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. Recently, Tong and Topaloglu (2011) show that the ALP resulting from an affine approximation for network revenue management under independent demand can be reduced to an equivalent, but much smaller linear program. They show that the reduced program can be solved much faster compared with the original ALPs. We offer a more concise proof of the reduction by exploring the relationship between the affine ALP and an appropriate Dantzig-Wolfe reformulation. More importantly, we show that the new proof technique can be applied to obtain dramatic reductions of ALPs from (i) separable piecewise linear approximation for network revenue management under independent demand model, and (ii) affine approximation for network revenue management with customer choice behavior.

This version: January 7, 2013

1. Introduction

Network revenue management (RM) 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 network RM, the state of the system is the vector of remaining resources, where each resource corresponds to a flight leg. The dynamic programming formulation therefore 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 the last two decades (Talluri and van Ryzin 1998, Bertsimas and Popescu 2003).

The path-breaking paper of Adelman (2007) introduces a solution framework based on an equivalent linear programming formulation of the corresponding dynamic program. His work builds on the stream of literature on linear programming based approximate dynamic programming (LP-based ADP) (Schweitzer and Seidmann 1985, de Farias and Van Roy 2003, de Farias and Van Roy

* Authors are listed alphabetically. Both authors contributed equally.
The central idea is to approximate the value function with linearly weighted basis functions. He illustrates the idea by implementing an affine approximation where the value function is approximated by an affine function of the state (resource vector). The resulting approximate linear program (ALP) has a relatively small number of variables but a huge number of constraints, growing exponentially in the number of resources and the number of products. A column generation procedure was proposed to solve the dual of the ALP, where the subproblems are linear mixed integer programs. The coefficients of each resource in this affine approximation can be interpreted as time-dependent dynamic bid-prices, which produce a control policy superior to the static bid-prices obtained using a widely used deterministic linear program (DLP) (Talluri and van Ryzin 1998, Cooper 2002).

Adelman’s work inspired the development of stronger functional approximations. It is known that the stronger the functional approximation, the stronger the bounds from the corresponding ALP. Even though stronger bounds does not guarantee stronger heuristic policies, numerical studies indeed found positive correlations between the two (Talluri 2008). A powerful and intuitively appealing functional approximation is the separable piecewise linear approximation, which was used in many ADP applications (Bertsekas and Tsitsiklis 1996, Powell 2007). In network RM contexts, this approximation is recently 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 compared with the ALPs from the affine approximation.

Both affine and separable piecewise linear approximations lead to large-scale linear programs that are computationally challenging, even when we are equipped with the powerful modern linear programming tools. The two standard approaches to tackle the computational challenges are column generation and constraint sampling. Adelman (2007) proposes a column generation approach to the affine ALP. Column generation is also employed in Meissner and Strauss (2012). Farias and Van Roy (2007) considers a constraint sampling approach. Their research is based on the earlier work of de Farias and Van Roy (2004). The fundamental idea of both approaches are to successively solve smaller versions of the ALPs, because brute force solutions are infeasible.

An interesting research question is whether it is possible to reduce the size of ALPs. Farias and Van Roy (2007) consider a relaxed ALP (which they call rALP), which is shown to be equivalent to the original ALP under affine approximation. For 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 formulation is still an exponential formulation. More recently, Tong and Topaloglu (2011) show great promise, both theoretically and computationally, of reducing ALPs from affine approximations. They prove that the affine ALP in Adelman (2007) can be reduced a-priori to a more compact linear program, which grows linearly rather exponentially in model primitives (number of resources, number of products, etc.). They illustrate via numerical experiments that the reduced program can be solved orders of magnitude faster than the original ALP. The significant improvement in computational speed is not totally unexpected given the dramatic reduction of problem size. The reduction proof in Tong and Topaloglu (2011) is based on relating the dual of the affine ALP to a network flow problem.

We show that the reduction result obtained in Tong and Topaloglu (2011) can be much more general. Instead of relating the dual to a network problem, we show that the same reduction for affine approximation can be obtained by relating the original ALP with the Dantzig-Wolfe reformulation of the reduced program. Since Dantzig-Wolfe reformulation can be applied to essentially all linear programs, our proof seems to be much more general. Indeed, we are able to apply the same idea to show similar reductions of (i) separable piecewise linear approximation for network RM under independent demand, and (ii) affine approximations for choice-based network RM.

The proof technique proposed in this research is quite general. It involves the following steps. First, once an ALP is formulated, a reduced program can be found by variable aggregation. Second, an appropriate subset of the constraints is considered in a Dantzig-Wolfe reformulation. In order to establish the equivalence between the reduced program and the original ALP, it is sufficient to show that the subset of constraints has integral extreme points. Even though we only consider network RM problems in this paper, we believe the proof technique can be generalized to other problem contexts.

For separable piecewise linear approximation, we show the ALP can be reduced to a program that grows linearly in model primitives, including the number of resources, the number of products, and the number of resource levels. Even though this reduced program can still be quite large, it is much more compact compared with the original ALP. Our preliminary numerical experiments show that it can be solved orders of magnitude faster than the original ALP. We point out that existing approaches (such as constraint sampling) aim at solving the ALP approximately. Given the equivalence result, it suffices to consider approximate solution of the reduced program, which should further reduce the solution time.

Interestingly, the reduced program from separable piecewise linear approximation has a very
intuitive interpretation. 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. As a by-product of our reduction, we show that the reduced program is equivalent to the Lagrangian relaxation proposed in Topaloglu (2009). Therefore, we obtain an alternative proof of the equivalence between separable piecewise linear approximation and Lagrangian relaxation (Kunnumkal and Talluri 2011). The Lagrangian relaxation is a somewhat distinctive approach in the ADP literature, which is also considered in the earlier work of Adelman and Mersereau (2008).

We can show that the affine ALP can be significantly reduced for choice-based network RM (Talluri and van Ryzin 2004, Gallego et al. 2004, Zhang and Cooper 2005, Liu and van Ryzin 2008). The affine ALP for choice-based network RM was first considered by Zhang and Adelman (2009). For choice-based network RM, the decision in each period is the set of products to be offered to customers, which is often a difficult combinatorial optimization problem. The reduction result we obtained is a direct generalization of the reduction to the choice setup.

The remainder of the paper is organized as follows. Section 2 formulates the network RM problem. Sections 3 considers reductions of affine approximations for network RM under independent demand. Section 4 considers reductions and interpretations of separable piecewise linear approximation. Section 5 considers reductions to affine approximations for choice-based network RM. Section 6 summarizes and points out future research directions.

2. Model Formulation and Preliminaries

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

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

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

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

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

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

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

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


\begin{align*}
\text{(LP)} \quad \min_{\{v(t)\}_{t=1}^\tau} v_1(c) \\
v_t(x) \geq \sum_j \lambda_{t,j} [f_j u_j + v_{t+1}(x - A^j u_j)] + \left(1 - \sum_j \lambda_{t,j}\right) v_{t+1}(x), & \quad \forall t, x \in \mathcal{X}_t, u \in \mathcal{U}(x). 
\end{align*}

In the above,

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

where $\mathbb{Z}_+$ denotes nonnegative integers. Since (LP) is a minimization problem, any feasible solution to (LP) gives an upper bound to the total expected revenue $v_1(c)$ from the dynamic programming formulation (Adelman 2007).

The number of variables and constraints in (LP) increases exponentially in the number of resources $m$. Therefore, brute-force solution of (LP) is at least as difficult as solving the dynamic programming formulation (1) directly.

### 2.1. Affine Approximation

One approach to reducing (LP) is approximating the value function $v_t(x)$ by weighted basis functions. Adelman (2007) demonstrates this idea using the affine functional approximation

$$v_t(x) \approx \theta_t + \sum_i V_{t,i} x_i, \quad \forall t, x,$$
where it is assumed that $\theta_{t+1} = 0$ and $V_{t+1,i} = 0$ for all $i$. Plugging (2) into (LP) leads to the following linear program:

\[(LP1) \quad z_1 = \min_{\theta, V} \theta_1 + \sum_i V_{1,i} c_i \]

\[\theta_t - \theta_{t+1} + \sum_i \left[ V_{t,i} x_i - V_{t+1,i} \left( x_i - \sum_j \lambda_{t,j} a_{ij} u_j \right) \right] \geq \sum_j \lambda_{t,j} f_j u_j, \quad \forall t, x \in \mathcal{X}_t, u \in \mathcal{U}(x).\]

Since an optimal solution to (LP1) offers a feasible solution for (LP), the objective value $z_1$ of (LP1) gives an upper bound to $v_1(c)$. Adelman (2007) proves that $z_1$ is a tighter upper bound than the widely used revenue bound from the so-called deterministic linear program (Talluri and van Ryzin 1998, Cooper 2002).

By introducing the dual variables $p_{t,x,u}$, the dual of (LP1) can be written as

\[(D1) \quad \max_p \sum_{t,x \in \mathcal{X}_t, u \in \mathcal{U}(x)} \left( \sum_j \lambda_{t,j} u_j f_j \right) p_{t,x,u} \]

\[\sum_{x \in \mathcal{X}_t, u \in \mathcal{U}(x)} x_i p_{t,x,u} = \begin{cases} \sum_{x \in \mathcal{X}_{t-1}, u \in \mathcal{U}(x)} (x_i - \sum_j \lambda_{t-1,j} a_{ij} u_j) p_{t-1,x,u}, & \text{if } t > 1, \\ 1, & \text{if } t = 1, \end{cases} \quad \forall i, t,\]

\[\sum_{x \in \mathcal{X}_t, u \in \mathcal{U}(x)} p_{t,x,u} = \begin{cases} \sum_{x \in \mathcal{X}_{t-1}, u \in \mathcal{U}(x)} p_{t-1,x,u}, & \text{if } t = 2, \ldots, \tau, \\ 1, & \text{if } t = 1, \end{cases} \quad \forall t.\]

Note that constraint (3) can be reduced to:

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

Adelman (2007) introduces a column generation approach to solving (D1). Given an optimal primal-dual solution $(\theta^*, V^*, p^*)$, a dynamic bid-price policy is constructed where a class-$j$ customer is accepted in period $t$ if $x \geq A^j$ (i.e., there are enough resources to satisfy demand for product $j$) and $f_j \geq \sum_i a_{ij} V_{t+1,i}^*$. His numerical study demonstrates that the dynamic bid-price policy often beats the widely used static bid-price policy (Talluri and van Ryzin 1998).

### 2.2. Separable Piecewise Linear Approximation

The affine approximation (2) forces linearity on resource levels which clearly does not hold in general for network RM problems, and motivated research on stronger functional approximation forms. Notably, several authors, including Farias and Van Roy (2007) and Meissner and Strauss (2012), consider the following separable piecewise linear approximation for (LP):

\[v_t(x) \approx \theta_t + \sum_i x_i \sum_{k=1}^k V_{t,i,k}, \quad \forall t, x,\]
where \( \theta_{\tau+1} = 0 \) and \( V_{\tau+1,i,k} = 0 \) for all \( i, k = 1, \ldots, c_i \).

The approximation (4) has been a popular approximation architecture in the approximate dynamic programming literature under various contexts (Bertsekas and Tsitsiklis 1996, Powell 2007). Since (4) does not require linearity, it is a more flexible approximation architecture compared to the affine approximation (2), and often leads to strong heuristic policies. In the network RM context, both Farias and Van Roy (2007) and Meissner and Strauss (2012) show that heuristic policies from the approximation give strong revenue performance. Note that (4) does not immediately imply concavity (i.e., \( V_{t,i,k} \) is decreasing in \( k \)). Nevertheless, concavity was assumed in much of the existing work, including Farias and Van Roy (2007) and Meissner and Strauss (2012). The recent work of Kunnumkal and Talluri (2011) shows that it is without loss of optimality to assume concavity in the sense that there always exists an optimal solution under (4) that satisfies this concavity constraint. We also assume concavity in the rest of the paper.

Plugging (4) into (LP) and rearranging terms leads to the following linear program:

\[
(LP-SC) \quad z_{sc} = \min_{\theta, V} \sum_{t=1}^{\tau} \sum_{i,k=1}^{c_i} V_{t,i,k} \theta_t - \theta_{t+1} + \sum_{i,k=1}^{c_i} (V_{t,i,k} - V_{t+1,i,k}) + \sum_j \lambda_{t,j} a_{ij} u_j V_{t+1,i,x_i} \geq \sum_j \lambda_{t,j} f_j u_j, \quad \forall t, x \in \mathcal{X}_t, u \in \mathcal{U}(x).
\]

The dual of (LP-SC) is given by

\[
(D-SC) \quad \max_p \sum_{t \in \mathcal{X}_t, u \in \mathcal{U}(x)} \left( \sum_j \lambda_{t,j} u_j f_j \right) p_{t,x,u} \\
= \begin{cases} 
\sum_{x \in \mathcal{X}_t, u \in \mathcal{U}(x)} p_{t-1,x,u} - \sum_{x \in \mathcal{X}_{t-1}, u \in \mathcal{U}(x)} \lambda_{t-1,j} a_{ij} u_j p_{t-1,x,u}, & \text{if } t > 1,
\end{cases}
\]

\[
\sum_{x \in \mathcal{X}_t, u \in \mathcal{U}(x)} p_{t,x,u} = \begin{cases} 
1, & \text{if } t = 1,
\sum_{x \in \mathcal{X}_{t-1}, u \in \mathcal{U}(x)} p_{t-1,x,u}, & \text{if } t = 2, \ldots, \tau,
\end{cases}
\]

\( p \geq 0. \)

2.3. Affine Approximation for Choice-based Network Revenue Management

Our discussion so far focuses on network RM with independent demand. Under the independent demand model, it is assumed that each customer requests a specific product but does not consider
any other products. Recently, there has been a considerable amount of research to generalize this demand model to discrete choice models, where each arriving customer chooses product \( j \in S \) with probability \( P_j(S) \). Here \( S \subseteq N \) is the set of open products (offer set) and \( N = \{1, \ldots, n\} \) denotes the set of all products. Within each period, it is assumed that there is a customer arrival with probability \( \lambda \). \(^1\) An early formulation of the choice-based network RM problem is given by Gallego et al. (2004). Our formulation closely follows Liu and van Ryzin (2008). The decision in each period is the offer set \( S \).

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

\[
v_t(x) = \max_{S \subseteq N(x)} \left\{ \lambda \sum_j P_j(S)[f_j + v_{t+1}(x - A^j)] + \left( 1 - \lambda \sum_j P_j(S) \right) v_{t+1}(x) \right\}, \quad \forall t, x, \tag{5}
\]

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


\[(\text{CLP1}) \quad \min_{\theta, V} \theta_1 + \sum_i V_{1,i}c_i \]

\[
\theta_t - \theta_{t+1} + \sum_i (V_t,i x_i - V_{t+1,i} (x_i - \lambda Q_i(S))) \geq \lambda R(S), \quad \forall t, x, S \subseteq N(x).
\]

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

\[(\text{CD1}) \quad \max_y \sum_{t,x \in A_t, S \subseteq N(x)} \lambda R(S)p_{t,x,S} \]

\[
\sum_{x \in A_t, S \subseteq N(x)} x_i p_{t,x,S} = \begin{cases} c_t, & \text{if } t = 1, \\ \sum_{x \in A_{t-1}, S \subseteq N(x)} (x_i - \lambda Q_i(S))p_{t-1,x,S}, & \text{if } t > 1, \end{cases} \quad \forall t, i
\]

\[
\sum_{x \in A_t, S \subseteq N(x)} p_{t,x,S} = 1, \quad \forall t,
\]

\[y \geq 0.\]

\(^1\) Our formulation here assumes stationary arrival rates and choice probabilities for notational simplicity. The model can be extended to non-homogeneous arrival rates and choice probabilities by dividing the booking horizon into segments with stationary parameters; see also discussion in Liu and van Ryzin (2008).
3. Reduction of Affine ALP under Independent Demand

In this section, we show that the Affine ALP (D1) can be reduced to an equivalent, but much smaller linear program. This reduction was recently considered by Tong and Topaloglu (2011). We provide an alternative proof technique by relating (D1) to a Dantzig-Wolfe reformulation of the reduced program. More importantly, since this new proof relies heavily on the general idea of Dantzig-Wolfe reformulation, we are able to apply it to (D-SC) and (CD1) as well, which we discuss in Sections 4 and 5, respectively.

The number of columns in (D1) grows exponentially in the number of resources \( m \) and the number of products \( n \), and therefore can be huge. However, the number of constraints are only linearly increasing in the number of periods \( \tau \) and the number of resources \( m \). A column generation procedure was proposed to solve (D1) in Adelman (2007). His algorithm starts with an initial feasible solution which corresponds to “do nothing” in each period; that is, \( p_{t,c,0} = 1 \) for all \( t \) and \( p_{t,x,u} = 0 \) otherwise. For a given primal solution \((\theta, V)\), the following column generation subproblem is solved for each period \( t \):

\[
\begin{align*}
\text{(S1)} \quad & \max_{x,u} \sum_j \lambda_{t,j} \left[ f_j - \sum_i a_{ij} V_{t+1,i} \right] u_j - \sum_i (V_{t,i} - V_{t+1,i}) x_i - \theta_t + \theta_{t+1} \\
& a_{ij} u_j \leq x_i, \quad \forall i, j, \\
& u_j \in \{0, 1\}, \quad \forall j, \\
& x_i \in \{0, \ldots, c_i\}, \quad \forall i.
\end{align*}
\]

Adelman shows that there exists an optimal solution to (LP1) that satisfies an intuitive time monotonicity properties; that is, \( V_{t,i} \geq V_{t+1,i} \) for all \( t, i \). In Proposition 1, we show that with this monotonicity property, the column generation subproblem (S1) can be reduced to a selection problem, and therefore has no integrality gap. In fact, it is well-known that selection problems can be expressed as minimum cut problems (Balinski 1970), which allow for very efficient solution procedures. A key observation enabling us to link (S1) to a selection problem is the realization that we can, without loss of optimality, restrict \( x_i \) to be binary for each resource \( i \). This implies that only columns with \( x_i \in \{0, 1\} \) for each \( i \) will be generated. Hence, there exists an optimal solution to (D1) where \( p_{t,x,u} > 0 \) only when \( x_i \in \{0, 1, c_i\} \) (recall that \( c_i \) is used in the initial solution to the column generation algorithm).

**Proposition 1.** Suppose \( V \) is decreasing in \( t \); i.e., \( V_{t,i} \geq V_{t+1,i} \) for all \( t, i \). Then (S1) is equivalent to

\[
\begin{align*}
\text{(S2)} \quad & \max_{x,u} \sum_j \lambda_{t,j} \left[ f_j - \sum_i a_{ij} V_{t+1,i} \right] u_j - \sum_i (V_{t,i} - V_{t+1,i}) x_i - \theta_t + \theta_{t+1} \\
& x_i \in \{0, \ldots, c_i\}, \quad \forall i.
\end{align*}
\]
\[ a_{ij} u_j \leq x_i, \quad \forall i, j, \]
\[ 0 \leq u_j \leq 1, \quad \forall j, \]
\[ 0 \leq x_i \leq 1, \quad \forall i. \]

Proof. Since \( V_{t,i} \geq V_{t+1,i} \), the coefficient of \( x_i \) is non-positive. On the other hand, \( a_{ij} u_j \leq 1 \) for all \( i, j \). Hence, it is without loss of optimality to require \( x_i \in \{0, 1\} \) in \((S1)\). To show that \((S1)\) with \( x_i \) binary has no integrality gap, note that we can take \( u_j = 0 \) in an optimal solution for any product \( j \) with \( f_j - \sum_i a_{ij} V_{t+1,i} \leq 0 \). Summarizing the above, \((S1)\) can be rewritten as

\begin{align*}
\text{(S1')} & \max_{x,u} \sum_{j: f_j - \sum_i a_{ij} V_{t+1,i} > 0} \lambda_{t,j} \left[ f_j - \sum_i a_{ij} V_{t+1,i} \right] u_j - \sum_t (V_{t,i} - V_{t+1,i}) x_i - \theta_t + \theta_{t+1} \\
& \quad u_j \leq x_i, \quad \forall i, j : a_{ij} = 1, \\
& \quad u_j \in \{0, 1\}, \quad \forall j, \\
& \quad x_i \in \{0, 1\}, \quad \forall i.
\end{align*}

The problem \((S1')\) is a selection problem and therefore has no integrality gap; see Hochbaum (2004). This completes the proof.

In their recent work, Tong and Topaloglu (2011) show that \((S1)\) has no integrality gap. Their proof is based on the observation that the dual of \((S1)\) is a minimum cost network flow problem. Their result implies that \((S1)\) can be solved as a minimum cost network flow problem. Their result is slightly weaker than ours since a minimum cut problem is a special case of a minimum cost network flow problem, and can usually be solved more efficiently.

**Proposition 2 (Tong and Topaloglu (2011)).** The program \((D1)\) can be reduced to the following equivalent linear program:

\begin{align*}
\text{(D2)} & \max_{r,q} \sum_{t,j} \lambda_{t,j} f_j q_{t,j} \\
& r_{t,i} = \begin{cases} c_i, & \text{if } t = 1, \\
& r_{t-1,i} - \sum_j a_{ij} q_{t-1,j}, & \text{if } t > 1, \\
\end{cases} \forall i, t, \\
& a_{ij} q_{t,j} \leq r_{t,i}, \forall t, i, j, \quad (6) \\
& q_{t,j} \leq 1, \forall t, j, \quad (7) \\
& q \geq 0, r \geq 0. \quad (8)
\end{align*}

In order to establish the equivalence between \((D1)\) and \((D2)\), we proceed in two steps. First, we show that \((D2)\) can be derived from \((D1)\) through a change of variables, and hence is a relaxation of \((D1)\). Second, we show that a feasible solution of \((D1)\) can be constructed from a feasible solution of \((D2)\) reaching the same objective value. Combining the arguments shows the equivalence.
Lemma 1. Any feasible solution to \((D1)\) yields a feasible solution to \((D2)\) with the same objective value.

Proof. We introduce the following change of variables:

\[
    r_{t,i} = \sum_{x \in X, u \in U(x)} x_i p_{t,x,u}, \quad \forall t, i,
\]

\[
    q_{t,j} = \sum_{x \in X, u \in U(x)} u_j p_{t,x,u}, \quad \forall t, j.
\]

The objective value and the first constraint in \((D1)\) can be rewritten as in \((D2)\). Since \(\sum_{x \in X, u \in U(x)} p_{t,x,u} = 1 \) and \(u \in U(x)\), \(q\) must satisfy (8). Furthermore, for each \(t, i, j\), we have \(a_{ij} q_j \leq x_i\). Constraint (7) follows. Summarizing the above leads to \((D2)\). Note that \(r_{t,i} \leq c_i\) for each \(i\) is implied by constraints (6) and (9).

By construction, any feasible solution \(p\) to \((D1)\) corresponds to a feasible solution \((q,r)\) to \((D2)\). This completes the proof.

Next, we show that any feasible solution \((q,r)\) to \((D2)\) corresponds to a feasible solution \(p\) to \((D1)\), which implies that \((D1)\) are \((D2)\) are equivalent. To do so, we will use the following lemma.

For each \(t\), let \(\Omega_t\) be the polyhedron defined by constraints (7)–(9) with the given \(t\), and an additional constraint

\[
    r_{t,i} \leq c_i, \quad \forall i. \tag{10}
\]

Note that in \((D2)\), (10) holds for each \(t\) from (6) and (9). The following lemma shows that \(\Omega_t\) is a bounded polyhedron with integer extreme points.

Lemma 2. For each \(t\), \(\Omega_t\) is a bounded polyhedron with integer extreme points.

Proof. We first note that the constraint matrix \(M_t\) for the constraint \(a_{ij} q_{t,j} \leq r_{t,i}\) \(\forall i, j\) is totally unimodular since \(M_t\) is a difference system, and therefore corresponds to the dual of a network problem. By Proposition 3.1 of Wolsey (1998), \((M_t^T, I)^T\) is also totally unimodular, where \(I\) is an \(m + n\) dimensional identity matrix. This completes the proof.

Then, the desired result follows from the following lemma.

Lemma 3. Suppose \((q,r)\) is a feasible solution to \((D2)\). Then, there exists a feasible solution \(p\) to \((D1)\) with the same objective value.

Proof. We establish this result using the Dantzig-Wolfe reformulation of \((D2)\). Let \(\{(q_t^{(k)}, r_t^{(k)}) : k = 1, \ldots, K_t\}\) be the extreme points of polyhedron \(\Omega_t\), for each period \(t\). Then, any feasible solution \((q,r)\) to \((D2)\) can be expressed as

\[
    (q_t, r_t) = \sum_{k=1}^{K_t} (q_t^{(k)}, r_t^{(k)}) \mu_{t,k},
\]
where $\sum_{k=1}^{K_t} \mu_{t,k} = 1$ and $\mu \geq 0$.

The Dantzig-Wolfe reformulation of (D2) is given by

$$\text{(DW)} \quad \max_{\mu} \sum_{t,k=1,\ldots,K_t} \left( \sum_j \lambda_{t,j} q_{t,j} f_j \right) \mu_{t,k} \quad \text{if } t = 1, \quad \forall i, t,$$

$$\sum_{k=1}^{K_t} \mu_{t,k} = 1, \quad \forall t,$$

$$\mu \geq 0.$$

By Lemma 2, all extreme points are integral. Furthermore, from the defining constraints of $\Omega_t$, the extreme points correspond to feasible state-action pairs. Thus, it is straightforward to construct a feasible solution to (D1) by letting $p_{t,r}(k) q_{t} = \mu_{t,k}$ for each $t$ and $k = 1, \ldots, K_t$, and letting $p_{t,x,u} = 0$ otherwise.

It is easy to check that this construction generates a solution that will satisfy $\sum_{t,j} \lambda_{t,j} q_{t,j} f_j = \sum_{t,x \in X_t, u \in U(t)} \left( \sum_j \lambda_{t,j} u_j f_j \right) p_{t,x,u}$. This completes the proof. \hfill \blacksquare

We note that our construction can be seen as a way to recover the “original” formulation corresponding to the Dantzig-Wolfe reformulation expressed in (D1). Observe, however, that the connection is not direct because not all state-action pairs are extreme points in the polyhedra $\Omega_t$. Our proof relies on the fact that extreme points of $\Omega_t$ are integral, which allows us to relate them to feasible state-action pairs and thus construct feasible solutions to (D1) from a solution in the Dantzig-Wolfe reformulation. Since any linear program has an equivalent Dantzig-Wolfe reformulation, this proof technique has the potential to be extended to other settings.

4. Reduction of Separable Piecewise Linear Approximation under Independent Demand

In this section, we consider the reduction of (D-SC), which is the ALP resulting from separable piecewise linear approximation for network RM under independent demand. Section 4.1 reviews the column generation approach for (D-SC), which lays the foundation for the rest of the section. Section 4.2 shows that (D-SC) can be reduced to an equivalent, but much smaller linear program. Section 4.3 offers an intuitive interpretation of the reduced program. Section 4.4 shows that the reduced program is equivalent to the Lagrangian relaxation considered in Topaloglu (2009).

4.1. A Column Generation Approach to (D-SC)

Our ultimate goal is to reduce the formulation (D-SC). To achieve this, we begin by determining how state-action pairs in (D-SC) can be generated using column generation.
For each fixed \( t > 1 \), the column generation subproblem is given by

\[
(CG) \quad \max_{x \in X, \ u \in U} \sum_j \lambda_{t,j} \left[ f_j - \sum_i a_{ij} V_{t+1,i},x_i \right] u_j - \sum_i \sum_{k=1}^{c_i} (V_{t,i,k} - V_{t+1,i,k}) + \theta_{t+1} - \theta_t.
\]

Note that objective function in \((CG)\) is not linear in \( x \), but can be linearized by introducing additional decision variables. To that end, let

\[
y_{t,i,k} = \begin{cases} 
1, & \text{if } x_{t,i} \geq k, \\
0, & \text{otherwise}, 
\end{cases} \quad \forall i, k = 1, \ldots, c_i. \tag{11}
\]

Observe that the definition above requires \( y_{t,i,k} \geq y_{t,i,k+1} \) for all \( i, k \). Here, we assume \( y_{t,i,c_i+1} = 0 \) for all \( t, i \). We also note that \( x_{t,i} = k \) if and only if \( y_{t,i,k} - y_{t,i,k+1} = 1 \) for \( i, k = 1, \ldots, c_i \).

With this definition, we can reformulate \((CG)\) as

\[
(CG1) \quad \max_{y, u} \sum_j \lambda_{t,j} \left[ f_j - \sum_i a_{ij} V_{t+1,i},x_i \right] u_j - \sum_i \sum_{k=1}^{c_i} (V_{t,i,k} - V_{t+1,i,k}) y_{t,i,k} + \theta_{t+1} - \theta_t
\]

\[
u_j \leq y_{t,i,1}, \quad \forall i, j : a_{ij} = 1,
\]

\[
y_{t,i,k} \geq y_{t,i,k+1}, \quad \forall i, k = 1, \ldots, c_i,
\]

\[
y, u \text{ binary.}
\]

The first term in the objective function above is still nonlinear because it involves the multiplication between \( y \) and \( u \). To linearize the first term in the objective function, we first observe that it can be rewritten as

\[
\sum_j \lambda_{t,j} \left[ f_j - \sum_i a_{ij} V_{t+1,i},x_i \right] u_j - \sum_i \sum_{k=1}^{c_i} (V_{t,i,k} - V_{t+1,i,k}) y_{t,i,k} + \theta_{t+1} - \theta_t
\]

\[
u_j \leq y_{t,i,1}, \quad \forall i, j : a_{ij} = 1
\]

\[
y_{t,i,k} \geq y_{t,i,k+1}, \quad \forall i, k = 1, \ldots, c_i
\]

\[
y, u \text{ binary.}
\]

Using (12), this term can be reduced to

\[
\sum_j \lambda_{t,j} \left[ f_j - \sum_i a_{ij} V_{t+1,i},x_i \right] u_j + \sum_i \sum_{k=2}^{c_i} a_{ij} \lambda_{t,j} (V_{t+1,i,k-1} - V_{t+1,i,k}) y_{t,i,k} u_j.
\]

Now, we introduce a new decision variable

\[
z_{t,i,j,k} = \begin{cases} 
1, & \text{if } y_{t,i,k} = 1 \text{ and } u_j = 1, \\
0, & \text{otherwise.}
\end{cases} \quad \forall i, j, k = 2, \ldots, c_i, a_{ij} = 1. \tag{13}
\]

Observe that this definition requires

\[
z_{t,i,j,k} \leq y_{t,i,k}, z_{t,i,j,k} \leq u_j, y_{t,i,k} + u_j \leq 1 + z_{t,i,j,k}, \quad \forall i, j, k = 2, \ldots, c_i.
\]

It is easy to see that when \( V_{t,i,k} \) is monotonically decreasing in \( k \), the constraint \( y_{t,i,k} + u_j \leq 1 + z_{t,i,j,k} \) will become redundant. To see this, suppose we have an optimal solution where \( y_{t,i,k} \) and
both equal 1, but that $z_{t,i,j,k} = 0$. Because the $\lambda_{t,j}(V_{t+1,i,k-1} - V_{t+1,i,k}) \geq 0$, setting $z_{t,i,j,k} = 1$ will never worsen the solution value.

Summarizing the above, the column generation subproblem can be rewritten as the following binary integer problem

$$
\text{(CG2) } \max_{y,u,z} \sum_{t,j} \lambda_{t,j} \left[ f_j - \sum_i a_{ij} V_{t+1,i,1} \right] u_j + \sum_{t,j} \sum_{k=2}^{c_i} a_{ij} \lambda_{t,j} (V_{t+1,i,k-1} - V_{t+1,i,k}) z_{t,i,j,k} - \sum_{i} \sum_{k=1}^{c_i} (V_{t,i,k} - V_{t+1,i,k}) y_{t,i,k} + \theta_{t+1} - \theta_t
$$

$$
u_j \leq y_{t,i,1}, \quad \forall i, j : a_{ij} = 1,$n$$

$$y_{t,i,k+1} \leq y_{t,i,k}, \quad \forall i, k = 1, \ldots, c_i,$n$$

$$z_{t,i,j,k} \leq u_j, \quad \forall j, i : a_{ij} = 1,$n$$

$$z_{t,i,j,k} \leq y_{t,i,k}, \quad \forall i, k = 2, \ldots, c_i,$n$$

$y, u, z$ binary.

Note that the constraint matrix for (CG2) is a difference system. Therefore, the corresponding constraint polyhedron has integer extreme points. Hence the binary constraints can be relaxed.

4.2. Reduction

The column generation approach to (D-SC), although theoretically possible, is impractical for even small problems. For network RM problems with customer choice, Meissner and Strauss (2012) show that the column generation algorithm can take hours even for relatively small problem instances. Here, we consider an equivalent reduction that dramatically reduce the size of the problem. The reduced program can be viewed as an aggregated version of (D-SC). Nevertheless, we are able to establish that the reduced program produce the same objective value as (D-SC). In this sense, the reduced program is equivalent to (D-SC).

With slight abuse of notation, we introduce the following change of variables

$$y_{t,i,k} = \sum_{x : x_i \geq k} p_{t,x,u}, \quad \forall t, i, k \in 1, \ldots, c_i,$n$$

$$q_{t,j} = \sum_{x : x_j \geq k} u_j p_{t,x,u}, \quad \forall t, j,$n$$

$$z_{t,i,j,k} = \sum_{x : x_i \geq k} u_j p_{t,x,u}, \quad \forall t, i, j, k \in 1, \ldots, c_i : a_{ij} = 1.$n$$

We point out that $y$ and $z$ defined above have slightly different interpretation as the binary variables $y$ and $z$ in the column generation subproblem, even though they are intricately related. In
particular, \(y_{t,i,k}\) can be interpreted as the probability of having at least \(k\) units of remaining capacity on leg \(i\) at the beginning of period \(t\) (observe that the expected remaining capacity on leg \(i\) will equal \(\sum_{k=1}^{c_i} y_{t,i,k}\)). Similarly, we interpret \(q_{t,j}\) as the probability with which we open itinerary \(j\) for sale during time period \(t\), and \(h_{t,i,j,k}\) as the probability with which we open itinerary \(j\) for sale during time period \(t\) when \(k\) units of capacity are available on leg \(i\).

With this change of variables, we can reformulate the dual program \((\text{D-SC})\) as

\[
\text{(D-R)} \quad \max_{q,y,z} \sum_{t,j} \lambda_{t,j} f_j q_{t,j}
\]

\[
y_{t,i,k} = \begin{cases} 
1, & \text{if } t = 1, \\
\sum_j \lambda_{t-1,j} a_{ij} (z_{t-1,i,j,k} - z_{t-1,i,j,k+1}), & \text{if } t > 1
\end{cases}, \quad \forall t, i, k = 1, \ldots, c_i,
\]

\[
y_{t,i,k+1} \leq y_{t,i,k}, \quad \forall t > 1, i, k = 1, \ldots, c_i, \quad (14)
\]

\[
z_{t,i,j,k+1} \leq z_{t,i,j,k}, \quad \forall t > 1, i, j, k = 1, \ldots, c_i : a_{ij} = 1, \quad (15)
\]

\[
z_{t,i,j,k} \leq y_{t,i,k}, \quad \forall t, i, j : a_{ij} = 1, \quad (16)
\]

\[
z_{t,i,j,1} = q_{t,j}, \quad \forall t, i, j : a_{ij} = 1, \quad (17)
\]

\[
q, y, z \geq 0, \leq 1. \quad (18)
\]

In the above, the objective function and the first constraint follows by plugging in the definitions of \(y, q, z\), while the constraints \((14)-(18)\) can be derived from the definitions of \(y, q, z\) and the nonnegativity of \(p\). We also use the convention that \(y_{t,i,c_i+1} = 0\) and \(z_{t,i,j,c_i+1} = 0\) for all \(t, i, j, k\).

By construction, any feasible solution \(p\) to \((\text{D-SC})\) yields a feasible solution \((q, y, z)\) to \((\text{D-R})\). As in the affine case, it is possible to show that the reverse statement also holds, which implies that \((\text{D-SC})\) and \((\text{D-R})\) are equivalent. To do so, we will use the following lemma.

**Lemma 4.** For each \(t\), let \(R_t\) be the polyhedron defined by constraints \((14)-(18)\). Then \(R_t\) is a bounded polyhedron with integer extreme points.

**Proof.** This immediately follows, since defining constraints of \(R_t\) correspond to a closure problem, which is known to be integral as its dual corresponds to a network (Ahuja et al. 1993).

The desired result follows from the following proposition.

**Proposition 3.** There exists an optimal solution \((q, y, z)\) to \((\text{D-R})\) with a corresponding feasible solution \(p\) to \((\text{D-SC})\) with the same objective value.

**Proof.** We establish this result using the Dantzig-Wolfe reformulation of \((\text{D-R})\). For each \(t\), let \(\{(q^\omega_t, y^\omega_t, z^\omega_t) : \omega = 1, \ldots, \Omega_t\}\) be the set of extreme points of polyhedron \(R_t\).
For each \( t \), let \( \mu_t \) be a weight vector such that \( \sum_{\omega=1}^{\Omega_t} \mu_t^\omega = 1 \). Thus, any feasible solution \((q, y, z)\) to (D-R) can be expressed as \((q_t^\omega, y_t^\omega, z_t^\omega)\mu_t^\omega \). Then, the Dantzig-Wolfe reformulation is given by

\[
\text{(D-R2)} \quad \max \mu \sum_{t} \sum_{\omega=1}^{\Omega_t} \left( \sum_{j} \lambda_{t,j} q_{t,j}^\omega f_j \right) \mu_t^\omega \\
\sum_{\omega=1}^{\Omega_t} y_{t,i,k}^\omega \mu_t^\omega = \begin{cases} 
1, & \text{if } t = 1, \\
\sum_{\omega=1}^{\Omega_{t-1}} y_{t-1,i,k}^\omega \sum_{\omega=1}^{\Omega_t} \sum_{l} \lambda_{t-1,j} a_{ij} (z_{t-1,i,j,k}^\omega - z_{t-1,i,j,k+1}^\omega) \mu_{t-1}^\omega, & \text{if } t > 1,
\end{cases} \quad \forall t, i, k \in X_{t,i},
\]

\[
\sum_{\omega=1}^{\Omega_t} \mu_t^\omega = 1, \quad \forall t, \\
\mu \geq 0.
\]

Next we show that a feasible solution to (D-SC) can be constructed from a solution \( \mu \) of (D-R2). For each \( t, x \in X_t, u \in U(x) \), let

\[
p_{t,x,u} = \sum_{\omega=1}^{\Omega_t} \mu_t^\omega.
\]

Now, we show that by defining \( p \) above, we recover the objective function in (D-SC). To see this, note that

\[
\sum_{t \in T, u \in U(x)} \left( \sum_{j} \lambda_{t,j} u_j f_j \right) p_{t,x,u} = \sum_{t \in T, u \in U(x)} \left( \sum_{j} \lambda_{t,j} u_j f_j \right) \sum_{\omega=1}^{\Omega_t} \mu_t^\omega \\
= \sum_{t \in T, u \in U(x)} \left( \sum_{\omega=1}^{\Omega_t} \sum_{l} \lambda_{t,j} q_{t,j}^\omega f_j \mu_t^\omega \right) \\
= \sum_{t \in T} \sum_{\omega=1}^{\Omega_t} \left( \sum_{j} \lambda_{t,j} q_{t,j}^\omega f_j \mu_t^\omega \right).
\]

It remains to show that \( p \) defined in (19) satisfies the constraints in (D-SC). It is easy to show that the second constraint in (D-SC) is satisfied. So we focus on the first constraint in (D-SC). For fixed \( t > 1, i, k = 1, \ldots, c_i - 1 \), we have,

\[
\sum_{x \in X_t, u \in U(x)} p_{t,x,u} = \sum_{x \in X_t, u \in U(x)} \sum_{\omega=1}^{\Omega_t} \sum_{\omega=1}^{\Omega_t} \mu_t^\omega \\
= \sum_{x \in X_t, u \in U(x)} \sum_{\omega=1}^{\Omega_t} \sum_{\omega=1}^{\Omega_t} \mu_t^\omega.
\]
\[
= \sum_{\omega=1, \ldots, \Omega_t: \sum_{h=1}^{c_l} y_{t,i,h}^\omega \geq k} \mu_t^\omega
\]
\[
= \sum_{\omega=1, \ldots, \Omega_t: \sum_{h=1}^{c_l} y_{t,i,k}^\omega \geq 1} \mu_t^\omega
\]
\[
= \sum_{\omega=1}^{\Omega_t} y_{t,i,k}^\omega \mu_t^\omega.
\]

In the second to last step above, we used the monotonicity of \(y_{t,i,h}^\omega\) with respect to \(h\) to establish the equivalence between the two conditions \(\sum_{h=1}^{c_l} y_{t,i,h}^\omega \geq k\) and \(y_{t,i,k}^\omega = 1\).

For fixed \(t, i, j, k\), we also have
\[
\sum_{x \in X_t \cup U(t(x))} u_{j} p_{t,x,u} = \sum_{x \in X_t \cup U(t(x))} u_{j} \sum_{\omega=1, \ldots, \Omega_t: \sum_{h=1}^{c_l} y_{t,i,h}^\omega = \sum_{t} y_{t,i,h}^\omega} \mu_t^\omega
\]
\[
= \sum_{\omega=1, \ldots, \Omega_t: \sum_{h=1}^{c_l} y_{t,i,h}^\omega = \sum_{t} y_{t,i,h}^\omega} \mu_t^\omega - \sum_{\omega=1, \ldots, \Omega_t: \sum_{h=1}^{c_l} y_{t,i,h}^\omega \geq k+1} \mu_t^\omega
\]
\[
= \sum_{\omega=1, \ldots, \Omega_t: \sum_{h=1}^{c_l} y_{t,i,h}^\omega = \sum_{t} y_{t,i,h}^\omega} \mu_t^\omega - \sum_{\omega=1, \ldots, \Omega_t: \sum_{h=1}^{c_l} y_{t,i,h}^\omega \geq k+1} \mu_t^\omega
\]
\[
= \sum_{\omega=1}^{\Omega_t} (z_{t,i,j,k}^\omega - z_{t,i,j,k+1}^\omega) \mu_t^\omega.
\]

In the second to last step above, we used the result of Lemma 5. Combining the two identities above shows that \(p\) defined in (19) satisfies the first constraint in (D-SC). This completes the proof.

**Lemma 5.** Suppose \(V_{t,i,k}\) is monotone decreasing in \(k\) for each \(t\) and \(i\). Then it is without loss of optimality to restrict (D-R2) to columns with \(z_{t,i,j,k} = 1\) when \(y_{t,i,k} = 1\) and \(q_{t,j} = 1\) for all \(t, i, j, k\).

Proof. For each \(t\), the column generation subproblem for (D-R2) is given by

\[
(\text{CG-R2}) \max_{y_t, q_t, z_t} \sum_{j} \lambda_{t,j} \left[ f_j - \sum_{i} a_{ij} V_{t+1,1,i} \right] q_{t,j} + \sum_{j} c_l \lambda_{t,j} (V_{t+1,1,k-1} - V_{t+1,1,k}) z_{t,i,j,k}
\]
\[- \sum_{i} (V_{t,i,k} - V_{t+1,1,k}) y_{t,i,k} + \theta_{t+1} - \theta_t
\]
subject to (14)–(18).

Fix $t, i, j, k$. Suppose $y_{t, i, k} = 1$ and $q_{t, j} = 1$. From the monotonicity constraint of $y$, $y_{t, i, k'} = 1$ for all $k' \leq k$. Therefore, we can set $z_{t, i, j, k'} = 1$ for all $k' \leq k$ without violating any constraint. Furthermore, since $V_{t, i, k}$ is monotone decreasing in $k$, the coefficient of $z_{t, i, j, k}$ is nonnegative in the objective function of (CG-R2). Hence, optimality will not be violated either. This completes the proof.

4.3. Interpretation of the Reduced Program

We first consider a slight reformulation of (D-R) by introducing the variable $h_{t, i, j, k} := z_{t, i, j, k} - z_{t, i, j, k+1}$, such that $z_{t, i, j, k} = \sum_{l=1}^{c_i} h_{t, i, j, k}$. Then, (D-R) can be rewritten as

\[
\text{(D-R1)} \quad \mathcal{V}_1(c) = \max_{q, y, h} \sum_{t,j} \lambda_{t,j} f_q q_{t,j}
\]

\[
y_{t, i, k} = \begin{cases} 
1, & \text{if } t = 1, \\
y_{t-1, i, k} - \sum_j a_{ij} h_{t-1, i, j, k}, & \text{if } t > 1,
\end{cases} \quad \forall t, i, k = 1, \ldots, c_i,
\]

\[
y_{t, i, k+1} \leq y_{t, i, k}, \quad \forall t, i, k = 1, \ldots, c_i,
\]

\[
q_{t,j} = \sum_{k=1}^{c_i} h_{t, i, j, k}, \quad \forall t, i, j : a_{ij} = 1,
\]

\[
\sum_{l=k}^{c_i} h_{t, i, j, l} \leq y_{t, i, k}, \quad \forall t, i, j, k = 1, \ldots, c_i : a_{ij} = 1,
\]

\[
0 \leq q \leq 1, h \geq 0.
\]

Note that the simple upper bounds on the variables are redundant in principle, even thought we retain the bound on $q$. Furthermore, the variable $y$ can be considered a free variable given the assumption that $y_{t, i, c_i+1} = 0$ for all $t, i$.

The program (D-R1) admits an interesting probabilistic interpretation. Let $(X_t, U_t)$ denotes the state-action pair, whose distribution is given by $p_{t, x, u}$. Then, by definition we have

\[
y_{t, i, k} = \sum_{x \in X_t, u \in U(x): x_i \geq k} p_{t, x, u} = P(X_{t,i} \geq k), \quad \forall t, i, k = 1, \ldots, c_i,
\]

\[
q_{t,j} = \sum_{x \in X_t, u \in U(x)} u_{j} p_{t, x, u} = E U_{t,j}, \quad \forall t, j,
\]

\[
z_{t, i, j, k} = \sum_{x \in X_t, u \in U(x): x_i \geq k} u_{j} p_{t, x, u}, \\
= E[U_{t,j} | \{X_{t,i} \geq k\}] P(X_{t,i} \geq k) \\
= P(X_{t,i} \geq k, U_{t,j} = 1), \quad \forall t, i, j, k = 1, \ldots, c_i : a_{ij} = 1,
\]
\[ h_{t,i,j,k} = \sum_{x \in X_t, u \in U(t), x_t \geq k} u_j P(x_t, x, u), \]
\[ = \mathbb{E}[U_{t,j} | X_{t,i} = k] P(X_{t,i} = k) \]
\[ = P(X_{t,i} = k, U_{t,j} = 1), \quad \forall t, i, j, k \in 1, \ldots, c_i : a_{ij} = 1. \] (28)

Therefore, the variable \( y_{t,i,k} \) can be interpreted as the probability that the remaining capacity on resource \( i \) in period \( t \) is at least \( k \). The variable \( q_{t,j} \) is the probability that product \( j \) in period \( t \). The variable \( z_{t,i,j,k} \) is the probability that the remaining capacity on resource \( i \) is at least \( k \) and product \( j \) is open in period \( t \). Finally, The variable \( z_{t,i,j,k} \) is the probability that the remaining capacity on resource \( i \) is exactly \( k \) and product \( j \) is open in period \( t \).

Using equations (25)—(28), the objective function of (D-R1) can be written as

\[ \sum_{t,j} \lambda_{t,j} f_j \mathbb{E}U_{t,j}. \]

Note that the term \( \lambda_{t,j} f_j \mathbb{E}U_{t,j} \) can be interpreted as expected revenue from class \( j \) in period \( t \). Hence, the objective is the total expected revenue from all classes. The constraints in (D-R1) can be written as

\[ P(X_{t,i} \geq k) = \begin{cases} P(X_{t-1,i} \geq k) & \text{if } t = 1, \\ P(X_{t,i} \geq k + 1) & \text{if } t > 1, \end{cases} \quad \forall t, i, k \in 1, \ldots, c_i, \] (29)

\[ P(X_{t,i} \geq k + 1) \leq P(X_{t,i} \geq k), \quad \forall t, i, k \in 1, \ldots, c_i, \] (30)

\[ \mathbb{E}U_{t,j} = \sum_{k=1}^{c_i} P(X_{t,i} = k, U_{t,j} = 1), \quad \forall t, i, j : a_{ij} = 1, \] (31)

\[ \sum_{k=1}^{c_i} P(X_{t,i} = k, U_{t,j} = 1) \leq P(X_{t,i} \geq k), \quad \forall t, i, j, k \in 1, \ldots, c_i : a_{ij} = 1. \] (32)

The constraints above also have intuitive probabilistic interpretations. Constraint (29) 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 \( k = 1, \ldots, c_i \), 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 (30) requires that the probability \( X_{t,i} \geq k + 1 \) be smaller than the probability \( X_{t,i} \geq k \). To understand (31), we note the constraint can be rewritten as

\[ \mathbb{E}U_{t,j} = \sum_{k=1}^{c_i} P(U_{t,j} = 1 | X_{t,i} = k) P(X_{t,i} = k) \]
\[ = \sum_{k=1}^{c_i} [1 \times P(U_{t,j} = 1 | X_{t,i} = k) + 0 \times P(U_{t,j} = 0 | X_{t,i} = k)] P(X_{t,i} = k), \quad \forall t, i, j : a_{ij} = 1. \]
Therefore, this constraint merely expresses the expectation of \( U_{t,j} \) conditioning on \( X_{t,i} \). The constraint (32) can be rewritten as

\[
P(X_{t,i} \geq k, U_{t,j} = 1) \leq P(X_{t,i} \geq k), \quad \forall t, i, j, k = 1, \ldots, c : a_{ij} = 1.
\]

Hence, this constraint requires that the probability that \( X_{t,i} \geq k \) and \( U_{t,j} = 1 \) is less than the probability of \( X_{t,i} \geq k \).

The discussion above shows that the program \( \text{(D-R1)} \) enforces a set of necessary conditions on the stochastic process \( \{ (X_t, U_t) \} \). It is not hard to see that the conditions imposed are not sufficient to track the full dynamics of \( \{ (X_t, U_t) \} \), because it focuses on the marginal distributions on \( X_t \). Therefore, \( \text{(D-R1)} \) is a relaxation to the original dynamic programming formulation.

4.4. An Alternative Proof of the Equivalence between Separable Piecewise Linear Approximation and Lagrangian Relaxation

We first sketch the Lagrangian relaxation approach for network RM introduced in Topaloglu (2009). The dynamic programming formulation (1) can be equivalently stated as

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

s.t

\[
a_{ij}s_{ij} \leq x_i, \quad \forall i, j,
\]

\[
s_{ij} - u_j = 0, \quad \forall i, j,
\]

\[
s_i \in \{0,1\}^n, \quad \forall i,
\]

\[
u \in \{0,1\}^n.
\]

In the formulation above, \( s_i \in \{0,1\}^n \) denotes the acceptance decision on resource \( i \). The constraint (33) requires that \( s_{ij} = s_{i'j} = u_j \) for any pair of resources \( i, i' \). Associating Lagrangian multiplier \( \mu = \{\mu_{tij}\}_{\forall t,i,j} \) to constraint (33), we have

\[
v_t^{\mu}(x) = \max_{u,s} \left\{ \sum_j \lambda_{t,j} \left[ \left( f_j - \sum_i \mu_{tij} \right)^+ u_j + v_{t+1}^{\mu}(x - A^j u_j) + \sum_i \mu_{tij} s_{ij} \right] \right\}
\]

\[
+ \left( 1 - \sum_j \lambda_{t,j} \right) v_{t+1}^{\mu}(x), \quad \forall t, x,
\]

s.t

\[
a_{ij}s_{ij} \leq x_i, \quad \forall i, j,
\]

\[
s_i \in \{0,1\}^n, \quad \forall i,
\]

\[
u \in \{0,1\}^n.
\]
Following Proposition 1 in Topaloglu (2009), we have

\[ v^\mu_t(x) = \sum_{t'=t}^\tau \sum_j \lambda_{t',j} (f_j - \sum_i \mu_{t'ij})^+ + \sum_i v^\mu_{t,i}(x_i), \]

where

\[ v^\mu_{t,i}(x_i) = \max_{s_i \in S_i(x_i)} \sum_j \lambda_{t,j} [\mu_{tij} s_{ij} + v^\mu_{t',i}(x_i - a_{ij} s_{ij})] + \left(1 - \sum_j \lambda_{t,j}\right) v^\mu_{t+1,i}(x_i). \]

The value \( v^\mu_t(x) \) gives an upper bound to \( v_t(x) \); see Topaloglu (2009), Proposition 2. Recently, Talluri (2008) shows that there exists an optimal set of Lagrangian multiplier \( \mu \geq 0 \) such that \( f_j = \sum_i \mu_{tij} \) for all \( t \) and \( j \). It follows that we can restrict attention to \( v^\mu_t(x) \) such that \( v^\mu_t(x) = \sum_i v^\mu_{t,i}(x_i) \).

Therefore, the best such upper bound can be constructed by minimizing over the Lagrangian multiplier \( \mu \), i.e.,

\[
\min_{\mu \geq 0} \sum_i v^\mu_{t,i}(c_i). \tag{36}
\]

Invoking the linear programming formulation for the dynamic program associated with \( v^\mu_{t,i}(x_i) \) for each \( i \) leads to

\[
(LP-LG) \min_{\mu, \{v^\mu_{t,i}(x_i)\} \forall t,i} \sum_i v^\mu_{t,i}(c_i)
\]

\[
v^\mu_{t,i}(x_i) \geq \sum_j \lambda_{t,j} [\mu_{tij} s_{ij} + v^\mu_{t',i}(x_i - a_{ij} s_{ij})] + \left(1 - \sum_j \lambda_{t,j}\right) v^\mu_{t+1,i}(x_i), \quad \forall t,i,x_i, s_i \in S_i(x_i),
\]

\[
\sum_i \mu_{tij} = f_j, \quad \forall t,j,
\]

\[
\mu \geq 0.
\]

Using the change of variable

\[
v^\mu_{t,i}(x_i) = \sum_{k=0}^{x_i} V_{t,i,k}, \quad \forall t,i,x_i,
\]

\( (LP-LG) \) can be rewritten as

\[
(LP-LG1) \min_{\mu, \{V_{t,i,k}\} \forall t,i} \sum_i \sum_{k=0}^{x_i} V_{t,i,k}
\]

\[
\sum_{k=0}^{x_i} V_{t,i,k} \geq \sum_j \lambda_{t,j} [\mu_{tij} - a_{ij} V_{t+1,i,x_i}] + \sum_{k=0}^{x_i} V_{t+1,i,k}, \quad \forall t,i,x_i, s_i \in S_i(x_i),
\]

\[
\sum_i \mu_{tij} = f_j, \quad \forall t,j,
\]

\[
\mu \geq 0.
\]

It should be pointed out that the constraint \( \sum_i \mu_{tij} = f_j \) can be equivalently stated as \( \lambda_{t,j} \sum_i \mu_{tij} = \lambda_{t,j} f_j \) since \( \mu_{tij} \) only matters when \( \lambda_{t,j} \neq 0 \) in the first constraint. The dual of \( (LP-LG1) \) is given by

\[
(D-LG) \max_p \sum_{t,j} \lambda_{t,j} f_j q_{t,j}
\]
\[ q_{t,j} - \sum_{x_i} \sum_{s_i \in S_i(x_i)} s_{ij} p_{t,i,x_i,s_i} \leq 0, \quad \forall t, i, j, \]
\[ \sum_{x_i \geq k, s_i \in S_i(x_i)} p_{t,i,x_i,s_i} \]
\[ = \begin{cases} \sum_{x_i \geq k, s_i \in S_i(x_i)} p_{t-1,i,x_i,s_i} - \sum_{x_i = k, s_i \in S_i(x_i)} \sum_{j} \lambda_{t-1,j} a_{ij} s_{ij} p_{t-1,i,x_i,s_i} & \text{if } t = 1, \\ \forall t, i, k \in X_{t,i}, \end{cases} \]
\[ p \geq 0. \]

Observe that (D-LG) is bounded by (D-SC) and (D-R1). Since the latter two are equivalent, (D-LG) is equivalent to them as well.

5. Reduction for Affine ALP under Customer Choice Behavior

In this section, we generalize the result in Section 3 to the choice-based revenue management context. The affine approximation for choice-based revenue management was first considered by Zhang and Adelman (2009).

The program (CD1) has a large number of variables but relatively few constraints, so it can be solved via column generation (Zhang and Adelman 2009). The procedure starts with an initial feasible solution corresponding to closing all products in each period; i.e., \( y_{t,x,\emptyset} = 1 \) for all \( t, x \). At a given iteration, suppose the dual solution is \( (V, \theta) \). For fixed \( t \geq 1 \), we need to solve the following column generation subproblem

\[ \text{(CS1)} \quad \max_{x, S} \sum_{j \in S} \lambda P_j(S) \left[ f_j - \sum_{i} a_{ij} V_{t+1,i} \right] - \sum_{i} (V_{t,i} - V_{t+1,i}) x_i - \theta_t + \theta_{t+1} \]
\[ x_i \geq a_{ij}, \quad \forall i, j \in S, \]
\[ x_i \in \{0, \ldots, c_i\}, \quad \forall i. \]

The effectiveness of a column generation algorithm hinges on efficient solution of the column generation subproblems. For general choice probability \( P \), (CS1) is a nonlinear integer constrained optimization problem, which is quite difficult to solve. For the multinomial logit choice model with disjoint consideration sets (MNLD), Zhang and Adelman (2009) show that (CS1) can be rewritten as a linear mixed integer programming problem under mild technical assumptions; we provide more details in Appendix B.

5.1. A Reduced Formulation for (CD1)

In this section, we show that the reduction in Proposition 2 can be generalized to the choice-based model. For each \( S \subseteq N \), let \( I(S) = \{i = 1, \ldots, m : \max_{j \in S} a_{ij} = 1\} \) be the set of resources used by at least one product in the set \( S \). We have the following result:
Proposition 4. The program (CD1) can be reduced to the following equivalent linear program:

\[
\begin{align*}
\text{(CD2)} & \quad \max_{r,h} \sum_{t,S \subseteq N} \lambda R(S) h_{t,S} \\
r_{t,i} &= \begin{cases} 
  c_i, & \text{if } t = 1, \\
  r_{t-1,i} - \sum_{S \subseteq N} \lambda Q_i(S) h_{t-1,S}, & \text{if } t > 1,
\end{cases} \quad \forall t, i, \\
\sum_{S \subseteq N: i \in I(S)} h_{t,S} &\leq r_{t,i}, \quad \forall t, i, \\
\sum_{S \subseteq N} h_{t,S} &= 1, \quad \forall t, \\
r, h &\geq 0.
\end{align*}
\]

The proof for Proposition 4 is very similar to the proof for Proposition 2 in the no-choice setting. We first show that (CD1) can be reduced to (CD2) by a change of variables. Next, we show that a feasible solution to (CD1) can be constructed from a proper Dantzig-Wolfe reformulation of (CD2), establishing the equivalence between the two programs. The steps are summarized in the following lemmas.

Lemma 6. Any feasible solution to (CD1) yields a feasible solution to (CD2) with the same objective value.

Proof. We introduce the following change of variables

\[
\begin{align*}
  r_{t,i} &= \sum_{x \in X_t \cap S \subseteq N(x)} x_i p_{t,x,S}, \quad \forall t, i, \\
  h_{t,S} &= \sum_{x \in X_t(S)} p_{t,x,S}, \quad \forall t, S.
\end{align*}
\]

In the above, \( X_t(S) = \{ x \in X_t : x \geq A^j, \forall j \in S \} \).

With this change of variables, (CD1) can be rewritten as (CD2). In particular, for each \( t, i \),

\[
\begin{align*}
\sum_{S \subseteq N: i \in I(S)} h_{t,S} &= \sum_{S \subseteq N: i \in I(S)} \sum_{x \in X_t(S)} p_{t,x,S} \\
&= \sum_{x \in X_t \cap S \subseteq N(x): i \in I(S)} p_{t,x,S} \\
&\leq \sum_{x \in X_t \cap S \subseteq N(x)} x_i p_{t,x,S} \\
&= r_{t,i}.
\end{align*}
\]

In the above, the inequality follows since for all \( S \neq \emptyset \), if \( i \in I(S) \) and \( p_{t,x,S} > 0 \), then \( x_i \geq 1 \).

By construction, any feasible solution \( y \) to (CD1) corresponds to a feasible solution \( (h, r) \) to (CD2). This completes the proof.
Lemma 7. For each $t$, let
\[
\Omega_t = \left\{ (r_t, h_t) : \sum_{S \subseteq N} h_{t,S} \leq r_{t,i}, \forall i, \sum_{S \subseteq N} h_{t,S} = 1, r_{t,i} \leq c_i, \forall i, h_t \geq 0, r_t \geq 0 \right\}
\]
be a bounded polyhedron. Then, $\Omega_t$ has integer extreme points.

Proof. Fix a point $(r_t, h_t) \in \Omega_t$. We show by construction that $(r_t, h_t)$ can be expressed as a convex combination of integer points in $\Omega_t$, which immediately implies that all extreme points are integral. That is, we want to show
\[
r_{t,i} = \sum_{k=1}^{K} r_{t,i}^{(k)} \mu_t^{(k)}, \forall i, \quad h_{t,S} = \sum_{k=1}^{K} h_{t,S}^{(k)} \mu_t^{(k)}, \forall S,
\]
where $E_t = \{(r_{t,i}^{(k)}, h_{t,S}^{(k)}) \in \Omega_t : k = 1, \ldots, K\}$ is a collection of integer points in $\Omega_t$ for some constant $K$ and $\mu_t^{(k)}$’s are nonnegative constants with $\sum_{k=1}^{K} \mu_t^{(k)} = 1$.

In order to show (38), we start with a collection of integer feasible points $(r_t^{S'}, h_t^{S'})$ indexed by $S' \subseteq \bar{N} = \{S \subseteq N : h_{t,S} > 0\}$, where
\[
h_{t,S}^{S'} = \begin{cases} 1, & \text{if } S = S', \\ 0, & \text{if } S \neq S', \end{cases}, \quad \forall S \subseteq N,
\quad r_{t,i}^{S'} = \max_{j \in S'} a_{ij}, \quad \forall i.
\]
Furthermore, let $\mu_t^{S'} = h_{t,S'}$. It follows that $\sum_{S' \in \bar{N}} \mu_t^{S'} = \sum_{S' \subseteq N} h_{t,S'} = 1$. It can be easily verified that $(r_t^{S'}, h_t^{S'}) \in \Omega_t$ for each $S'$, and
\[
r_{t,i} \geq \sum_{S' \subseteq \bar{N}} r_{t,i}^{S'} \mu_t^{S'}, \forall i, \quad h_{t,S} \leq \sum_{S' \subseteq \bar{N}} h_{t,S}^{S'} \mu_t^{S'}, \forall S.
\]

The condition (38) would hold if we can enlarge the set of integer feasible points to reach equality in (39). For ease of reference, we denote the set of points $E$ and index the points by $k = 1, \ldots, K$ with $K = |\bar{N}|$. In order to reach equality in (39), for each point $(r_{t,i}^{(k)}, h_{t,S}^{(k)})$, add another point $(r_{t,i}^{(K+k)}, h_{t,S}^{(K+k)})$ that is equal to $(r_{t,i}^{(k)}, h_{t,S}^{(k)})$ except $r_{t,i}^{(K+k)} = c_i$. Note that it is possible to redistribute the weights $\mu_t$ to reach equality for $i$ since $r_{t,i} \leq c_i$. Repeating this process for all $i$ yields the desired result. \hfill \blacksquare

Lemma 8. Suppose $(r, h)$ is a feasible solution to (CD2). Then, there exists a feasible solution $y$ to (CD1) with the same objective value.

Proof. We again establish this result using the Dantzig-Wolfe reformulation of (CD2). Let $\{(r_{t,i}^{(k)}, h_{t,S}^{(k)}) : k = 1, \ldots, K\}$ be the extreme points of polyhedron $\Omega_t$, for each period $t$. Then, any feasible solution $(r, h)$ to (CD2) can be expressed as $(r_t, h_t) = \sum_{k=1}^{K_t} (r_t^{(k)}, h_t^{(k)}) \mu_{t,k}$, where $\sum_{k=1}^{K_t} \mu_{t,k} = 1$ and $\mu \geq 0$. 
The Dantzig-Wolfe reformulation of (CD2) is given by

\[
\begin{align*}
\text{(CD2-DW)} & \quad \max_{\mu} \sum_{t,k \in \{1,\ldots,K_t\}} \lambda R(S)^{(k)} \mu_{t,k} \\
& \quad \sum_{k=1}^{K_t} r_{t,i}^{(k)} \mu_{t,k} = \left\{ \begin{array}{ll}
\sum_{k=1}^{K_{t-1}} \left( r_{t,i}^{(k)} - \lambda Q_i(S) h_{t-1,S}^{(k)} \right) \mu_{t-1,k}, & \text{if } t > 1, \\
\mu_{t,k}, & \text{if } t = 1,
\end{array} \right. \forall i,t, \\
& \quad \sum_{k=1}^{K_t} \mu_{t,k} = 1, \forall t, \\
& \quad \mu \geq 0.
\end{align*}
\]

From Lemma 7, the extreme points are integral. Furthermore, from the definition of \( \Omega_t \), they correspond to feasible state-action pairs. Thus, it is straightforward to construct a feasible solution to (CD1) by letting \( y_{t,r}^{(k)} = \mu_{t,k} \) for each \( t \) and \( k \in 1,\ldots,K_t \) such that \( h_{t,S}^{(k)} = 1 \), and letting \( p_{t,x,S} = 0 \) otherwise.

It is easy to check that this construction generates a solution that will satisfy \( \sum_{t,S \subseteq N} \lambda R(S) h_{t,S} = \sum_{t,x \in \chi_t, S \subseteq N(s)} \lambda R(S) p_{t,x,S} \), which completes the proof.

\[\blacksquare\]

### 5.2. Customer Segment Partition under Disjoint Consideration Sets

The formulation (CD2) still has an exponential number of decision variables. In this section, we would like to investigate whether further reductions are possible with additional information about the choice model. To do so, we refine our choice model by allowing for multiple customer segments. Specifically, let \( L = \{1,\ldots,\bar{l}\} \) be a given set of customer segments. We assume that customer segment \( l \in L \) has consideration set \( N_l \subseteq N \). When presented with a choice set \( S_l \subseteq N_l \), a customer in segment \( l \) will choose product \( j \) with probability \( P_{lj}(S_l) \). We further assume that an arriving customer first chooses which segment she belongs to, and then chooses products within the given segment. In particular, we assume the probability an arriving customer belongs to segment \( l \) is \( \lambda_l / \lambda \) where \( \sum_l \lambda_l = \lambda \). It then follows that for any product \( j \in N \),

\[ P_j(S) = \frac{1}{\lambda} \sum_l \lambda_l P_{lj}(S \cap N_l). \]

We assume \( N_{l_1} \cap N_{l_2} = \emptyset \) for all \( l_1 \neq l_2 \); in other words, different customer segments have disjoint consideration sets. Discrete choice models with disjoint consideration sets are used widely in the recent research in this area; see, for example, Liu and van Ryzin (2008) and Zhang and Adelman (2009). In those papers, the authors consider multinomial logit choice models with disjoint consideration sets, while here we do not limit our discussion to multinomial logit choice models.
Proposition 5. For discrete choice model with disjoint consideration sets, the formulation (CD2) can be rewritten as

\[(CD3)\]

\[
\max_{r,g} \sum_{t,l,S} \lambda R(S_t) g_{t,l,S_t}
\]

\[
r_{t,i} = \begin{cases} \frac{c_i}{r_{t-1,i}} - \sum_{l,S_l \subseteq N_l} \lambda Q_i(S_l) g_{t-1,l,S_l}, & \text{if } t > 1, \\ c_i, & \text{if } t = 1, \end{cases} \quad \forall t, i,
\]

\[
\sum_{l,S_l \subseteq N_l} g_{t,l,S_l} \leq r_{t,i}, \quad \forall t, i, l
\]

\[
\sum_{l,S_l \subseteq N_l} g_{t,l,S_l} = 1, \quad \forall t, l
\]

\[r, g \geq 0,
\]

Proof: We show the equivalence of (CD2) and (CD3) by establishing the relationship between feasible solutions of the two programs.

First, Suppose $h^*$ is a feasible solution to (CD2). We can show that the solution

\[g_{t,l,S_t} = \frac{\lambda_l}{\lambda} \sum_{S \subseteq N : S \cap N_l = S_l} h^*_{t,S_t}
\]

for all $t, l, S_l \subseteq N_l$ is a feasible solution to (CD3) and has the same objective value.

Observe that $\sum_{S_l \subseteq N_l} g_{t,l,S_l} = \lambda_l$ for all $t, l$. We have for any $t$

\[
\sum_{l,S_l \subseteq N_l} R(S_t) g_{t,l,S_t} = \sum_{l,S_l \subseteq N_l} \sum_{j \in S_l} f_j P_{ij}(S_l) g_{t,l,S_l}
\]

\[
= \sum_{l,S_l \subseteq N_l} \sum_{j \in S_l} f_j P_{ij}(S_l) \frac{\lambda_l}{\lambda} \sum_{S \subseteq N : S \cap N_l = S_l} h^*_{t,S}
\]

\[
= \sum_{S \subseteq N} \sum_{j \in N} f_j \frac{1}{\lambda} \sum_{l} \lambda_l P_{ij}(S \cap N_l) h^*_{t,S}
\]

\[
= \sum_{S \subseteq N} \sum_{j \in N} f_j P_j(S) h^*_{t,S}
\]

and

\[
\sum_{l,S_l \subseteq N_l} Q_i(S_l) g_{t,l,S_l} = \sum_{l,S_l \subseteq N_l} \sum_{j \in S_l} a_{ij} P_{ij}(S_l) g_{t,l,S_l}
\]

\[
= \sum_{l,S_l \subseteq N_l} \sum_{j \in S_l} a_{ij} P_{ij}(S_l) \frac{\lambda_l}{\lambda} \sum_{S \subseteq N : S \cap N_l = S_l} h^*_{t,S}
\]

\[
= \sum_{S \subseteq N} \sum_{j \in N} a_{ij} \frac{1}{\lambda} \sum_{l} \lambda_l P_{ij}(S \cap N_l) h^*_{t,S}
\]

\[
= \sum_{S \subseteq N} \sum_{j \in N} a_{ij} P_j(S) h^*_{t,S}
\]

\[= \sum_{S \subseteq N} Q_i(S) h^*_{t,S}
\]
Finally, we also have

\[ \sum_{S \subseteq N} g_{t,S} = \sum_{S \subseteq N} \frac{\lambda_i}{\lambda} \sum_{S \subseteq N} h^*_{t,S} \leq \frac{\lambda_i}{\lambda} \sum_{S \subseteq N} h^*_{t,S} \leq \frac{\lambda_i}{\lambda} \rho_{t,i} = \lambda_i \rho_{t,i}. \]

A feasible solution with the same objective value can be constructed for (CD3) from a feasible solution to (CD2).

Next, we show that a feasible solution with the same objective value can be constructed for (CD2) from a feasible solution to (CD3). Suppose \( g^* \) is a feasible solution to (CD3), we can prove that the solution

\[ h_{t,S} = \prod_{i} \prod_{l} g^*_{t,l,S \cap N_l} \]

for all \( t,S \subseteq N \) is a feasible solution to (CD2) and has the same objective value.

By construction, \( \sum_{S \subseteq N} h_{t,S} = \lambda \) for all \( t \). We have for any \( t \)

\[
\sum_{t,S \subseteq N} R(S)h_{t,S} = \sum_{S \subseteq N} \sum_{j} f_j P_j(S) \prod_{l} \lambda_i \prod_{l} g^*_{t,l,S \cap N_l} = \sum_{t,S \subseteq N} \sum_{j} f_j \prod_{l} \lambda_i \sum_{S \subseteq N} \prod_{l} g^*_{t,l,S \cap N_l} = \sum_{t,S \subseteq N} \sum_{j} f_j \prod_{l} \lambda_i \sum_{S \subseteq N} \prod_{l} \lambda_i = \sum_{t,S \subseteq N} R(S)g^*_{t,S}.
\]

and

\[
\sum_{t,S \subseteq N} Q(S)h_{t,S} = \sum_{S \subseteq N} \sum_{j} a_{ij} P_j(S) \prod_{l} \lambda_i \prod_{l} g^*_{t,l,S \cap N_l} = \sum_{t,S \subseteq N} \sum_{j} a_{ij} \prod_{l} \lambda_i \sum_{S \subseteq N} \prod_{l} g^*_{t,l,S \cap N_l} = \sum_{t,S \subseteq N} \sum_{j} a_{ij} \prod_{l} \lambda_i \sum_{S \subseteq N} \prod_{l} \lambda_i = \sum_{t,S \subseteq N} Q(S)g^*_{t,S}.
\]
= \sum_{l,S_l \subseteq N_l} \sum_{j} a_{ij} P_{lj}(S_l) g_{t,l,S_l}^*
= \sum_{t,l,S_l \subseteq N_l} Q_t(S_l) g_{t,l,S_l}.

Finally, we also have

\[
\sum_{S \subseteq N : i \in I(S)} h_{t,S} = \sum_{S \subseteq N : i \in I(S)} \prod_{l} \frac{\lambda}{\lambda_N} g_{t,l,S \cap N_l} \leq \lambda \left[ \sum_{l,S_l \subseteq N_l : i \in S_l} \prod_{l'} \frac{1}{\lambda_l} g_{t,l',S_l}^{*} \prod_{l' \neq l} \lambda_{l'} \right]
\leq \lambda \left[ \sum_{l,S_l \subseteq N_l : i \in S_l} \prod_{l'} \frac{1}{\lambda_l} r_{t,i} \right] = r_{t,i}
\]

This completes the proof.

Observe that when each customer segment consists of a single product, this formulation will be equivalent to the reduced formulation that was obtained for the independent demand case.

6. Summary and Future Directions

This paper considers reductions of ALPs for network RM problems resulting from (i) an affine functional approximation under independent demand, (ii) separable piecewise linear approximation under independent demand, and (iii) affine approximation under customer choice behavior. Result (i) was recently shown in Tong and Topaloglu (2011), while results (ii) and (iii) are new.

Central to our research is the connection between a reduced linear program and its Dantzig-Wolfe reformulation, which is closely related to the corresponding ALP. In order to establish equivalence, we need to explore properties of the underlying polyhedrons. In particular, we require that the polyhedrons 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.

It is natural to ask whether a similar reduction holds for separable piecewise linear approximation of choice-based network RM. It can be shown that the reduction technique can be extended to that
setting. However, unlike the case with independent demand, the reduced program would involve variables that are indexed by products, choice sets, and resource levels. Hence the reduced program is not compact. For this reason, we have decided not to focus on that case. It is still possible that some compact reduced programs can be found for special classes of choice models. We leave that topic to future research.

Acknowledgment

We would like to thank Huseyin Topaloglu for helpful discussions about this research.

References


