programs* $(\boldsymbol{D})^{S,I}$ *and* $(\boldsymbol{D\text{-}R})^{S,I}$ *are weakly equivalent.*

Proposition 3 relies on a set of dual restrictions that are satisfied by at least one optimal solution to the reduced program. Our approach to identifying such dual optimal inequalities is to start by formulating the column generation subproblem as an integer program. We then consider conditions under which the formulation can be relaxed, and proceed to construct a reduced formulation *assuming* these conditions are satisfied. Finally, we analyze the structure of the reduced program's dual to verify these conditions.

5.2. Customer Choice Models

We obtain the separable piecewise linear approximation under customer choice by substituting (9) into $(\mathbf{D})^{\phi}$, which yields

$$
\begin{aligned}
(\mathbf{D})^{S,C} \max_{\mathbf{p}} \quad & \sum_{t,(\mathbf{x},\mathbf{u})\in\mathcal{S}} R_t(\mathbf{u}) p_{t,\mathbf{x},\mathbf{u}} \\
\text{s.t.} \quad & \sum_{\substack{(\mathbf{x},\mathbf{u})\in\mathcal{S}:\\ x_i\geq k}} p_{t,\mathbf{x},\mathbf{u}} = \begin{cases} 1, & \text{if } t=1, \\ \sum\limits_{\substack{(\mathbf{x},\mathbf{u})\in\mathcal{S}:\\ x_i\geq k}} p_{t-1,\mathbf{x},\mathbf{u}} - \sum\limits_{\substack{(\mathbf{x},\mathbf{u})\in\mathcal{S}:\\ x_i=k}} Q_{t-1,i}(\mathbf{u}) p_{t-1,\mathbf{x},\mathbf{u}}, & \text{if } t>1, \end{cases} \quad \forall t,i,k, && (54) \\
& \sum_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} p_{t,\mathbf{x},\mathbf{u}} = 1, \quad \forall t, && (55) \\
& \mathbf{p} \geq \mathbf{0}.
\end{aligned}
$$

Meissner and Strauss (2012) solve a relaxation of $(\mathbf{D})^{S,C}$ by aggregating resource inventory levels. Like Zhang and Adelman (2009), they use an integer program to solve the column generation subproblems for a multinomial logit choice model with disjoint consideration sets. As before, we construct reduced formulations for a general customer choice model. For a given t, the column generation subproblem equals

$$
\begin{aligned}
&(\mathbf{CG})_t^{S,C} \\
&\max_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} R_t(\mathbf{u}) - \sum_{i,k} V_{t,i,k} \mathbb{1}\{x_i \geq k\} - \sum_{i,k} V_{t+1,i,k} \left(Q_{t-1,i}(\mathbf{u}) \mathbb{1}\{x_i = k\} - \mathbb{1}\{x_i \geq k\}\right) - \theta_t, \\
&= \max_{(\mathbf{x},\mathbf{u})\in\mathcal{S}} \rho_t \sum_j P_{t,j}(\mathbf{u}) \left[f_j - \sum_{i,k} a_{ij} \left(V_{t+1,i,k} - V_{t+1,i,k-1}\right) \mathbb{1}\{x_i \geq k\} \right] - \sum_{i,k} (V_{t,i,k} - V_{t+1,i,k}) \mathbb{1}\{x_i \geq k\} - \theta_t,
\end{aligned}
$$

with (V,θ) an optimal dual solution to the restricted master problem and boundary conditions $V_{T+1,i,k} = 0$ and $V_{t,i,0} = 0$ for all t,i,k.

Again, the non-linear terms in the objective function pose a challenge. As in the independent demand case, however, we argue that $(\mathbf{CG})_t^{S,C}$ can be formulated as a linear program under certain restrictions on the dual variables V. In particular, suppose that $V_{t+1,i,k} \leq V_{t,i,k}$ for all t,i,k, and consider the following linear program for a given t.

$$
\begin{aligned}
(\mathbf{RCG})_t^{S,C} \max_{\mathbf{h},\mathbf{y},\mathbf{z}} \quad & \sum_{\mathbf{u}\in\mathcal{U}\setminus\{\mathbf{0}\}} R_t(\mathbf{u}) h_{\mathbf{u}} - \sum_{\mathbf{u}\in\mathcal{U}} \sum_{i\in\mathcal{I}(\mathbf{u}),k} \left(V_{t+1,i,k} - V_{t+1,i,k-1}\right) z_{\mathbf{u},i,k} - \sum_{i,k} (V_{t,i,k} - V_{t+1,i,k}) y_{i,k} - \theta_t \\
\text{s.t.} \quad & \sum_{\mathbf{u}\in\mathcal{U}\setminus\{\mathbf{0}\}} h_{\mathbf{u}} \leq 1, && (56) \\
& h_{\mathbf{u}} = z_{\mathbf{u},i,k}, \quad \forall \mathbf{u}\in\mathcal{U}, i\in\mathcal{I}(\mathbf{u}), && (57) \\
& z_{\mathbf{u},i,k+1} \leq z_{\mathbf{u},i,k}, \quad \forall \mathbf{u}\in\mathcal{U}, i\in\mathcal{I}(\mathbf{u}), k, && (58) \\
& \sum_{\mathbf{u}\in\mathcal{U}: i\in\mathcal{I}(\mathbf{u})} z_{\mathbf{u},i,k} \leq y_{i,k}, \quad \forall i,k, && (59)
\end{aligned}
$$

$$y_{i,k} \le 1, \qquad \forall \mathbf{u} \in \mathcal{U}, i \in \mathcal{I}(\mathbf{u}), k. \quad (60)$$

The decision variable $h_{\mathbf{u}} = 1$ iff the offer set corresponding to $\mathbf{u}$ is selected; note that $h_{\mathbf{0}}$ is defined implicitly as the slack variable in constraint (56). We also introduce additional decision variables $\mathbf{y}$ and $\mathbf{z}$, such that $y_{i,k} = 1$ iff $\mathbb{1}\{x_i \ge k\}$, and $z_{\mathbf{u},i,k} = 1$ iff $\mathbb{1}\{x_i \ge k\}$ and $h_{\mathbf{u}} = 1$. Constraint (56) ensures that one offer set is selected, and constraint (57) enforces that products can only be offered if the necessary resources are available. Constraints (58) and (59) enforce the definition of $z_{\mathbf{u},i,k}$. Because we adopt the convention that $z_{\mathbf{u},i,c_i+1} = 0$ for all $\mathbf{u}, i$, all bounds other than (60) are redundant.

To understand why $(\mathbf{RCG})_t^{S,C}$ is equivalent to $(\mathbf{CG})_t^{S,C}$ when $V_{t+1,i,k} \le V_{t,i,k}$ for all t, i, k, suppose that we have an integer optimal solution to $(\mathbf{RCG})_t^{S,C}$. Because the decision variables $y_{i,k}$ have non-positive objective function coefficients and only occur in the right hand side of constraint (59), we can let $y_{i,k} = \sum_{\mathbf{u} \in \mathcal{U}: i \in \mathcal{I}(\mathbf{u})} z_{\mathbf{u},i,k}$ without loss of optimality. Because the solution is integer, there is exactly one $\mathbf{u}'$ such that $h_{\mathbf{u}'} = 1$ and therefore $y_{i,k} = z_{\mathbf{u}',i,k}$ for all i, k. Thus, we confirm that $y_{i,k+1} \le y_{i,k}$ for all i, k by constraint (58) and verify that $z_{\mathbf{u},i,k} = h_{\mathbf{u}} y_{i,k}$ for all $\mathbf{u}, i, k$. We prove that $(\mathbf{RCG})_t^{S,C}$ has integer extreme points using the following lemma.

LEMMA 6. *The polyhedron $\mathcal{Q}^{S,C} = \{(\mathbf{h}, \mathbf{y}, \mathbf{z}) : (56) - (60)\}$ is integral.*

Proof. The proof follows by contradiction. Suppose $(\tilde{\mathbf{h}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ is a fractional extreme point of $\mathcal{Q}^{S,C}$. As a first step we observe that either $\tilde{y}_{i,k} = 1$ or $\tilde{y}_{i,k} = \sum_{\mathbf{u} \in \mathcal{U}: i \in \mathcal{I}(\mathbf{u})} \tilde{z}_{\mathbf{u},i,k}$, since $(\tilde{\mathbf{h}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ is an extreme point. Let $\tilde{\mathcal{I}} = \{(i,k) : \tilde{y}_{i,k} = 1\}$. In addition, let $\tilde{\mathcal{U}} = \{\mathbf{u} \in \mathcal{U} : 0 < \tilde{h}_{\mathbf{u}} < 1\}$ and $L = |\tilde{\mathcal{U}}|$ (note that $\tilde{\mathcal{U}}$ may contain the empty offer set $\mathbf{0}$). We must have $L > 1$, for otherwise $(\tilde{\mathbf{h}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ is integral. Without loss of generality, let $\tilde{\mathcal{U}} = \{\mathbf{u}^l : l = 1, \ldots, L\}$. For each $\mathbf{u} \in \tilde{\mathcal{U}}$, we also define the set $Z_{\mathbf{u}} = \{\tilde{z}_{\mathbf{u},i,k} : i \in \mathcal{I}(u), 1 \le k \le c_i\}$. Let $\upsilon_{\mathbf{u},1} > \cdots > \upsilon_{\mathbf{u},M_{\mathbf{u}}} > 0$ be the distinct positive fractional values in the set $Z_{\mathbf{u}}$. Note that $\upsilon_{\mathbf{u},1} = \tilde{h}_{\mathbf{u}} = \tilde{z}_{\mathbf{u},i,1}$ for all $i \in \mathcal{I}(\mathbf{u})$ by constraints (57) and (58). For convenience, let $\upsilon_{\mathbf{u},M_{\mathbf{u}}+1} = 0$.

Next, we define solutions $(\hat{\mathbf{h}}^{(l,m)}, \hat{\mathbf{y}}^{(l,m)}, \hat{\mathbf{z}}^{(l,m)})$ for $l = 1, \ldots, L$ and $m = 1, \ldots, M_{\mathbf{u}^l}$ by letting

$$\hat{h}_{\mathbf{u}}^{(l,m)} = \begin{cases} 1, \text{ if } \mathbf{u} = \mathbf{u}^l, \\ 0, \text{ otherwise}, \end{cases} \qquad \forall \mathbf{u} \in \mathcal{U},$$

$$\hat{z}_{\mathbf{u},i,k}^{(l,m)} = \begin{cases} 1, \text{ if } \mathbf{u} = \mathbf{u}^l \text{ and } \tilde{z}_{\mathbf{u},i,k} \ge \upsilon_{\mathbf{u},m}, \\ 0, \text{ otherwise}, \end{cases} \qquad \forall \mathbf{u} \in \mathcal{U}, i \in \mathcal{I}(\mathbf{u}), k,$$

$$y_{i,k}^{(l,m)} = \begin{cases} 1, & \text{if } (i,k) \in \tilde{\mathcal{I}}, \\ \hat{z}_{\mathbf{u},i,k}^{(l,m)}, & \text{if } (i,k) \notin \tilde{\mathcal{I}} \text{ and } i \in \mathcal{I}(\mathbf{u}), \\ 0, & \text{otherwise}, \end{cases} \qquad \forall i, k.$$

By construction, each solution $(\hat{\mathbf{h}}^{(l,m)}, \hat{\mathbf{y}}^{(l,m)}, \hat{\mathbf{z}}^{(l,m)})$ is integral and satisfies the constraints in $\mathcal{Q}^{S,C}$. Now, let $\omega_{l,m} = \upsilon_{\mathbf{u}^l,m} - \upsilon_{\mathbf{u}^l,m+1}$ for all $l = 1,\ldots,L$ and $m = 1,\ldots,M_{\mathbf{u}^l}$. In particular, for any $\mathbf{u}^l \in \tilde{\mathcal{U}}$ we have

$$\sum_{l=1}^{L}\sum_{m=1}^{M_{\mathbf{u}^l}} \omega_{l,m}\hat{h}_{\mathbf{u}^l}^{(l,m)} = \sum_{m=1}^{M_{\mathbf{u}^l}} \omega_{l,m} = \sum_{m=1}^{M_{\mathbf{u}^l}} (\upsilon_{\mathbf{u}^l,m} - \upsilon_{\mathbf{u}^l,m+1}) = \upsilon_{\mathbf{u}^l,1} = \tilde{h}_{\mathbf{u}^l}, \text{ and}$$
$$\sum_{l=1}^{L}\sum_{m=1}^{M_{\mathbf{u}^l}} \omega_{l,m}\hat{z}_{\mathbf{u}^l,i,k}^{(l,m)} = \sum_{\substack{m=1:\\ \tilde{z}_{\mathbf{u},i,k}\geq \upsilon_{\mathbf{u},m}}}^{M_{\mathbf{u}^l}} \omega_{l,m} = \sum_{\substack{m=1:\\ \tilde{z}_{\mathbf{u},i,k}\geq \upsilon_{\mathbf{u},m}}}^{M_{\mathbf{u}^l}} (\upsilon_{\mathbf{u}^l,m} - \upsilon_{\mathbf{u}^l,m+1}) = \tilde{z}_{\mathbf{u},i,k}, \quad \forall i,k.$$

Thus, we verify that

$$(\tilde{\mathbf{h}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}}) = \sum_{l=1}^{L}\sum_{m=1}^{M_{\mathbf{u}^l}} \omega_{l,m}\left(\hat{\mathbf{h}}^{(l,m)}, \hat{\mathbf{y}}^{(l,m)}, \hat{\mathbf{z}}^{(l,m)}\right),$$

which implies that $(\tilde{\mathbf{h}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ cannot be an extreme point of $\mathcal{Q}^{S,C}$. ∎

The preceding discussion establishes a linear programming formulation for the column generation subproblem, and restrictions on the dual under which this formulation is valid. Thus, we are in a position to apply Lemma 2. Assuming its conditions are satisfied, Lemma 2 states that $(\mathbf{D})^{S,I}$ is *weakly equivalent* to the following reduced formulation.

$$(\mathbf{D\text{-}R})^{S,C} \quad \max_{\mathbf{h},\mathbf{y},\mathbf{z}} \sum_{t,\mathbf{u}\in\mathcal{U}\setminus\{\mathbf{0}\}} R_t(\mathbf{u})h_{t,\mathbf{u}}$$

$$\text{s.t.} \quad y_{t,i,k} = \begin{cases} 1, & \text{if } t=1,\\ y_{t-1,i,k} - \sum_{\mathbf{u}\in\mathcal{U}} Q_{t-1,i}(\mathbf{u})(z_{t-1,\mathbf{u},i,k} - z_{t-1,\mathbf{u},i,k+1}), & \text{if } t>1, \end{cases} \quad \forall t,i,k, \tag{61}$$

$$\sum_{\mathbf{u}\in\mathcal{U}\setminus\{\mathbf{0}\}} h_{t,\mathbf{u}} \leq 1, \quad \forall t, \tag{62}$$

$$h_{t,\mathbf{u}} = z_{t,\mathbf{u},i,1}, \quad \forall t, \mathbf{u}\in\mathcal{U}, i\in\mathcal{I}(\mathbf{u}), \tag{63}$$

$$z_{t,\mathbf{u},i,k+1} \leq z_{t,\mathbf{u},i,k}, \quad \forall t, \mathbf{u}\in\mathcal{U}, i\in\mathcal{I}(u), k, \tag{64}$$

$$\sum_{\mathbf{u}\in\mathcal{U}: i\in\mathcal{I}(\mathbf{u})} z_{t,\mathbf{u},i,k} \leq y_{t,i,k}, \quad \forall t,i,k. \tag{65}$$

As in the independent demand case, we note that the upper bounds (60) are redundant in the reduced formulation.

To confirm the conditions in Lemma 2, we first introduce the following lemma. The lemma guarantees that the dual restrictions we impose are dual optimal inequalities.

LEMMA 7. *There exists an optimal solution $(V^*, \alpha^*, \beta^*, \gamma^*, \delta^*)$ to the dual of $(\boldsymbol{D\text{-}R})_{S,C}$ such that $V^*_{t+1,i,k} \leq V^*_{t,i,k}$ for all t,i,k.*

Proof. See Appendix. ∎

Condition (i) in Lemma 2 requires that for any given t and $(\mathbf{x},\mathbf{u})\in\mathcal{S}$, there exists a $(\mathbf{h},\mathbf{y},\mathbf{z})\in\mathcal{Q}^{S,C}$ such that

$$R_t(\mathbf{u})=\sum_{\mathbf{u}'\in\mathcal{U}}R_t(\mathbf{u}')h_{\mathbf{u}'}, \tag{66}$$

$$\mathbb{1}\{x_i\geq k\}=y_{i,k}, \qquad \forall i,k, \tag{67}$$

$$Q_{t-1,i}(\mathbf{u})\mathbb{1}\{x_i=k\}-\mathbb{1}\{x_i\geq k\}=Q_{t-1,i}(\mathbf{u})(z_{\mathbf{u},i,k}-z_{\mathbf{u},i,k+1})-y_{i,k}, \qquad \forall i,k. \tag{68}$$

We confirm this by letting $h_{\mathbf{u}'}=\mathbb{1}\{\mathbf{u}'=\mathbf{u}\}$ for all $\mathbf{u}'\in\mathcal{U}$, $y_{i,k}=\mathbb{1}\{x_i\geq k\}$, and $z_{\mathbf{u},i,k}=\mathbb{1}\{x_i\geq k\}\mathbb{1}\{\mathbf{u}'=\mathbf{u}\}$ for all $\mathbf{u}'\in\mathcal{U},i\in\mathcal{I}(\mathbf{u}'),k$. It follows that $(\mathbf{h},\mathbf{y},\mathbf{z})\in\mathcal{Q}^{S,C}$.

Now, fix t and suppose that $V_{t+1,i,k}\leq V_{t,i,k}$ for all i,k. Then, condition (iii) in Lemma 2 requires that there exists an optimal solution $(\mathbf{h},\mathbf{y},\mathbf{z})$ to $(\mathbf{RCG})_t^{S,C}$ such that (66)-(68) are satisfied for some $(\mathbf{x},\mathbf{u})\in\mathcal{S}$. Let $(\mathbf{h},\mathbf{y},\mathbf{z})$ be an optimal solution to $(\mathbf{RCG})_t^{S,C}$. By our previous discussion, we can assume $y_{i,k}=\sum_{\mathbf{u}\in\mathcal{U}:i\in\mathcal{I}(\mathbf{u})}z_{\mathbf{u},i,k}$ without loss of optimality. Because the resulting solution will be integral, we can define $x_i=\sum_k y_{i,k}$ for all i and choose $\mathbf{u}$ such that $h_{\mathbf{u}}=1$. We can verify that (66)-(68) are satisfied for the resulting $(\mathbf{x},\mathbf{u})\in\mathcal{S}$.

This implies that the conditions in Lemma 2 are satisfied, and we express the main result of this section in the following proposition.

Proposition 4. *The linear programs $(\boldsymbol{D})^{S,C}$ and $(\boldsymbol{D\text{-}R})^{S,C}$ are weakly equivalent.*

In contrast to the independent demand model, the proof of Proposition 4 does not require that the dual values $V_{t,i,k}$ are decreasing in the remaining capacity at any given point in time for some optimal solution to the dual of the reduced program, even though it is possible to show that such a solution exists. Intuitively, this is because the decision variables corresponds to offer sets rather than individual products. The size of the reduced program $(\mathbf{D\text{-}R})^{S,C}$ is still exponential in the number of products and therefore is not as compact as the reduced program $(\mathbf{D\text{-}R})^{S,I}$ in the independent demand case.

5.3. Interpretation of the Reduced Programs

The reduced programs considered in the paper admit an interesting probabilistic interpretation. In this section, we discuss a probabilistic interpretation for the reduced program $(\mathbf{D\text{-}R})^{S,I}$ resulting from separable piecewise linear approximation for NRM with independent demand. Let $\{\mathbf{X}_t,\mathbf{U}_t\}_{\forall t}$ be a collection of random variables that track the evolution of the state-action pairs over time, whose distribution is given by $p_{t,x,u}$. Then, comparing the flow-balance constraints in $(\mathbf{D})^{S,I}$ and $(\mathbf{D\text{-}R})^{S,I}$ reveals the following variable definitions:

$$q_{t,j}=\sum_{\mathbf{x}\in\mathcal{X},\mathbf{u}\in\mathcal{U}(\mathbf{x})}u_j p_{t,\mathbf{x},\mathbf{u}}=P(U_{t,j}=1)=\mathsf{E}U_{t,j}, \qquad \forall t,j, \tag{69}$$

$$
\begin{aligned}
z_{t,j,i,k} = \sum_{\substack{(\mathbf{x},\mathbf{u})\in\mathcal{S}:\\ x_i\geq k}} u_j p_{t,\mathbf{x},\mathbf{u}} &= \mathsf{E}[U_{t,j}|X_{t,i}\geq k]P(X_{t,i}\geq k) \\
&= P(X_{t,i}\geq k, U_{t,j}=1), \qquad \forall t,j,i,k: a_{ij}=1, \qquad (70)\\
y_{t,i,k} = \sum_{\substack{(\mathbf{x},\mathbf{u})\in\mathcal{S}:\\ x_i\geq k}} p_{t,\mathbf{x},\mathbf{u}} &= P(X_{t,i}\geq k), \qquad \forall t,i,k, \qquad (71)
\end{aligned}
$$

Therefore, the variable $y_{t,i,k}$ can be interpreted as the probability that the remaining capacity on leg i in period t is at least k. The variable $q_{t,j}$ represents the probability that product j is accepted in period t. The variable $z_{t,j,i,k}$ is the joint probability that the remaining capacity on leg i is at least k and product j is offered in period t.

Using equations (69), the objective function of $(\mathbf{D\text{-}R})^{S,I}$ can be written as $\sum_{t,j}\lambda_j f_j \mathsf{E}U_{t,j}$. Note that the term $\lambda_j f_j \mathsf{E}U_{t,j}$ can be interpreted as expected revenue from class-j customers in period t. Hence, the objective is the total expected revenue from all classes. The constraints in $(\mathbf{D\text{-}R})_{S,I}$ can be written as

$$
\begin{aligned}
&P(X_{t,i}\geq k) = \begin{cases} 1, & \text{if } t=1,\\ P(X_{t-1,i}\geq k) - \sum_j \lambda_j a_{ij} P(X_{t,i}=k, U_{t,j}=1), & \text{if } t>1, \end{cases} && \forall t,i,k, && (72)\\
&P(U_{t,j}=1) = P(X_{t,i}\geq 1, U_{t,j}=1), && \forall t,i,j: a_{ij}=1, && (73)\\
&P(X_{t,i}\geq k+1, U_{t,j}=1) \leq P(X_{t,i}\geq k, U_{t,j}=1), && \forall t,i,j,k: a_{ij}=1, && (74)\\
&P(X_{t,i}\geq k, U_{t,j}=1) \leq P(X_{t,i}\geq k), && \forall t,i,k. && (75)
\end{aligned}
$$

The constraints above have a natural probabilistic interpretation. Constraint (72) enforces time consistency on the marginal distribution of $X_{t,i}$. For $t=1$, it requires that $P(X_{1,i}\geq k)=1$ for all i,k, and therefore forces $X_{1,i}=c_i$. For $t>1$, the constraint enforces the relationship between marginal distributions of $X_{t,i}$ and $X_{t-1,i}$. Constraint (73) enforces the resource requirements: dividing both sides by $P(U_{t,j}=1)$, it states that $P(X_{t,i}\geq 1|U_{t,j}=1)=1$. Constraint (74) requires that the (inverse) cumulative distribution functions on X_t are monotone, conditional on $U_{t,j}=1$. Similarly, if we divide both sides by $P(X_{t,i}\geq k)$, constraint (75) requires that $P(U_{t,j}=1|X_{t,i}\geq k)\leq 1$.

The discussion above shows that the program $(\mathbf{D\text{-}R})^{S,I}$ enforces a set of necessary conditions on the stochastic process $\{(\mathbf{X}_t,\mathbf{U}_t)\}$. It is not hard to see that the conditions imposed are not sufficient to track the full dynamics of $\{(\mathbf{X}_t,\mathbf{U}_t)\}$, because it focuses on the marginal distributions on $\mathbf{X}_t$. Therefore, $(\mathbf{D\text{-}R})^{S,I}$ is a relaxation to the original dynamic programming formulation.

5.4. Lagrangian Relaxation

To further highlight the interpretation of the reduced programs, we briefly discuss their relationship to the Lagrangian relaxation approach for NRM introduced in Topaloglu (2009). A detailed discussion of the equivalence between separable piecewise linear approximations and Lagrangian relaxation approaches can be found in Kunnumkal and Talluri (2011, 2014).

To illustrate this connection, consider the reduced formulation $(\mathbf{D\text{-}R})^{S,I}$ under an independent demand model and take the Lagrangian relaxation with respect to constraint (48) with the associated Lagrangian multiplier β and the corresponding Lagrangian value $\vartheta(\beta)$. Because the resulting problem decomposes by leg, this yields

$$\vartheta(\beta) = \sum_{t,j} \lambda_{t,j} \left[f_j - \sum_i a_{ij} \beta_{t,j,i} \right]^+ + \sum_i \hat{\vartheta}_i(\beta), \tag{76}$$

where

$$\begin{aligned}
\hat{\vartheta}_i(\beta) = \max_{\mathbf{y},\mathbf{z}} \ & \sum_{t,j} \beta_{t,j,i} z_{t,j,i,1} \\
\text{s.t.} \quad & y_{t,i,k} = \begin{cases} 1, & t=1, \\ y_{t-1,i,k} - \sum_{j:a_{ij}=1} \lambda_{t-1,j} a_{ij} (z_{t-1,j,i,k} - z_{t-1,j,i,k+1}), & t>1, \end{cases} && \forall t,k, \\
& z_{t,j,i,k+1} \le z_{t,j,i,k}, && \forall t,j,k : a_{ij}=1, \\
& z_{t,j,i,k} \le y_{t,i,k}, && \forall t,j,k : a_{ij}=1.
\end{aligned}$$

We observe that the dual of this program is equivalent to the linear programming formulation of the dynamic program for a single-leg revenue management problem (see the proof of Lemma 5 for additional details). Therefore, the Lagrangian relaxation (76) is identical to the expression in Proposition 2 in Topaloglu (2009). Since $(\mathbf{D\text{-}R})^{S,I}$ is a linear program, it follows that the Lagrangian bound $\min_\beta \vartheta(\beta)$ equals the optimal solution value of $(\mathbf{D\text{-}R})^{S,I}$.

A similar line of reasoning can also be used to obtain a Lagrangian relaxation for the choice setting. With slight abuse of notation, taking the Lagrangian relaxation with respect to constraint (63) in $(\mathbf{D\text{-}R})^{S,C}$ yields

$$\vartheta(\beta) = \sum_t \max_{\mathbf{u}\in\mathcal{U}} \left[R_t(\mathbf{u}) - \sum_{i\in\mathcal{I}(\mathbf{u})} \beta_{t,\mathbf{u},i} \right] + \sum_i \hat{\vartheta}_i(\beta), \tag{77}$$

where

$$\begin{aligned}
\hat{\vartheta}_i(\beta) = \max_{\mathbf{y},\mathbf{z}} \ & \sum_{t,\mathbf{u}\in\mathcal{U}: i\in\mathcal{I}(\mathbf{u})} \beta_{t,\mathbf{u},i} z_{t,\mathbf{u},i,k} \\
\text{s.t.} \quad & y_{t,i,k} = \begin{cases} 1, & t=1, \\ y_{t-1,i,k} - \sum_{\mathbf{u}\in\mathcal{U}} Q_{t-1,i}(\mathbf{u}) (z_{t-1,\mathbf{u},i,k} - z_{t-1,\mathbf{u},i,k+1}), & t>1, \end{cases} && \forall t,k, \\
& z_{t,\mathbf{u},i,k+1} \le z_{t,\mathbf{u},i,k}, && \forall t, \mathbf{u}\in\mathcal{U}, i\in\mathcal{I}(\mathbf{u}), k, \\
& \sum_{\mathbf{u}\in\mathcal{U}: i\in\mathcal{I}(\mathbf{u})} z_{t,\mathbf{u},i,k} \le y_{t,i,k}, && \forall t,k.
\end{aligned}$$

We leave it to the reader to verify that the dual of this program is again equivalent to the linear programming formulation of the dynamic program for a choice-based single-leg revenue management problem. Thus, solving the Lagrangian dual $\min_\beta \vartheta(\beta)$ yields the same bound as the reduced formulation $(\mathbf{D\text{-}R})^{S,C}$. Note though that in this case the Lagrangian multipliers are offer set specific; for further discussion, we refer to Kunnumkal and Talluri (2014).

6. Numerical Experiments

We conduct numerical experiments on test instances taken from Topaloglu (2009) to investigate the computational performance of the reduced formulations introduced in this paper. The test instances in Topaloglu (2009) consider hub-and-spoke networks with independent demand. There are two flight legs on each spoke, where one is from the hub and the other one is to the hub, and two fare classes on each possible itinerary. Each test instance is labeled by (T, N, κ, ρ), where T is the number of periods, N is the number of non-hub locations, κ is the ratio between high and low fare classes for each itinerary, and $\rho = \frac{\sum_{i,j,t} a_{ij}\lambda_{t,j}}{\sum_i c_i}$ is the load factor. For more details about the test instances, please refer to Topaloglu (2009).[1]

Standard solution approaches for solving the ALP formulation for NRM with independent demand $(\mathbf{D})^{S,I}$ are column generation or constraint sampling. Due to the dramatic reduction in problem size, however, the reduced program $(\mathbf{D\text{-}R})^{S,I}$ can be solved directly using a linear programming solver. Our tests were performed on a Intel Quad Core Q9650 3.00GHZ computer running Windows 7 Professional 64-bit. The formulation was implemented using Visual C++ 2010 and CPLEX 12.3. Our code uses the default interior point implementation in CPLEX, which relies on a barrier method. Note that standard implementations of interior point algorithms for linear programming do not give either upper or lower bounds on optimal objective value in each iteration. Even though it is possible to construct feasible primal and dual solutions from crossover procedures, our experience is that these procedures can take a long time. The upper and lower bounds we obtain are constructed using two recursive procedures. The Appendix gives more details on the construction procedures. The resulting bounds allows us to evaluate the optimality gap during the solution process.

Kunnumkal and Talluri (2011) show that the Lagrangian relaxation is equivalent to the separable piecewise approximation $(\mathbf{D})^{S,I}$; see also the relevant discussion in Section 5.4. Therefore, an alternative approach to solving the separable piecewise linear approximation $(\mathbf{D})^{S,I}$ is to solve the Lagrangian relaxation proposed in Topaloglu (2009). Topaloglu (2009) proposes a subgradient algorithm to solve the Lagrangian relaxation of the NRM problem under independent demand; using

[1] See also `http://people.orie.cornell.edu/huseyin/research/rm_datasets/rm_datasets.html`.

policy simulation, Topaloglu (2009) establishes superiority of the Lagrangian relaxation approach compared with several benchmark policies. Kunnumkal and Talluri (2011) observe in numerical tests that the subgradient algorithm proposed in Topaloglu (2009) is much faster than solving the separable piecewise linear ALP $(\mathbf{D})^{S,I}$ using a column generation approach. When reporting numerical results, we also report the corresponding results in Topaloglu (2009). Note that the computational setup (i.e., CPU speed, memory, etc.) in Topaloglu (2009) is different from ours. In particular, it uses a 2.4GHz CPU. For this reason, the computational times are not directly comparable to ours, but we report the results to give an overall sense on the computational times. In addition to computational setup, the solution time of a subgradient algorithm also depends on parameters that often require some tuning. One potential weakness of the subgradient algorithm used in Topaloglu (2009) is its dependence on an ad-hoc stopping criterion that can cause premature termination. In contrast, solving the reduced program $(\mathbf{D\text{-}R})^{S,I}$ directly does not require parameter tuning and can achieve arbitrary optimality guarantees.

6.1. Computational Results

Tables 1 and 2 report upper bounds, optimality gaps, and computational times when solving $(\mathbf{D\text{-}R})^{S,I}$ with three convergence tolerance parameter values for test instances with 200 periods and 600 periods, respectively. We tested with three convergence tolerance parameter values 10^{-2}, 10^{-3}, and 10^{-4}, since the computational times of interior point algorithms can be greatly affected by the convergence tolerance parameter. The computational times are reported in the last three columns in the tables. The tables show that the computational times for different convergence tolerance values differ dramatically. Understandably, a smaller convergence tolerance means longer time to converge.

We also report the computational times taken from Topaloglu (2009) in the third column of the two tables. With very few exceptions, the solution times for all three convergence tolerance values are shorter than the ones reported in Topaloglu (2009), even though the computational setups are somewhat different. We emphasize that solving $(\mathbf{D\text{-}R})^{S,I}$ directly has at least two merits: (i) adjusting convergence tolerance parameter provides a way to trade-off solution time with solution quality, and (ii) little customization or tuning is required as the solution method uses standard linear programming algorithms.

6.2. Bounds

The fourth to sixth columns in Tables 1 and 2 report upper bounds for the three convergence tolerance values, while the seventh to ninth columns report percentage gaps between upper and lower bounds. The percentage gaps provide an optimality guarantee. The optimality gaps for the majority of problem instances are within 1% for all three convergence tolerance values. The maximum optimality gap is less than 2%. As we decrease the convergence tolerance value, the optimality gaps reduce significantly. The optimality gaps for smaller convergence tolerance values also tend to be smaller. In particularly, for the convergence tolerance value 10^{-4}, the optimality gaps are no larger than 0.02% across the two tables, suggesting that the solutions we obtain are very close to optimality. We point out, however, that for all three convergence tolerance values, the upper bounds are very close, and the optimality gaps are mainly due to the lower bounds. Therefore, even for convergence tolerance value 10^{-2}, the primal solution is already quite close to optimal. Therefore, solving the reduced programs using an interior point algorithm does give high-quality solutions.

We also report the Lagrangian upper bounds from Topaloglu (2009) in the second column in Tables 1 and 2. One immediate observation is that the Lagrangian upper bounds are looser than the upper bounds for all three convergence tolerance values. Therefore, even with a convergence tolerance value 10^{-2}, solving $(\mathbf{D})^{S,I}$ directly gives better upper bounds, even though the computational times overall are considerably shorter. There are instances where the subgradient algorithms converge very quickly, e.g., for problem instance $(600, 8, 1.0, 4)$ in Table 2. However, the reported Lagrangian bound (22960) is more than 1% higher than the three upper bounds from solving $(\mathbf{D\text{-}R})^{S,I}$. This suggests that for this particular problem instance, the subgradient algorithm used to solve the Lagrangian relaxation stopped prematurely.

7. Summary and Future Directions

This paper considers reductions of ALPs for NRM problems resulting from the affine and separable piecewise linear approximations. Our results apply to settings with and without customer choice. Central to our research is the connection between the reduced linear programs and their Dantzig-Wolfe reformulations. To establish equivalence, we explore properties of the underlying polyhedra in the column generation subproblems. In particular, we require that the polyhedra have integral extreme points. For this reason, we do not expect the equivalence to hold for general stochastic dynamic programs. Nevertheless, the idea of Dantzig-Wolfe reformulation is a very general one. Therefore, it would be interesting to explore our idea in other problem contexts.

Throughout the paper, our focus has been on equivalent reductions — the reduced program produces solutions that are exact for the original, much larger ALPs. While equivalence is certainly desirable, an approximate solution may be acceptable. For this reason, we believe there is considerable value to consider reduced programs even when we cannot establish equivalence to original ALPs. In that case, it would be valuable to show performance guarantees or error bounds.

The numerical experiment in Section 6 shows that solving the reduced formulation for separable piecewise linear approximation in the independent demand case yields high quality solutions. The reductions in our paper also open up possibilities to develop specialized algorithms to improve numerical performance, which is a promising research direction not pursued in the current paper. For affine approximations to NRM, Vossen and Zhang (2014) show that a specialized aggregation/disaggregation algorithm dramatically improves numerical performance. The reduced formulation for separable piecewise linear approximation in the choice case is still exponential in the number of products. Our numerical testing suggests that directly solving the reduced formulations is still quite computationally intensive. Therefore, specialized algorithms need to be developed to take advantage of the reduced formulations in this case. We leave this development to future research.

Acknowledgment

The authors thank the department editor Daniel Adelman, the associate editor, and three reviewers for their constructive comments. The authors benefited from discussions with Guillermo Gallego, Sumit Kunnumkal, and Huseyin Topaloglu at different stages of the project.

References

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

Adelman, D., A. J. Mersereau. 2008. Relaxations of weakly coupled stochastic dynamic programs. *Operations Research* **56**(3) 712–727.

Ben Amor, H., J. Desrosiers, J. M. Valério de Carvalho. 2006. Dual-optimal inequalities for stabilized column generation. *Operations Research* **54**(3) 454–463.

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

Bertsimas, D., I. Popescu. 2003. Revenue management in a dynamic network environment. *Transportation Science* **37**(3) 257–277.

Bertsimas, D., J. N. Tsitsiklis. 1997. *Introduction to linear optimization*. Athena Scientific Belmont, MA.

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

Dantzig, G., P. Wolfe. 1960. Decomposition principle for linear programs. *Operations Research* **8**(1) 101–111.

Dantzig, G. B., P. Wolfe. 1961. The decomposition algorithm for linear programs. *Econometrica* **29**(4) 767–778.

de Farias, D., B. Van Roy. 2004. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research* **29**(3) 462–478.

de Farias, D. P., B. Van Roy. 2003. The linear programming approach to approximate dynamic programming. *Operations Research* **51**(6) 850–865.

Desrosiers, J., M. E. Lübbecke. 2005. A primer in column generation. *Column Generation*. Springer US, 1–32.

Farias, V. F., B. Van Roy. 2007. An approximate dynamic programming approach to network revenue mangement. Working paper, MIT Sloan School of Management.

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

Kunnumkal, S., K. Talluri. 2011. Equivalence of piecewise-linear approximation and lagrangian relaxation for network revenue management. Working paper, Barcelona Graduate School of Economics.

Kunnumkal, S., K. Talluri. 2014. On the tractability of the piecewise-linear approximation for general discrete-choice network revenue management. Working paper, Barcelona Graduate School of Economics.

Lautenbacher, C., S. Stidham. 1999. The underlying Markov decision process in the single-leg airline yield management problem. *Transportation Science* **33**(2) 136–146.

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.

Powell, W. 2007. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience, Hoboken, New Jersey.

Puterman, M. L. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York.

Schweitzer, P. J., A. Seidmann. 1985. Generalized polynomial approximations in Markovian decision processes. *Journal of Mathematical Analysis and Applications* **110**(2) 568–582.

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

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.

Tardella, F. 1990. On the equivalence between some discrete and continuous optimization problems. *Annals of Operations Research* **25**(1) 291–300.

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.

Vossen, T., D. Zhang. 2014. A dynamic disaggregation approach to approximate linear programs for network revenue management. Forthcoming, *Production and Operations Management*.

Wolsey, L. 1998. *Integer Programming*. Wiley, New York.

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.

Appendix

Summary of Notation

I	The number of legs
J	The number of products
T	The number of periods
$\mathcal{I}$	The set of legs: $\mathcal{I} = \{1, \ldots, I\}$
$\mathcal{J}$	The set of products: $\mathcal{J} = \{1, \ldots, J\}$
$\mathcal{T}$	The set of periods: $\mathcal{T} = \{1, \ldots, T\}$
i	Leg index: $i \in \mathcal{I}$
j	Product index: $j \in \mathcal{J}$
t	Time period: $t \in \mathcal{T}$
$\mathbf{A}$	Consumption matrix of dimension $I \times J$ with entries $a_{ij} \in \{0, 1\}$
$\mathbf{a}^j$	The j-th column of $\mathbf{A}$
$\mathbf{c}$	Resource capacity vector: $\mathbf{c} = (c_1, \ldots, c_I) \in \mathbb{Z}_+^I$
f_j	Fare of product j, $j \in \mathcal{J}$
ρ_t	The probability of a customer arrival during period t, $t \in \mathcal{T}$
$\mathcal{U}$	The set of all possible (characteristic vectors of) offer sets: $\mathcal{U} = \{0, 1\}^J$
$\mathcal{U}(\mathbf{x})$	Feasible (characteristic vectors of) offer sets in state $\mathbf{x}$: $\mathcal{U}(\mathbf{x}) = \{\mathbf{u} \in \mathcal{U} : \mathbf{a}^j u_j \leq \mathbf{x}, \forall j\}$
$P_{t,j}(\mathbf{u})$	The probability that an arriving customer will choose product j during period t when the offer set is $\mathbf{u}$
$\lambda_{t,j}$	The probability that a class-j customer arrives during period t, $t \in \mathcal{T}$, $j \in \mathcal{J}$, for the independent demand model
$\mathcal{X}$	State space: $\mathcal{X} = \{\mathbf{x} \in \mathbb{Z}_+^I : \mathbf{x} \leq \mathbf{c}\}$
$\mathcal{S}$	Set of feasible state-action pairs: $\mathcal{S} = (\mathbf{x}, \mathbf{u}) \in \mathcal{S} \equiv \{(\mathbf{x}, \mathbf{u}) \in \mathcal{X} \times \mathcal{U} : \mathbf{u} \in \mathcal{U}(\mathbf{x})\}$
$\mathcal{I}_j$	The set of legs used by product $j \in \mathcal{J}$: $\mathcal{I}_j = \{i \in \mathcal{I} : a_{ij} = 1\}$
$\mathcal{I}(\mathbf{u})$	The set of legs used by offer set $\mathbf{u}$: $\mathcal{I}(\mathbf{u}) \equiv \bigcup_{j : u_j = 1} \mathcal{I}_j$

Proof of Lemma 5

The dual of $(\mathbf{D\text{-}R})^{S,I}$ equals

$$
\begin{aligned}
(\mathbf{P\text{-}R})^{S,I} \quad & \max_{V,\beta,\gamma,\delta} \sum_i \sum_{k=1}^{c_i} V_{1,i,k} \\
\text{s.t.} \quad V_{t,i,k} - V_{t+1,i,k} &= \sum_{j : a_{ij} = 1} \delta_{t,j,i,k} && \forall t, i, k, && (78) \\
\delta_{t,j,i,k} - \gamma_{t,j,i,k} &= \begin{cases} \beta_{t,j,i} - \lambda_{t,j} V_{t+1,i,1}, & \text{if } k = 1, \\ \lambda_{t,j}(V_{t+1,i,k-1} - V_{t+1,i,k}) - \gamma_{t,j,i,k-1}, & \text{if } k > 1, \end{cases} \\
& && \forall t, i, j, k : a_{ij} = 1, && (79) \\
\sum_i a_{ij} \beta_{t,j,i} &= \lambda_{t,j} f_j, && \forall t, j, && (80) \\
\gamma_{t,j,i,k}, \delta_{t,j,i,k} &\geq 0, && \forall t, i, j, k : a_{ij} = 1. && (81)
\end{aligned}
$$

Observe that constraint (78), together with the non-negativity of $\delta_{t,j,i,k}$, implies that $V_{t+1,i,k} \leq V_{t,i,k}$ for all t, i, k in any dual solution $(V, \beta, \gamma, \delta)$.

To prove there exists an optimal dual solution $(V^*, \beta^*, \gamma^*, \delta^*)$ such that $V^*_{t,i,k} \leq V^*_{t,i,k-1}$ for all t, i, k, we first observe that $(\mathbf{P\text{-}R})^{S,I}$ decomposes by leg for a given β. Then, we show that the resulting subproblems are equivalent to the dynamic programming formulation of a single-leg resource revenue management problem. This completes the proof, as it is well-known that the value function in a single-leg revenue management problem is marginally decreasing in the remaining capacity at any given point in time (Lautenbacher and Stidham 1999).

Specifically, for any given β that satisfies constraint (80) we can obtain an optimal solution to $(\mathbf{P\text{-}R})^{S,I}$ by solving the following subproblems for all i:

$$
\begin{aligned}
(\mathbf{P\text{-}R})_i(\beta) \ \max_{V,\delta,\gamma} \ & \sum_{k=1}^{c_i} V_{1,i,k} \\
\text{s.t.} \quad V_{t,i,k} - V_{t+1,i,k} \ &= \sum_{j:a_{ij}=1} \delta_{t,j,i,k} && \forall t, k, \quad (82)\\
\delta_{t,j,i,k} - \gamma_{t,j,i,k} \ &= \begin{cases} \beta_{t,j,i} - \lambda_{t,j} V_{t+1,i,1}, & \text{if } k = 1, \\ \lambda_{t,j}(V_{t+1,i,k-1} - V_{t+1,i,k}) - \gamma_{t,j,i,k-1}, & \text{if } k > 1, \end{cases} \\
& && \forall t, j, k : a_{ij} = 1, \quad (83)\\
\gamma_{t,j,i,k}, \delta_{t,j,i,k} \ &\geq 0, && \forall t, j, k : a_{ij} = 1. \quad (84)
\end{aligned}
$$

To clarify the relation to the dynamic programming formulation, we reformulate $(\mathbf{P\text{-}R})_i(\beta)$ as

$$
\begin{aligned}
(\mathbf{P\text{-}R})'_i(\beta) \ \max_{\vartheta,\Delta} \ & \vartheta_{1,i,c_i} \\
\text{s.t.} \quad \vartheta_{t,i,k} - \vartheta_{t+1,i,k} \ &= \sum_{j:a_{ij}=1} \Delta_{t,j,i,k}, && \forall t, k, \quad (85)\\
\Delta_{t,j,i,k} \ &\geq \beta_{t,j,i} - \lambda_{t,j} \left(\vartheta_{t,i,k} - \vartheta_{t,i,k-1} \right), && \forall t, j, k : a_{ij} = 1, \quad (86)\\
\Delta_{t,j,i,k} \ &\geq \Delta_{t,j,i,k-1}, && \forall t, j, k : a_{ij} = 1. \quad (87)
\end{aligned}
$$

In the above, we apply the change of variables $\vartheta_{t,i,k} = \sum_{k'=1}^{k} V_{t,i,k'}$ and $\Delta_{t,j,i,k} = \sum_{k'=1}^{k} V_{t,i,k'}$ for all t, j, i, k. Note that $\vartheta_{t,i,0} = 0$ for all t, i and $\Delta_{t,j,i,0} = 0$ for all $t, j, i : a_{ij} = 1$. We obtain constraints (85) and (86) by adding the corresponding constraints (82) and (83) for $k' = 1, \ldots, k$. This effectively transforms the decision variables $\gamma_{t,j,i,k}$ into surplus variables in constraint (86), which justifies their removal from the formulation. Constraint (87) derives from the non-negativity of $\delta_{t,j,i,k}$. To see that $(\mathbf{P\text{-}R})_i(\beta)$ is equivalent $(\mathbf{P\text{-}R})'_i(\beta)$, suppose we have a feasible solution $(\tilde{\vartheta}, \tilde{\Delta})$ to $(\mathbf{P\text{-}R})'_i(\beta)$. Then, the solution $\tilde{V}_{t,i,k} = \tilde{\vartheta}_{t,i,k} - \tilde{\vartheta}_{t,i,k-1}$ for all t, k, $\tilde{\delta}_{t,j,i,k} = \tilde{\Delta}_{t,j,i,k} - \tilde{\Delta}_{t,j,i,k-1}$ for all t, j, k, and $\tilde{\gamma}_{t,j,i,k} = \beta_{t,j,i} - \lambda_{t,j} \tilde{V}_{t,i,k} - \tilde{\Delta}_{t,j,i,k}$ for all t, j, k is feasible to $(\mathbf{P\text{-}R})_i(\beta)$.

Observe that $(\mathbf{P\text{-}R})'_i(\beta)$ is an equivalent linear programming formulation of a single-leg revenue management problem, in that the decision variables ϑ correspond to the value function of the single-leg dynamic program and constraints (85) and (86) express the DP recursion

$$
\vartheta_{t,i,k} = \vartheta_{t+1,i,k} + \sum_{j:a_{ij}=1} \left[\beta_{t,j,i} - \lambda_{t,j} \left(\vartheta_{t,i,k} - \vartheta_{t,i,k-1} \right) \right]^+, \quad \forall t, i, k.
$$

Note that constraint (87) is in fact redundant, because the LP formulation using only (85) and (86) yields an optimal solution such that $V_{t,i,k} = \vartheta_{t,i,k} - \vartheta_{t,i,k-1} \leq \vartheta_{t,i,k-1} - \vartheta_{t,i,k-2} = V_{t,i,k-1}$. Aside from the concavity result we set out to show, this also implies that the right hand side in constraint (86) is increasing in k for all t, j.

Proof of Lemma 7

The dual of $(\mathbf{D\text{-}R})^{S,C}$ equals

$$
\begin{aligned}
(\mathbf{P\text{-}R})^{S,C} \quad & \min_{V,\alpha,\beta,\gamma,\delta} \sum_t \alpha_t + \sum_{i,k} V_{1,i,k} \\
\text{s.t.} \quad & V_{t,i,k} - V_{t+1,i,k} = \delta_{t,i,k}, && \forall t,i,k, \quad (88)\\
& \delta_{t,i,k} - \gamma_{t,\mathbf{u},i,k} = \begin{cases} \beta_{t,\mathbf{u},i} - Q_{t,i}(\mathbf{u}) V_{t+1,i,k}, & \text{if } k=1,\\ Q_{t,i}(\mathbf{u})\left[V_{t+1,i,k-1} - V_{t+1,i,k}\right] - \gamma_{t,\mathbf{u},i,k-1}, & \text{if } k>1, \end{cases} \\
& && \forall t, \mathbf{u}\in\mathcal{U}, i \in \mathcal{I}(\mathbf{u}), k, \quad (89)\\
& \alpha_t + \sum_{i\in\mathcal{I}(\mathbf{u})} \beta_{t,\mathbf{u},i} \geq R_t(\mathbf{u}), && \forall t, \mathbf{u}\in\mathcal{U}, \quad (90)\\
& \gamma_{t,\mathbf{u},i,k}, \delta_{t,i,k} \geq 0, && \forall t, \mathbf{u}\in\mathcal{U}, i, k. \quad (91)
\end{aligned}
$$

Constraint (88), together with the non-negativity of $\delta_{t,i,k}$, implies $V_{t+1,i,k} \leq V_{t,i,k}$ for all t, i, k.

Procedures to Construct Primal and Dual Feasible Solutions

The corresponding primal formulation of $(\mathbf{D\text{-}R})^{S,I}$ is given by $\mathbf{(P\text{-}R)}^{S,I}$. Given an infeasible solution $(\tilde{\mathbf{V}}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta})$ to $\mathbf{(P\text{-}R)}^{S,I}$, a feasible solution $(\hat{\mathbf{V}}, \hat{\beta}, \hat{\gamma}, \hat{\delta})$ can be constructed using the following recursive procedure.

Initialization: Take $\hat{V}_{T+1,i,k} = 0$ for all i, k, $\hat{\beta}_{t,j,i} = \sigma_{t,j}\tilde{\beta}_{t,j,i}$.

Backward recursion:

For $t = T, \dots, 1$; $k = 1, \dots, c_i$

$$
\hat{\delta}_{t,j,i,k} = \begin{cases} [\hat{\beta}_{t,j,i} - \lambda_{t,j}\hat{V}_{t+1,i,k}]^+, & \text{if } k=1,\\ [\lambda_{t,j}(\hat{V}_{t+1,i,k-1} - \hat{V}_{t+1,i,k}) - \hat{\gamma}_{t,j,i,k-1}]^+, & \text{if } k>1, \end{cases} \forall j,i,k : a_{ij} = 1,
$$

$$
\hat{\gamma}_{t,j,i,k} = \begin{cases} [\hat{\beta}_{t,j,i} - \lambda_{t,j}\hat{V}_{t+1,i,k}]^-, & \text{if } k=1,\\ [\lambda_{t,j}(\hat{V}_{t+1,i,k-1} - \hat{V}_{t+1,i,k}) - \hat{\gamma}_{t,j,i,k-1}]^-, & \text{if } k>1, \end{cases} \forall j,i,k : a_{ij} = 1,
$$

$$
\hat{V}_{t,i,k} = \hat{V}_{t+1,i,k} + \sum_j a_{ij}\hat{\delta}_{t,j,i,k}, \forall i,k.
$$

End For

In the above, $\sigma_{t,j} = \frac{\lambda_{t,j} f_j}{\sum_i a_{ij}\tilde{\beta}_{t,j,i}}$ is a scale factor to ensure feasibility. We note that the backward recursion does in fact solve, for each leg, the dynamic programming formulation of the corresponding single-leg revenue management problem (see the proof of Lemma 5 for additional details).

Similarly, given an infeasible solution $(\tilde{\mathbf{q}}, \tilde{\mathbf{y}}, \tilde{\mathbf{z}})$ to $(\mathbf{D\text{-}R})^{S,I}$, a feasible solution $(\hat{\mathbf{q}}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ can be constructed using the following constructive procedure.

Initialization: $\hat{y}_{1,i,k} = 1$ for all i, k.

Forward recursion:

For $t = 1, \ldots, T$

$$\hat{q}_{t,j} = \min\{\tilde{q}_{t,j}, \min_{i:a_{ij}=1} \hat{y}_{t,i,1}\}, \qquad \forall j.$$

For $k = 1, \ldots, c_i$

$$\hat{z}_{t,j,i,k} = \begin{cases} \hat{q}_{t,j}, & \text{if } k = 1, \\ \min\{\hat{y}_{t,i,k}, \hat{z}_{t,j,i,k-1}\}, & \text{if } k > 1, \end{cases} \qquad \forall j, i, k : a_{ij} = 1,$$

$$\hat{y}_{t+1,i,k} = \hat{y}_{t,i,k} - \sum_j \lambda_{t,j} a_{ij} (\hat{z}_{t,j,i,k} - \hat{z}_{t,j,i,k+1}), \qquad \forall i, k.$$

End For

End For

Instance	Lagrangian Relaxation		Tolerance 10^{-2}			Tolerance 10^{-3}			Tolerance 10^{-4}		
	UB	Time (s.)	UB	Gap	Time (s.)	UB	Gap	Time (s.)	UB	Gap	Time (s.)
(200,4,1.0,4)	20439	103	20423	0.71%	23.2	20411	0.08%	33.8	20411	0.00%	63.6
(200,4,1.0,8)	33305	104	33243	0.76%	21.8	33229	0.05%	35.5	33229	0.00%	59.1
(200,4,1.2,4)	18938	106	18866	0.69%	18.3	18857	0.10%	25	18856	0.00%	40
(200,4,1.2,8)	31737	177	31633	0.96%	16.8	31615	0.09%	26.6	31614	0.01%	40.2
(200,4,1.6,4)	16600	93	16522	0.65%	12.4	16508	0.10%	17.9	16507	0.00%	29.3
(200,4,1.6,8)	29413	82	29223	0.70%	12.1	29210	0.11%	16.5	29208	0.01%	25
(200,5,1.0,4)	21298	161	21268	0.73%	35.6	21259	0.09%	53.5	21257	0.01%	81.4
(200,5,1.0,8)	34393	244	34340	1.11%	33.7	34325	0.11%	50.1	34323	0.01%	80
(200,5,1.2,4)	20184	78	20097	0.56%	24.5	20090	0.08%	36.3	20089	0.00%	67.9
(200,5,1.2,8)	33165	159	33042	1.00%	23.8	33028	0.12%	35.3	33027	0.01%	63.2
(200,5,1.6,4)	17704	46	17645	0.93%	18.3	17626	0.11%	29.7	17625	0.01%	41.9
(200,5,1.6,8)	30594	82	30488	0.97%	16.5	30459	0.17%	23.8	30457	0.02%	35.9
(200,6,1.0,4)	21128	231	21089	0.66%	46.5	21077	0.06%	79.6	21075	0.01%	105
(200,6,1.0,8)	34178	67	34082	0.71%	43.4	34061	0.09%	70.3	34059	0.01%	97.8
(200,6,1.2,4)	19649	106	19619	0.85%	37.1	19604	0.11%	61.9	19602	0.03%	85.7
(200,6,1.2,8)	32566	190	32502	1.07%	37.2	32478	0.13%	59.1	32475	0.02%	83.3
(200,6,1.6,4)	17304	107	17242	0.82%	26.3	17228	0.10%	36.6	17227	0.01%	51.4
(200,6,1.6,8)	30170	260	30040	0.92%	24.5	30026	0.10%	36.9	30024	0.00%	56.4
(200,8,1.0,4)	18975	45	18729	1.03%	91.7	18714	0.14%	141.4	18712	0.02%	214.2
(200,8,1.0,8)	30490	196	30206	1.93%	120.2	30199	0.21%	185	30198	0.02%	266
(200,8,1.2,4)	17472	245	17438	1.05%	77	17427	0.14%	118.9	17426	0.02%	168.8
(200,8,1.2,8)	28908	291	28835	0.90%	69.7	28814	0.09%	117.6	28813	0.00%	194.3
(200,8,1.6,4)	15295	205	15249	1.21%	47.6	15242	0.13%	67.2	15241	0.01%	94.1
(200,8,1.6,8)	26661	135	26503	1.03%	50.7	26491	0.14%	69.1	26489	0.01%	99.3

Table 1 Upper bounds, optimality gaps, and computational times when solving $(\mathbf{D\text{-}R})^{S,I}$ with three convergence tolerance parameter values for test instances with 200 periods. Note: Values for upper bounds and computational times for Lagrangian relaxation are taken from Topaloglu (2009).

Instance	Lagrangian Relaxation		Tolerance 10^{-2}			Tolerance 10^{-3}			Tolerance 10^{-4}		
	UB	Time (s.)	UB	Gap	Time (s.)	UB	Gap	Time (s.)	UB	Gap	Time (s.)
(600,4,1.0,4)	30995	731	30983	0.61%	163.5	30970	0.12%	216.8	30969	0.01%	327.8
(600,4,1.0,8)	50444	844	50397	1.08%	145.9	50373	0.12%	220.1	50371	0.01%	376.7
(600,4,1.2,4)	28668	333	28601	0.85%	125.3	28583	0.11%	177.6	28580	0.02%	238.5
(600,4,1.2,8)	48054	985	47934	0.95%	116.4	47903	0.12%	167.5	47897	0.02%	233.3
(600,4,1.6,4)	25148	1301	25074	0.68%	81.5	25054	0.14%	123.9	25051	0.03%	173.4
(600,4,1.6,8)	44555	494	44328	0.63%	77	44313	0.11%	106	44310	0%	177.3
(600,5,1.0,4)	32254	1431	32225	0.93%	245.7	32206	0.11%	451.2	32203	0.01%	769.1
(600,5,1.0,8)	52071	1211	52012	1.12%	241.5	51986	0.11%	455.4	51983	0.01%	716.1
(600,5,1.2,4)	30604	2430	30539	0.96%	167.4	30524	0.15%	260.9	30522	0.02%	478.5
(600,5,1.2,8)	50282	669	50148	1.08%	173.2	50131	0.07%	308.3	50129	0%	530.3
(600,5,1.6,4)	26936	462	26859	0.82%	121.6	26821	0.09%	219	26817	0.02%	362.5
(600,5,1.6,8)	46497	297	46351	1.14%	105.2	46299	0.15%	227.1	46293	0.01%	338.6
(600,6,1.0,4)	25541	1006	25519	0.77%	248.8	25500	0.09%	439	25497	0.01%	754.8
(600,6,1.0,8)	41412	1031	41229	0.83%	262.2	41202	0.1%	436	41199	0.01%	790.3
(600,6,1.2,4)	23687	745	23667	0.68%	182.5	23652	0.08%	276.2	23649	0.01%	437.1
(600,6,1.2,8)	39307	812	39241	1.21%	184.1	39215	0.13%	289	39210	0.02%	477.3
(600,6,1.6,4)	20817	463	20747	1.17%	133.8	20733	0.11%	190.2	20731	0.02%	252.8
(600,6,1.6,8)	36391	1231	36237	1.67%	130.4	36213	0.19%	186.9	36211	0.02%	253.1
(600,8,1.0,4)	22960	11	22712	1.97%	520.5	22704	0.17%	879.8	22704	0.02%	1155.9
(600,8,1.0,8)	36933	10	36639	1.15%	456.5	36614	0.16%	760.9	36612	0.01%	1221.9
(600,8,1.2,4)	21102	1076	21049	1.96%	363.2	21043	0.18%	550	21042	0.02%	783.9
(600,8,1.2,8)	34931	857	34840	0.78%	327.2	34821	0.06%	606.2	34820	0%	954.6
(600,8,1.6,4)	18500	974	18460	1.42%	207.3	18454	0.17%	289.8	18452	0.02%	418.9
(600,8,1.6,8)	32247	717	32092	1.42%	211.8	32080	0.17%	299.8	32078	0.02%	411.2

Table 2 Upper bounds, optimality gaps, and computational times when solving $(\mathbf{D\text{-}R})^{S,I}$ with three convergence tolerance parameter values for test instances with 600 periods. Note: Values for upper bounds and computational times for Lagrangian relaxation are taken from Topaloglu (2009).