An Approximate Dynamic Programming Approach to Dynamic Pricing for Network Revenue Management

Jiannan Ke
School of Economics and Management, Wuhan University
Wuhan, 430072, China
hskejiannan@gmail.com

Dan Zhang*
Leeds School of Business, University of Colorado at Boulder
UCB 419, 995 Regent Dr, Boulder, CO 80309
dan.zhang@colorado.edu

Huan Zheng
Antai College of Economics and Management, Shanghai Jiao Tong University
1954 Huashan Road, Shanghai, 200030, China
zhenghuan@sjtu.edu.cn

Much of the network revenue management literature considers capacity control problems where product prices are fixed and the product availability is controlled over time. However, for industries with imperfect competition, firms typically retain some pricing power and dynamic pricing models are more realistic than capacity control models. Dynamic pricing problems are more challenging to solve; even the deterministic version is typically nonlinear. In this paper, we consider a dynamic programming model and use approximate linear programs (ALPs) to solve the problem. Unlike capacity control problems, the ALPs are semi-infinite linear programs, for which we propose a column generation algorithm. Furthermore, for the affine approximation under a linear independent demand model, we show that the ALPs can be reformulated as compact second order cone programs (SOCPs). The size of the SOCP formulation is linear in model primitives, including the number of resources, the number of products, and the number of periods. In addition, we consider a version of the model with discrete price sets and show that the resulting ALPs admit compact reformulations. We report numerical results on computational and policy performance on a set of hub-and-spoke problem instances.

Key words: network revenue management, dynamic pricing, approximate linear programs, semi-infinite linear programs, column generation algorithm, second order cone programs

History: Received: December 2017; Accepted: June 2019 by Huseyin Topaloglu, after 3 revisions.

1. Introduction

Network revenue management (NRM) is widely applied in industries where a set of products share a common set of resources and are sold in a finite selling horizon. Canonical examples exist in the airline and hotel industries, although its application is reported in more than a dozen industries (Talluri and van Ryzin 2004). In classical NRM applications, product prices are fixed and product

* Corresponding Author

availability is controlled dynamically over time to maximize expected revenue. Such problems are often referred to as the capacity control problem. Alternatively, product prices are varied over time, resulting in a dynamic pricing problem. For industries with imperfect competition, firms typically retain some pricing power and dynamic pricing models may be more realistic than capacity control models.

Capacity control problems received much more attention in the existing literature. It is common to formulate the problem using dynamic programming (Talluri and van Ryzin 1998, Bertsimas and Popescu 2003, Cooper 2002, Cooper and Homem-de-Mello 2007). However, solving the dynamic programming models is challenging due to the well-known curse of dimensionality, because of the need to keep track of resource levels over time. Much of the development in this literature concerns heuristic solution approaches to solving the NRM problems, including the aforementioned work of Bertsimas and Popescu (2003) and Cooper and Homem-de-Mello (2007).

While heuristic solution approaches to solving NRM problems can all be categorized as approximate dynamic programming (ADP), the development of a formal ADP framework was not accomplished until the work of Adelman (2007). Adelman (2007) considers the NRM problem and proposes a linear programming based ADP approach. His approach starts with an equivalent linear programming formulation of the dynamic programming formulation and approximates the value function with the weighted sum of a set of pre-selected basis functions. The resulting linear program is typically referred to as the approximate linear program (ALP). This framework was adopted by many authors who consider the NRM problem (Farias and Van Roy 2007, Zhang and Adelman 2009, Kunnumkal and Topaloglu 2010, Zhang 2011, Meissner and Strauss 2012, Kunnumkal and Talluri 2016).

The purpose of the current paper is to extend the framework in Adelman (2007) to dynamic pricing problems. The problem we consider is a discrete-time version of the dynamic pricing problem introduced in Gallego and van Ryzin (1997). Unlike the capacity control problem considered in Adelman (2007), the ALP is in general a semi-infinite linear program since prices can take infinitely many values. We show that this fact has little consequence as the ALP primal-dual pair is feasible and therefore has zero duality gap (Charnes et al. 1963). Consequently, the general framework in Adelman (2007) can be effortlessly generalized to the dynamic pricing setting with general demand functions. This should pave the way for further development in applying this approach to the dynamic pricing problem.

For the affine approximation under a linear independent demand model, we show that the ALP can be readily solved using column generation, where the column generation subproblem is a linear program. Furthermore, we show that the ALP can be equivalently formulated as a compact second order cone program (SOCP). Unlike the ALP, the size of the SOCP formulation is linear in model primitives, including the number of resources, the number of products, and the number of periods. We show via numerical test that the SOCP formulation can be efficiently solved using off-the-shelf commercial solvers. The solution speed is orders of magnitude faster than solving the ALP using column generation. We are also able to establish the superiority of heuristics based on the SOCP solution, relative to several benchmarks, including the dynamic programming decomposition based on a deterministic formulation. The latter is considered one of the strongest heuristic in the capacity control literature and was employed in existing research on dynamic pricing (Zhang and Lu 2013, Zhang and Weatherford 2017).

The popularity of linear demand functions is discussed in Besbes and Zeevi (2015). They study the cost of mis-specifying the demand function, where the simple parametric demand differs markedly from the underlying demand curve. They show that under fairly general conditions, this "price of misspecification" is not significant. In particular, "A two parameter linear model, in conjunction with the rather simple structure of the semimyopic pricing policy, guarantees that the resulting sequence of prices recovers the optimal price corresponding to the true (and unknown) underlying demand curve, regardless of the functional form of the latter." (Page 729) Even though it is possible to discretize prices, formulating the optimization problems with prices as decision variables has its own appeal, see also relevant discussion in Maglaras and Meissner (2006). This is confirmed by our discussions with industry practitioners; using price as the decision variable is quite common in the hotel industry and the linear demand curve is widely adopted due to its simplicity. In a recent presentation at the RM section conference in Toronto, Schön and Hohberger (2018) describe pricing problems at Deutsche Bahn, the Germany railway company. Interestingly, they mentioned that Deutsche Bahn recently switched from discrete prices to continuous prices in their decision support system. According to Wittman and Belobaba (2018), a similar trend is observed for the airline industry as well. Our paper focuses on the continuous price model, but we also consider a version with discrete price sets. We numerically test the performance of both continuous and discrete price formulations. Our results indicate that a large number of price points are needed in the discrete formulation to reach the solution accuracy of the continuous price model for the instances we considered. However, the model with discrete price sets can handle more general demand functions. Therefore, the model with discrete price sets is a reasonable choice when the continuous price model is difficult to handle. Our work provides a compact formulation and efficient solution method for the case of linear independent demand. For more general demand functions, we propose a column generation algorithm to solve the problem, which can be time consuming. In those cases, the formulation with discrete price sets is a reasonable alternative.

The comparison between continuous price model and its discrete price counterpart also leads to insights on pricing flexibility. The continuous price model can be viewed as a model of situations with full pricing flexibility, while the case with discrete price sets matches situations with limited pricing flexibility. Therefore, the revenue comparison between the two can be viewed as the value of pricing flexibility. Our numerical results indicate that the value of pricing flexibility can be substantial when the number of discrete price points is small.

Our work is closely related to studies that investigate alternative, and typically more compact, reformulations of ALPs for NRM problems. Farias and Van Roy (2007) propose a relaxed ALP reformulation (rALP) with reduced number of constraints and (almost) the same number of variables. They use numerical experiments to show that rALPs give the same optimal solutions as the original ALPs. Tong and Topaloglu (2013) show that the affine approximation admits a compact reformulation, which is called the reduced program. Vossen and Zhang (2015) point out that the original ALP can be viewed as a Dantzig-Wolfe reformulation of the reduced program (Dantzig and Wolfe 1960, 1961). They further show that this connection also holds for the separable piecewise linear approximation and for general discrete choice models of demand. In all these studies, the ALPs and their corresponding reformulations are regular linear programs since they all deal with the capacity control problem. In this paper, the original ALP is a semi-infinite linear program and the reformulation is a nonlinear program. Our analysis utilizes a Lagrangian dual approach and variable aggregation. Specifically, we develop a SOCP to handle the case with linear independent demand.

Our use of SOCP in NRM is preceded by Kirshner and Nediak (2015), who consider a continuous-time formulation of the capacity control problem and use splines to represent derivatives of bid prices, which modifies the problem into a SOCP. Their formulation takes advantage of the continuous time formulation to ensure that the number of variables does not grow with the length of the booking horizon, unlike ALPs from the typical discrete time formulation. However, the number of constraints is still growing exponentially in the number of resources. In contrast, we study the dynamic pricing problem and the size of our SOCP formulation grows linearly in the number of periods and the number of resources.

Dynamic pricing for NRM can be traced back to Gallego and van Ryzin (1997). It shares the curse of dimensionality with the capacity control problem, but received much less attention in the literature. Heuristic solutions for the dynamic pricing problem were considered by several authors, including Gallego and van Ryzin (1997), Maglaras and Meissner (2006), Dong et al. (2009), Erdelyi and Topaloglu (2010), and Zhang and Lu (2013). With the exception of Zhang and Lu (2013) and Lim et al. (2015), these papers do not utilize linear programming based ADP approach. Even though Zhang and Lu (2013) briefly discuss the ALP formulation, their proposed approach is a network decomposition method that does not require solving the ALP and they focus on the multinomial logit demand model. Lim et al. (2015) formulate a hotel pricing problem as a finite-horizon dynamic program. Their formulation assumes that the price is day-specific but not product-specific; that is, a different price is charged for each day, and customers who stay multiple days pay the sum of prices across different days. They study structural properties of the dynamic program and use column generation to solve the affine ALP. Zhang and Weatherford (2017) apply the network decomposition approach from Zhang (2011) to the dynamic hotel pricing problem and validate their approach using data from an international hotel chain. Even though we do not emphasize the specific application context in this paper, we are hopeful that the approach proposed here can be applied to these specific settings.

ADP is a fast-growing research area; see Powell (2011) and Bertsekas and Tsitsiklis (1996) for various approaches and applications. The idea of functional approximation in approximate linear programming, i.e., using a linear combination of a set of basis functions to approximate value functions, is first raised by Schweitzer and Seidmann (1985) and further considered by de Farias and Van Roy (2003). As evidenced by its application to network capacity control problems, solving the large-scale ALPs is the main challenge of this approach. Our result should encourage researchers to look for compact representations of ALPs in other applications, even if they are not necessarily linear. The hope is that these compact reformulations can be more efficiently solved.

SOCP is a well studied convex optimization problem, second perhaps only to linear programs. The standard form of SOCP minimizes a linear objective function over a set of affine and second-order cone constraints. SOCP is widely used in analyzing and solving convex quadratic constrained quadratic programs (QCQP), norm optimizations, and robust optimization problems. Alizadeh and Goldfarb (2003) offer an excellent overview of SOCP reformulations to classical optimization problems. Lobo et al. (1998) discuss SOCP applications to engineering problems and the complexity of solving SOCP with interior-point methods. There are many commercial and open source solvers for SOCPs, including Gurobi, CVX, Knitro, and SDPT3. We use Gurobi in our numerical study.

The rest of the paper is organized as follows. Section 2 introduces the problem formulation. Section 3 considers a general ADP framework. Section 4 specializes to the case of affine approximation with linear independent demand and Section 5 shows a compact SOCP formulation. Section 6 considers pricing problems with discrete price sets. Section 7 introduces several heuristic policies and

Section 8 reports numerical results. Section 9 concludes. The Appendix contains some additional analysis and proofs.

2. Problem Formulation

Consider a firm endowed with m resources. The initial capacity of the resources is given by the vector $\mathbf{c} = (c_1, c_2, ..., c_m)^T$, where the i-th element c_i is the capacity for resource i and T denotes vector transpose. A set of products $N = \{1, \dots, n\}$ are offered, each using a subset of resources. An $m \times n$ matrix A represents the resource consumption and the (i, j)-th element a_{ij} is the quantity of resource i consumed by one unit of product j; $a_{ij} = 1$ if product j uses resource i and $a_{ij} = 0$ otherwise. We also let A_i and A^j denote the i-th row and the j-th column of A, respectively. The firm sells the products in a finite selling horizon, which is divided into T "small" discrete time periods with no more than one customer arrival in each period. Let $\mathbf{r}_t = (r_{t,1}, \dots, r_{t,n})^T$ be the vector of prices in period t. Sales of a unit of product j occurs in period t if there is a customer arrival and the customer purchases product j after observing the price vector $\mathbf{r}_t$; we denote the probability of this event $\lambda_{t,j}(\mathbf{r}_t)$. Since there is at most one customer arrival in each period, $\sum_j \lambda_{t,j}(\mathbf{r}_t) \le 1$ for each t. In this general form, the demand function $\lambda_{t,j}(\mathbf{r}_t)$ for product j in period t depends on the prices of all products and therefore explicitly models product substitution. In later sections, we specialize to a linear independent demand model that does not allow product substitution.

The firm's objective is to maximize its total expected revenue by adjusting prices over time. The firm's dynamic pricing problem can be formulated as a discrete-time finite-horizon dynamic program. The state in period t is the remaining capacity $\mathbf{x} = (x_1, \dots, x_m)^T$. The state space is

$$\mathcal{X}_t = \begin{cases} \{\mathbf{c}\}, & \text{if } t = 1, \\ \{\mathbf{x} \in \mathbb{Z}_+^m : \mathbf{x} \le \mathbf{c}\}, & \text{if } t > 1, \end{cases}$$

where $\mathbb{Z}_+$ denotes the set of nonnegative integers.

Let $v_t(\mathbf{x})$ be the maximum total expected future revenue given state $\mathbf{x}$ in period t. For each t and $\mathbf{x}$, the optimality equation is

$$\begin{aligned} v_t(\mathbf{x}) &= \max_{\mathbf{r}_t \in \mathcal{R}_t(\mathbf{x})} \left\{ \sum_j \lambda_{t,j}(\mathbf{r}_t) \left(r_{t,j} + v_{t+1}(\mathbf{x} - A^j) \right) + \left(1 - \sum_j \lambda_{t,j}(\mathbf{r}_t) \right) v_{t+1}(\mathbf{x}) \right\} \\ &= \max_{\mathbf{r}_t \in \mathcal{R}_t(\mathbf{x})} \left\{ \sum_j \lambda_{t,j}(\mathbf{r}_t) \left(r_{t,j} + v_{t+1}(\mathbf{x} - A^j) - v_{t+1}(\mathbf{x}) \right) \right\} + v_{t+1}(\mathbf{x}), \end{aligned} \tag{1}$$

where the action set is $\mathcal{R}_t(\mathbf{x}) = \times_{j=1}^n \mathcal{R}_{t,j}(\mathbf{x})$ and

$$\mathcal{R}_{t,j}(\mathbf{x}) = \begin{cases} \mathbb{R}_+, & \text{if } \mathbf{x} \ge A^j, \\ r_\infty, & \text{otherwise}, \end{cases} \qquad \forall t, j.$$

In the above, $\mathbb{R}_+$ denotes nonnegative reals and r_∞ is the null price such that $\lambda_{t,j}(\mathbf{r}_t)=0$ if $r_{t,j}=r_\infty$ for all t,j. The boundary conditions are $v_{T+1}(\mathbf{x})=0$ for each $\mathbf{x}$ and $v_t(\mathbf{0})=0$ for each t.

Due to the well-known curse of dimensionality, the dynamic programming model is difficult to solve when $\mathbf{x}$ is high dimensional. We use linear programming based approximate dynamic programming (Adelman 2007, Zhang and Adelman 2009) to solve the problem.

The linear programming formulation for the dynamic program above is

$$(\mathbf{P}) \qquad \inf_{\{v_t(\cdot)\}_{\forall t}} \quad v_1(\mathbf{c})$$

$$\text{s.t.} \qquad v_t(\mathbf{x}) \geq \sum_j \lambda_{t,j}(\mathbf{r}_t)\left(r_{t,j}+v_{t+1}(\mathbf{x}-\mathbf{A}^j)-v_{t+1}(\mathbf{x})\right)+v_{t+1}(\mathbf{x}), \quad \forall\ t,\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x}).$$

The decision variables in $(\mathbf{P})$ are the $v_t(\mathbf{x})$ for all t and $\mathbf{x}$. When $\mathbf{x}$ is high dimensional, the number of decision variables can be huge. The constraints are indexed by $t,\mathbf{x}$, and $\mathbf{r}_t$. Since $\mathcal{R}_t(\mathbf{x})$ is a continuous set, $(\mathbf{P})$ has an infinite number of constraints. Therefore, $(\mathbf{P})$ is a linear semi-infinite program. In the next section, we lay out a general framework to approximately solve $(\mathbf{P})$.

3. General Framework

In this section, we give an ADP framework under a general value function approximation. Section 3.1 establishes the strong duality of the primal and dual formulations. Section 3.2 introduces a column generation algorithm.

Consider the value function approximation

$$v_t(\mathbf{x}) \approx \theta_t + \sum_{b\in\mathcal{B}} V_{t,b}\phi_b(\mathbf{x}), \quad \forall\ t,\mathbf{x}\in\mathcal{X}_t, \tag{2}$$

where $\{\phi_b(\mathbf{x})\}_{\forall b\in\mathcal{B}}$ is a set of pre-selected basis functions with index set $\mathcal{B}$. The parameter θ_t corresponds to a constant basis function and $V_{t,b}$ weighs basis function $\phi_b(\mathbf{x})$ at time t. For notational convenience, we take $\theta_{T+1}=0$ and $V_{T+1,b}=0$ for all $b\in\mathcal{B}$. The formulation $(\mathbf{P})$ under the approximation (2) continues to be a linear semi-infinite program. Unlike regular linear programs, strong duality does not necessarily hold for semi-infinite linear programs. We show that strong duality holds in our setting, i.e., the duality gap is zero. We introduce a column generation algorithm to solve the dual program. Taken together, our results in this section imply that $(\mathbf{P})$ under the approximation (2) can be solved for quite general demand functions for reasonable sized problems.

3.1 Strong Duality

Plugging (2) into $(\mathbf{P})$, we obtain

$$(\mathbf{P})^\phi \qquad \inf_{\theta,\mathbf{V}} \quad \theta_1 + \sum_{b\in\mathcal{B}} V_{1,b}\phi_b(\mathbf{c})$$

$$\text{s.t.} \quad \theta_t - \theta_{t+1} + \sum_{b\in\mathcal{B}} (V_{t,b} - V_{t+1,b})\phi_b(\mathbf{x}) + \sum_j \lambda_{t,j}(\mathbf{r}_t) \sum_{b\in\mathcal{B}} V_{t+1,b}\left(\phi_b(\mathbf{x}) - \phi_b(\mathbf{x}-\mathbf{A}^j)\right) \\ \geq \sum_j \lambda_{t,j}(\mathbf{r}_t) r_{t,j}, \quad \forall\ t, \mathbf{x}\in\mathcal{X}_t, \mathbf{r}_t\in\mathcal{R}_t(\mathbf{x}). \tag{3}$$

Like $(\mathbf{P})$, $(\mathbf{P})^\phi$ is a linear semi-infinite program, albeit with a smaller number of decision variables.

The dual of $(\mathbf{P})^\phi$ has finite constraints but infinite number of variables and can be written as

$$(\mathbf{D})^\phi \quad \sup_{\mathbf{p}} \sum_{t,\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} \left(\sum_j \lambda_{t,j}(\mathbf{r}_t) r_{t,j}\right) p_{t,\mathbf{x},\mathbf{r}_t}$$

$$\text{s.t.} \quad \sum_{\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} \phi_b(\mathbf{x}) p_{t,\mathbf{x},\mathbf{r}_t} \\ = \begin{cases} \phi_b(\mathbf{c}), & \text{if } t=1, \\ \displaystyle\sum_{\mathbf{x}\in\mathcal{X}_{t-1},\mathbf{r}_{t-1}\in\mathcal{R}_{t-1}(\mathbf{x})} \phi_b(\mathbf{x}) p_{t-1,\mathbf{x},\mathbf{r}_{t-1}} - \\ \displaystyle\sum_{\mathbf{x}\in\mathcal{X}_{t-1},\mathbf{r}_{t-1}\in\mathcal{R}_{t-1}(\mathbf{x})} p_{t-1,\mathbf{x},\mathbf{r}_{t-1}} \sum_j \lambda_{t-1,j}(\mathbf{r}_{t-1})\left(\phi_b(\mathbf{x}) - \phi_b(\mathbf{x}-\mathbf{A}^j)\right), & \text{if } t>1, \end{cases} \\ \forall\ t, b\in\mathcal{B}, \tag{4}$$

$$\sum_{\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} p_{t,\mathbf{x},\mathbf{r}_t} = 1, \qquad \forall t, \tag{5}$$

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

Since $(\mathbf{P})^\phi$-$(\mathbf{D})^\phi$ are linear semi-infinite programs, the duality gap can be positive. Following Charnes et al. (1963), the duality gap is zero as long as both programs are consistent (feasible). Therefore, it suffices to check the feasibility of both programs to establish strong duality, which is presented in the following proposition. The proof is in Appendix A.

Proposition 1. *The duality gap between the primal-dual pair* $(\mathbf{P})^\phi$-$(\mathbf{D})^\phi$ *is 0.*

3.2 A Column Generation Algorithm for $(\mathbf{P})^\phi$-$(\mathbf{D})^\phi$

Since $(\mathbf{P})^\phi$-$(\mathbf{D})^\phi$ does not have duality gap, the program can be solved by applying the column generation algorithm to the dual program $(\mathbf{D})^\phi$. For each period t and primal solution $(\theta, \mathbf{V})$, the column generation subproblem is

$$\pi_t = \min_{\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} \theta_t - \theta_{t+1} + \sum_{b\in\mathcal{B}} (V_{t,b} - V_{t+1,b})\phi_b(\mathbf{x}) \\ + \sum_j \lambda_{t,j}(\mathbf{r}_t) \sum_{b\in\mathcal{B}} V_{t+1,b}\left(\phi_b(\mathbf{x}) - \phi_b(\mathbf{x}-\mathbf{A}^j)\right) - \sum_j \lambda_{t,j}(\mathbf{r}_t) r_{t,j}.$$

Optimality is reached when $\pi_t \geq 0$ for all t.

Since it may take great computational effort to reach optimality in the column generation algorithm, it is common to stop the column generation algorithm when optimality is guaranteed within an optimality tolerance. Let z^ϕ denote the optimal objective function value for $(\mathbf{P})^\phi$-$(\mathbf{D})^\phi$ and $z^{\tilde{\pi}}$ be the objective value reached in the column generation algorithm when the column generation subproblem takes the value $\tilde{\pi}_t$ ($\tilde{\pi}_t \leq 0$) for each t. Then the following relationship holds:

PROPOSITION 2. $z^{\tilde{\pi}} \geq z^\phi + \sum_t \tilde{\pi}_t$.

Proof. Let $(\tilde{\theta}, \tilde{\mathbf{V}})$ stand for the restricted solution $z^{\tilde{\pi}}$ with reduced cost $\tilde{\pi}$ and $\mathbf{p}^*$ stand for the dual solution z^ϕ of dual problem $(\mathbf{D})^\phi$. We have

$$\begin{aligned}
& z^{\tilde{\pi}} - z^\phi \\
= & \left(\tilde{\theta}_1 + \sum_b \phi_b(\mathbf{c})\tilde{V}_{1,b}\right) - \sum_{t,\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} \left(\sum_j \lambda_{t,j}(\mathbf{r}_t) r_{t,j}\right) p^*_{t,\mathbf{x},\mathbf{r}_t} \\
= & \sum_{t,\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} p^*_{t,\mathbf{x},\mathbf{r}_t}(\tilde{\theta}_t - \tilde{\theta}_{t+1}) + \left[\sum_{t,b}\left(\sum_{\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} \phi_b(\mathbf{x}) p^*_{t,\mathbf{x},\mathbf{r}_t}\right)\tilde{V}_{t,b} - \right. \\
& \left. \sum_{t,b}\left(\sum_{\mathbf{x}\in\mathcal{X}_{t+1},\mathbf{r}_{t+1}\in\mathcal{R}_{t+1}(\mathbf{x})} \phi_b(\mathbf{x}) p^*_{t+1,\mathbf{x},\mathbf{r}_{t+1}}\right)\tilde{V}_{t+1,b}\right] - \sum_{t,\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} \left(\sum_j \lambda_{t,j}(\mathbf{r}_t) r_{t,j}\right) p^*_{t,\mathbf{x},\mathbf{r}_t} \\
= & \sum_{t,\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} p^*_{t,\mathbf{x},\mathbf{r}_t} \\
& \left[\tilde{\theta}_t - \tilde{\theta}_{t+1} + \sum_{b\in\mathcal{B}} (\tilde{V}_{t,b} - \tilde{V}_{t+1,b})\phi_b(\mathbf{x}) + \sum_j \lambda_{t,j}(\mathbf{r}_t) \sum_{b\in\mathcal{B}} \tilde{V}_{t+1,b}\left(\phi_b(\mathbf{x}) - \phi_b(\mathbf{x}-\mathbf{A}^j)\right) - \sum_j \lambda_{t,j}(\mathbf{r}_t) r_{t,j}\right] \\
\geq & \sum_{t,\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} \tilde{\pi}_t p^*_{t,\mathbf{x},\mathbf{r}_t} = \sum_t \tilde{\pi}_t.
\end{aligned}$$

In the above, the inequality is satisfied because the terms in the bracket represent the objective function of the column generation subproblem, whose smallest value is π_t in period t. The second equality is satisfied because of (5) and $\theta_{T+1} = 0, V_{t+1,b} = 0\ \forall b \in \mathcal{B}$. The third equality is satisfied if we replace the second term in the bracket with the right hand side of (4). ■

Since $\tilde{\pi}_t < 0$ for all t, $z^{\tilde{\pi}}$ is smaller than z^ϕ. Proposition 2 bounds the gap between $z^{\tilde{\pi}}$ and z^ϕ. Before proceeding, we comment that Proposition 2 allows us to construct a relative optimality guarantee. Suppose $-\sum_t \tilde{\pi}_t / z^{\tilde{\pi}} \leq \Omega$, then

$$\frac{z^\phi}{z^{\tilde{\pi}}} \leq \frac{z^{\tilde{\pi}} - \sum_t \tilde{\pi}_t}{z^{\tilde{\pi}}} \leq 1 + \Omega.$$

We use this relative optimality guarantee in our numerical study.

4. Affine Approximation with Linear Independent Demand

In this section, we consider the general problem introduced in the previous section with two restrictions: affine approximation and linear independent demand. Section 4.1 introduces a column generation algorithm for the restricted setup.

The affine approximation is given by

$$v_t(\mathbf{x}) \approx \theta_t + \sum_i V_{t,i} x_i, \quad \forall\, t, \mathbf{x}. \tag{6}$$

We also adopt the notational convention that $\theta_{T+1} = 0$ and $V_{T+1,i} = 0$ for all i. Affine approximation (6) is a special case of (2) where $\mathcal{B} = \{1, \ldots, m\}$ and $\phi_i(\mathbf{x}) = x_i$ for $i = 1, \ldots, m$.

The linear independent demand function is given by

$$\lambda_{t,j}(\mathbf{r}_t) \equiv \lambda_{t,j}(r_{t,j}) = (\alpha_{t,j} - \beta_{t,j} r_{t,j})^+, \qquad \forall\, t, j, r_{t,j} \in \mathbb{R}_+. \tag{7}$$

In (7), the demand in period t for product j is a linear function of product j's price $r_{t,j}$, where parameters $\alpha_{t,j}, \beta_{t,j}$ are positive for each t and j. Naturally, we restrict the demand function to be nonnegative and $\sum_j \alpha_{t,j} \leq 1$ for each t. Alternatively, we can write (7) as

$$\lambda_{t,j}(r_{t,j}) = \alpha_{t,j} - \beta_{t,j} r_{t,j}, \qquad \forall\, t, j, r_{t,j} \in \left[0, \frac{\alpha_{t,j}}{\beta_{t,j}}\right]. \tag{8}$$

Note that (8) explicitly restricts the price range for product j in period t to $[0, \frac{\alpha_{t,j}}{\beta_{t,j}}]$. To accommodate this price restriction, we introduce a slightly different definition of price set $\hat{\mathcal{R}}_t(\mathbf{x}) = \times_{j=1}^n \hat{\mathcal{R}}_{t,j}(\mathbf{x})$ for each t and

$$\hat{\mathcal{R}}_{t,j}(\mathbf{x}) = \begin{cases} \left[0, \frac{\alpha_{t,j}}{\beta_{t,j}}\right], & \text{if } \mathbf{x} \geq A^j, \\ \frac{\alpha_{t,i}}{\beta_{t,i}}, & \text{otherwise}, \end{cases} \qquad \forall t, j.$$

In the above, the null price r_∞ is taken to be $\frac{\alpha_{t,i}}{\beta_{t,i}}$ for product j, which forces the demand to 0. In Appendix B, we show that the analysis can be extended to log-linear independent demand.

Using (6) and (8) in $(\mathbf{D})^\phi$, we have

$$(\mathbf{D})_L^A \quad z^A = \sup_{\mathbf{p}} \sum_{t, \mathbf{x} \in \mathcal{X}_t, \mathbf{r}_t \in \hat{\mathcal{R}}_t(\mathbf{x})} \left(\sum_j (\alpha_{t,j} - \beta_{t,j} r_{t,j}) r_{t,j} \right) p_{t,\mathbf{x},\mathbf{r}_t}$$

$$\text{s.t.} \quad \sum_{\mathbf{x} \in \mathcal{X}_t, \mathbf{r}_t \in \hat{\mathcal{R}}_t(\mathbf{x})} x_i p_{t,\mathbf{x},\mathbf{r}_t}$$

$$= \begin{cases} c_i, & \text{if } t = 1, \\ \sum\limits_{\mathbf{x} \in \mathcal{X}_t, \mathbf{r}_{t-1} \in \hat{\mathcal{R}}_{t-1}(\mathbf{x})} \left(x_i - \sum_j (\alpha_{t-1,j} - \beta_{t-1,j} r_{t-1,j}) a_{ij} \right) p_{t-1,\mathbf{x},\mathbf{r}_{t-1}}, & \text{if } t > 1, \end{cases} \quad \forall\, t, i,$$

$$\sum_{\mathbf{x} \in \mathcal{X}_t, \mathbf{r}_t \in \hat{\mathcal{R}}_t(\mathbf{x})} p_{t,\mathbf{x},\mathbf{r}_t} = 1, \qquad \forall\, t,$$

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

The dual problem $(\mathbf{D})_L^A$ has only $(m+1)T$ constraints but infinite number of variables $p_{t,\mathbf{x},\mathbf{r}_t}$ because $\mathbf{r}_t$ is continuous.

One useful property of the solution of the affine approximation is that there exists an optimal solution where θ and $\mathbf{V}$ are non-increasing in t. Adelman (2007) shows this result in the context of capacity controls problems. We can show the same result for the pricing problem. The proof is relegated to Appendix C.

4.1 Column Generation Subproblem

Column generation can be applied to solve the problem $(\mathbf{D})_L^A$. For a given primal solution $(\theta, \mathbf{V})$, the column generation subproblem for each period t is

$$
\begin{aligned}
(\mathbf{CG})_L^A \qquad & \min_{\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t} \quad \theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i})x_i - \sum_j (\alpha_{t,j} - \beta_{t,j} r_{t,j}) \left(r_{t,j} - \sum_i a_{ij} V_{t+1,i} \right), \\
& x_i \in \{0, 1, ..., c_i\}, \qquad \forall i, \\
& r_{t,j} \in \hat{\mathcal{R}}_{t,j}(\mathbf{x}), \qquad \forall\ t, j, \mathbf{x} \in \mathcal{X}_t.
\end{aligned}
$$

Note that $(\mathbf{CG})_L^A$ is a nonlinear integer programming problem and therefore is potentially difficult to solve. In the following, we show that this problem can be efficiently solved as a linear program by exploiting the special structure of the problem.

Fix t. Observe that the last term in the objective function should be maximized for given $(\theta, \mathbf{V})$ and $\mathbf{x}$. With slight abuse of notation, for each j, define

$$
H_{t,j}(\mathbf{V}_{t+1}, \mathbf{x}) = \max_{r_{t,j} \in \hat{\mathcal{R}}_{t,j}(\mathbf{x})} (\alpha_{t,j} - \beta_{t,j} r_{t,j}) \left(r_{t,j} - \sum_i a_{ij} V_{t+1,i} \right).
$$

If $x_i < a_{ij}$ for any i, $H_{t,j}(\mathbf{V}_{t+1}, \mathbf{x}) = 0$. Now suppose $\mathbf{x} \geq A^j$, $H_{t,j}(\mathbf{V}_{t+1}, \mathbf{x})$ can be obtained through first order condition taking into account the bound on price $r_{t,j}$. Then the optimal solution $r^*_{t,j}(\mathbf{V}_{t+1})$ is

$$
r^*_{t,j}(\mathbf{V}_{t+1}) = \begin{cases} \frac{\beta_{t,j} \sum_i a_{ij} V_{t+1,i} + \alpha_{t,j}}{2\beta_{t,j}}, & \text{if } \sum_i a_{ij} V_{t+1,i} \leq \frac{\alpha_{t,j}}{\beta_{t,j}}, \\ \frac{\alpha_{t,j}}{\beta_{t,j}}, & \text{otherwise.} \end{cases} \qquad \forall t, j.
$$

The corresponding value for $H_{t,j}(\mathbf{V}_{t+1}, \mathbf{x})$ is independent of $\mathbf{x}$ and can be written as

$$
h_{t,j}(\mathbf{V}_{t+1}) = \begin{cases} \frac{\left(\alpha_{t,j} - \beta_{t,j} \sum_i a_{ij} V_{t+1,i}\right)^2}{4\beta_{t,j}}, & \text{if } \sum_i a_{ij} V_{t+1,i} \leq \frac{\alpha_{t,j}}{\beta_{t,j}}, \\ 0, & \text{otherwise.} \end{cases} \qquad \forall t, j. \tag{9}
$$

Let w_j indicate whether $\mathbf{x} \geq A^j$ for each j. Then we have $H_{t,j}(\mathbf{V}_{t+1}, \mathbf{x}) = w_j h_{t,j}(\mathbf{V}_{t+1})$. Note that w needs to satisfy

$$
a_{ij} w_j \leq x_i, \quad \forall i, j.
$$

Since $V_{t,i}$ is non-increasing in t, $V_{t,i} - V_{t+1,i}$ is non-negative in the objective function of problem $(\mathbf{CG})_L^A$. To minimize the objective function value, x_i should take the value 1 if $a_{ij}w_j = 1$ and 0 otherwise. Thus x_i can be reduced to a binary variable instead of an integer variable in $\{0, 1, ..., c_i\}$. Then the column generation subproblem $(\mathbf{CG})_L^A$ in each period can be reformulated as a binary integer programming problem

$$
\begin{aligned}
(\mathbf{CG1})_L^A \qquad & \min_{\mathbf{x},\mathbf{w}} \quad \theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i})x_i - \sum_j w_j h_{t,j}(\mathbf{V}_{t+1}), \\
\text{s.t.} \qquad & x_i \geq a_{ij} w_j, \quad \forall\ i, j, && (10) \\
& \mathbf{x}, \mathbf{w} \text{ binary}. && (11)
\end{aligned}
$$

Note that $(\mathbf{CG1})_L^A$ can be solved by solving its linear programming relaxation, where the binary constraints (11) is relaxed to $0 \leq x_i, w_j \leq 1, \forall i, j$. To see this, it suffices to show that the constraint matrix of (10) is totally unimodular. Observe that the transpose of the constraint matrix has at most two nonzero entries in each column. Furthermore, there are two nonzero entries only when $a_{ij} \neq 0$ and the corresponding two nonzero coefficients sum to zero. Thus the constraint matrix of (10) is totally unimodular; see Section III.1, Proposition 2.6 in Wolsey and Nemhauser (1988). This observation should greatly reduce the computational effort to solve $(\mathbf{D})_L^A$ since we only need to solve linear programs to generate columns.

5. A Compact Nonlinear Programming Formulation

Our overall objective in this section is to introduce a compact reformulation of $(\mathbf{D})_L^A$. Section 5.1 develops a reformulation based on duality arguments. Section 5.2 shows that the reformulated program can be further aggregated and is equivalent to a compact SOCP with a small number of constraints and variables. Section 5.3 considers the deterministic nonlinear programming formulation and explores its relationship to the compact SOCP formulation.

5.1 Reformulation and Strong Duality

The reformulated column generation subproblem $(\mathbf{CG1})_L^A$ implies that the primal problem $(\mathbf{P})^\phi$ under affine approximation and the linear independent demand function can be written as

$$
\begin{aligned}
(\mathbf{P})^{A,R} \quad & \min_{\theta,\mathbf{V}} \quad \theta_1 + \sum_i V_{1,i} c_i \\
\text{s.t.} \quad & \theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i}) x_i - \sum_j w_j h_{t,j}(\mathbf{V}_{t+1}) \geq 0 \quad \forall\ \mathbf{x} \in \mathcal{X}_t, \mathbf{w} \text{ binary} : a_{ij} w_j \leq x_i, \forall\ t, i, j.
\end{aligned}
\tag{12}
$$

In the above, we take $\mathbf{x} \in \mathcal{X}_t$ instead of binary $\mathbf{x}$ for development later in the paper. Observe that $(\mathbf{P})^{A,R}$ is a regular nonlinear program instead of a linear semi-infinite program.

Because $h_{t,j}(\mathbf{V}_{t+1})$ is nonnegative, it is without loss of optimality to restrict our attention to $w_j = 1$ whenever $\mathbf{x} \geq A^j$ and $w_j = 0$ otherwise. Define an indicator function

$$w_j(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \geq A^j, \\ 0, & \text{otherwise,} \end{cases} \quad \forall\, j.$$

Introducing a new variable $\rho_{t,j} = \max\left\{\frac{\alpha_{t,j}}{\beta_{t,j}} - \sum_i a_{ij} V_{t+1,i}, 0\right\}$ and using the expression for $h_{t,j}(\cdot)$ in (9), $(\mathbf{P})^{A,R}$ becomes

$$\begin{aligned}
(\mathbf{P}1)^{A,R} \qquad & \min_{\theta,\mathbf{V},\rho} \quad \theta_1 + \sum_i V_{1,i} c_i \\
\text{s.t.} \qquad & \theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i}) x_i - \sum_j \frac{\beta_{t,j} w_j(\mathbf{x})}{4} \rho_{t,j}^2 \geq 0, && \forall\, t, \mathbf{x} \in \mathcal{X}_t, && (13) \\
& \rho_{t,j} \geq \frac{\alpha_{t,j}}{\beta_{t,j}} - \sum_i a_{ij} V_{t+1,i}, && \forall\, t, j, && (14) \\
& \rho_{t,j} \geq 0, && \forall\, t, j.
\end{aligned}$$

Since constraint (13) is convex, $(\mathbf{P}1)^{A,R}$ is a convex quadratically constrained quadratic problem (QCQP).

Associating dual variables (μ, δ) to the constraints in $(\mathbf{P}1)^{A,R}$, the Lagrangian function of $(\mathbf{P}1)^{A,R}$ can be written as

$$\begin{aligned}
L(\theta, \mathbf{V}, \rho, \mu, \delta) = \; & \theta_1 + \sum_i V_{1,i} c_i - \sum_{t,\mathbf{x} \in \mathcal{X}_t} \mu_{t,\mathbf{x}} \left[\theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i}) x_i - \sum_j \frac{\beta_{t,j} w_j(\mathbf{x})}{4} \rho_{t,j}^2 \right] \\
& - \sum_{t,j} \delta_{t,j} \left(\rho_{t,j} - \frac{\alpha_{t,j}}{\beta_{t,j}} + \sum_i a_{ij} V_{t+1,i} \right) \\
= \; & (1 - \mu_{1,\mathbf{c}}) \theta_1 + \sum_{t=2}^{T} \sum_{\mathbf{x} \in \mathcal{X}_t} (\mu_{t-1,\mathbf{x}} - \mu_{t,\mathbf{x}}) \theta_t \\
& + \sum_i c_i \left(1 - \mu_{1,\mathbf{c}}\right) V_{1,i} + \sum_{t=2}^{T} \sum_i \left[\sum_{\mathbf{x} \in \mathcal{X}_t} (\mu_{t-1,\mathbf{x}} - \mu_{t,\mathbf{x}}) x_i - \sum_j \delta_{t-1,j} a_{ij} \right] V_{t,i} \\
& + \sum_{t,j} \frac{\beta_{t,j} \sum_{\mathbf{x} \in \mathcal{X}_t} \mu_{t,\mathbf{x}} w_j(\mathbf{x})}{4} \rho_{t,j}^2 - \sum_{t,j} \delta_{t,j} \rho_{t,j} + \sum_{t,j} \delta_{t,j} \frac{\alpha_{t,j}}{\beta_{t,j}}.
\end{aligned}$$

To arrive at the second equality, we used $\theta_{T+1} = 0$, $V_{T+1,i} = 0$ and $\mathbf{x} = \mathbf{c}$ in period 1. Let

$$g(\mu, \delta) = \min_{\theta, \mathbf{V}, \rho \geq 0} \; L(\theta, \mathbf{V}, \rho, \mu, \delta). \tag{15}$$

The tightest Lagrangian bound can be obtained by maximizing over the set of Lagrangian multipliers (μ, δ); i.e.,

$$z^L = \max_{\mu \geq 0, \delta \geq 0} g(\mu, \delta).$$

Notice that $L(\theta, \mathbf{V}, \rho, \mu, \delta)$ is linear in θ and $\mathbf{V}$. We must have

$$
\begin{aligned}
&1-\mu_{1,\mathbf{c}}=0, && && (16)\\
&c_i(1-\mu_{1,\mathbf{c}})=0, && \forall\ i, && (17)\\
&\sum_{\mathbf{x}}(\mu_{t-1,\mathbf{x}}-\mu_{t,\mathbf{x}})=0, && \forall\ t\geq 2, &&\\
&\sum_{\mathbf{x}}(\mu_{t-1,\mathbf{x}}-\mu_{t,\mathbf{x}})x_i-\sum_j \delta_{t-1,j}a_{ij}=0, && \forall\ i,t\geq 2. &&
\end{aligned}
$$

In the above, equation (17) is redundant because $c_i>0$. It follows that

$$z^L=\max_{\mu\geq,\delta\geq 0}\min_{\rho\geq 0}\sum_{t,j}\frac{\beta_{t,j}\sum_{\mathbf{x}\in\mathcal{X}_t}\mu_{t,\mathbf{x}}w_j(\mathbf{x})}{4}\rho_{t,j}^2-\sum_{t,j}\delta_{t,j}\rho_{t,j}+\sum_{t,j}\delta_{t,j}\frac{\alpha_{t,j}}{\beta_{t,j}}.$$

For each t, define the set $Q_t(\mu)=\{j:\sum_{\mathbf{x}\in\mathcal{X}_t}\mu_{t,\mathbf{x}}w_j(\mathbf{x})\neq 0\}$. We have

$$
\begin{aligned}
z^L=\max_{\mu\geq,\delta\geq 0}\min_{\rho\geq 0}\sum_{t,j\in Q_t(\mu)}&\left(\frac{\beta_{t,j}\sum_{\mathbf{x}\in\mathcal{X}_t}\mu_{t,\mathbf{x}}w_j(\mathbf{x})}{4}\left[\rho_{t,j}-\frac{2\delta_{t,j}}{\beta_{t,j}\sum_{\mathbf{x}\in\mathcal{X}_t}\mu_{t,\mathbf{x}}w_j(\mathbf{x})}\right]^2-\frac{\delta_{t,j}^2}{\beta_{t,j}\sum_{\mathbf{x}\in\mathcal{X}_t}\mu_{t,\mathbf{x}}w_j(\mathbf{x})}\right)\\
&-\sum_{t,j\notin Q_t(\mu)}\delta_{t,j}\rho_{t,j}+\sum_{t,j}\delta_{t,j}\frac{\alpha_{t,j}}{\beta_{t,j}}.
\end{aligned}
$$

Since $\rho_{t,j}\geq 0$ for all t and j in $(\mathbf{P}1)^{A,R}$, the first term can be minimized at

$$\rho_{t,j}=\frac{2\delta_{t,j}}{\beta_{t,j}\sum_{\mathbf{x}\in\mathcal{X}_t}\mu_{t,\mathbf{x}}w_j(\mathbf{x})},\qquad \forall\ t,j\in Q_t(\mu).$$

Also, to ensure boundedness, we must have

$$\delta_{t,j}=0,\qquad \forall\ t,j\notin Q_t(\mu).$$

Therefore,

$$g(\mu,\delta)=\sum_{t,j}\delta_{t,j}\frac{\alpha_{t,j}}{\beta_{t,j}}-\sum_{t,j\in Q_t(\mu)}\frac{\delta_{t,j}^2}{\beta_{t,j}\sum_{\mathbf{x}\in\mathcal{X}_t}\mu_{t,\mathbf{x}}w_j(\mathbf{x})}.$$

Summarizing the above discussion, the dual program (15) can be written as

$$
\begin{aligned}
(\mathbf{D})^{A,R}\qquad &\max_{\mu,\delta}\quad \sum_{t,j}\delta_{t,j}\frac{\alpha_{t,j}}{\beta_{t,j}}-\sum_{t,j\in Q_t(\mu)}\frac{\delta_{t,j}^2}{\beta_{t,j}\sum_{\mathbf{x}\in\mathcal{X}_t}\mu_{t,\mathbf{x}}w_j(\mathbf{x})}\\
\text{s.t.}\qquad &1-\mu_{1,\mathbf{c}}=0, && && (18)\\
&\sum_{\mathbf{x}\in\mathcal{X}_t}(\mu_{t-1,\mathbf{x}}-\mu_{t,\mathbf{x}})=0, && \forall\ t\geq 2, && (19)\\
&\sum_{\mathbf{x}\in\mathcal{X}_t}(\mu_{t-1,\mathbf{x}}-\mu_{t,\mathbf{x}})x_i-\sum_j\delta_{t-1,j}a_{ij}=0, && \forall\ t\geq 2,i, &&\\
&\delta_{t,j}=0, && \forall\ t,j\notin Q_t(\mu), && (20)
\end{aligned}
$$

$$\mu, \delta \geq 0.$$

Slater's Theorem states that strong duality holds for the convex problem if Slater's Condition holds, namely, there exists a strictly feasible point (Section 5.2.3, Boyd and Vandenberghe 2004). To apply this result to our setting, first note that $(\mathbf{P}1)^{A,R}$ is a convex problem and has a bounded optimal value due to the nonnegativity of θ and $\mathbf{V}$. Furthermore, the solution

$$\begin{aligned}
\theta_t &= \sum_{t'=t}^{T} \sum_j \left(\frac{\beta_{t',j}(\alpha_{t',j}/\beta_{t',j}+1)^2}{4} + 1 \right), && \forall t,\\
V_{t,i} &= V_{t+1,i}, && \forall\ t,i,\\
\rho_{t,j} &= \alpha_{t,j}/\beta_{t,j}+1, && \forall\ t,j
\end{aligned}$$

is strictly feasible for $(\mathbf{P}1)^{A,R}$. Thus strong duality holds for $(\mathbf{P}1)^{A,R}$-$(\mathbf{D})^{A,R}$.

Combining constraints (18) and (19), we get

$$\sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t.\mathbf{x}} = 1, \qquad \forall t.$$

Introducing a new variable $\mathbf{s}$, $(\mathbf{D})^{A,R}$ can be written as

$$\begin{aligned}
(\mathbf{D}1)^{A,R} \qquad & \max_{\delta,\mu,\mathbf{s}} \quad \sum_{t,j} \left(\frac{\alpha_{t,j}}{\beta_{t,j}} \delta_{t,j} - \frac{1}{\beta_{t,j}} s_{t,j} \right) \\
\text{s.t.} \qquad & \delta_{t,j}^2 \leq s_{t,j} \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} w_j(\mathbf{x}), && \forall\ t,j, \qquad (21)\\
& \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} x_i = \begin{cases} c_i, & \text{if } t=1,\\ \sum_{\mathbf{x}\in\mathcal{X}_{t-1}} \mu_{t-1,\mathbf{x}} x_i - \sum_j a_{ij}\delta_{t-1,j}, & \text{if } t\geq 2, \end{cases} && \forall\ t,i,\\
& \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} = 1, && \forall\ t,\\
& \delta, \mu,\ \mathbf{s} \geq 0.
\end{aligned}$$

Note that we write (21) as an inequality constraint instead of an equality constraint. This can be done due to the structure of the problem. Constraint (21) is in general not convex, but can be rewritten as a convex constraint (Alizadeh and Goldfarb 2003).

It is also possible to derive $(\mathbf{D}1)^{A,R}$ using SOCP duality, instead of the direct application of Lagrangian conditions. We include the alternative proof in Appendix D. However, applying SOCP duality seems to be more cumbersome. For this reason, we include the analysis based on Lagrangian conditions in the main text.

5.2 Aggregation and Equivalence

Because the variable μ is indexed by $\mathbf{x}$, $(\mathbf{D}1)^{A,R}$ still has a large number of decision variables. In this section, we first use variable aggregation in $(\mathbf{D}1)^{A,R}$ to produce a more compact formulation. Since the compact formulation is the result of variable aggregation, it gives an upper bound on the objective value of $(\mathbf{D}1)^{A,R}$. We will show, however, the aggregate formulation is equivalent to $(\mathbf{D}1)^{A,R}$ in the sense that its objective value is exactly the same as that of $(\mathbf{D}1)^{A,R}$.

To that end, let

$$y_{t,i} = \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} x_i, \qquad \forall t, i, \tag{22}$$

$$z_{t,j} = \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} w_j(\mathbf{x}) = \sum_{\mathbf{x}\in\mathcal{X}_t:\mathbf{x}\geq A^j} \mu_{t,\mathbf{x}}, \qquad \forall t, j. \tag{23}$$

Since $\sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} = 1$ for all t,

$$0 \leq z_{t,j} \leq 1, \qquad \forall\ t, j.$$

Furthermore, since $x_i \geq a_{ij} w_j(\mathbf{x})$ for all i, j, t,

$$y_{t,i} \geq a_{ij} z_{t,j}, \qquad \forall t, i.$$

Putting everything together, applying variable aggregation to $(\mathbf{D}1)^{A,R}$ gives

$$
\begin{aligned}
(\mathbf{D}2)^{A,R} \qquad & \max_{\delta,\mathbf{s},\mathbf{y},\mathbf{z}} \quad \sum_{t,j} \left(\frac{\alpha_{t,j}}{\beta_{t,j}} \delta_{t,j} - \frac{1}{\beta_{t,j}} s_{t,j} \right) \\
\text{s.t.} \qquad & \delta_{t,j}^2 - s_{t,j} z_{t,j} \leq 0, && \forall\ t, j, \qquad (24) \\
& y_{t,i} = \begin{cases} c_i, & \text{if } t = 1, \\ y_{t-1,i} - \sum_j a_{ij} \delta_{t-1,j}, & \text{if } t \geq 2, \end{cases} && \forall\ t, i, \qquad (25) \\
& y_{t,i} \geq a_{ij} z_{t,j}, && \forall\ t, i, j, \qquad (26) \\
& 0 \leq z_{t,j} \leq 1, && \forall\ t, j. \\
& \delta,\ \mathbf{y},\ \mathbf{s} \geq 0.
\end{aligned}
$$

The following proposition shows that $(\mathbf{D}1)^{A,R}$ and $(\mathbf{D}2)^{A,R}$ are equivalent.

Proposition 3. *$(\mathbf{D}1)^{A,R}$ and $(\mathbf{D}2)^{A,R}$ are equivalent.*

Proof: $(\mathbf{D}1)^{A,R}$ can be written as

$$\max_{\delta\geq 0,\mathbf{s}\geq 0} \quad f_0(\delta, \mathbf{s}),$$

where

$$\begin{aligned} f_0(\delta,\mathbf{s}) = \quad & \max_{\mu} \quad \sum_{t,j}\left(\frac{\alpha_{t,j}}{\beta_{t,j}}\delta_{t,j} - \frac{1}{\beta_{t,j}}s_{t,j}\right) \\ \text{s.t.} \quad & \delta_{t,j}^2 \le s_{t,j} \sum_{\mathbf{x}\in\mathcal{X}_t:\mathbf{x}\ge A^j} \mu_{t,\mathbf{x}}, && \forall\, t,j \\ & \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} x_i = \begin{cases} c_i, & \text{if } t=1, \\ \sum_{\mathbf{x}\in\mathcal{X}_{t-1}} \mu_{t-1,\mathbf{x}} x_i - \sum_j a_{ij}\delta_{t-1,j}, & \text{if } t\ge 2, \end{cases} && \forall\, t,i, \\ & \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} = 1, && \forall\, t, \\ & \mu \ge 0. \end{aligned}$$

Similarly, $(\mathbf{D}2)^{A,R}$ can be written as

$$\max_{\delta\ge 0,\mathbf{s}\ge 0} \quad f_1(\delta,\mathbf{s})$$

where

$$\begin{aligned} f_1(\delta,\mathbf{s}) = \quad & \max_{\mathbf{y},\mathbf{z}} \quad \sum_{t,j}\left(\frac{\alpha_{t,j}}{\beta_{t,j}}\delta_{t,j} - \frac{1}{\beta_{t,j}}s_{t,j}\right) \\ \text{s.t.} \quad & \delta_{t,j}^2 \le s_{t,j} z_{t,j}, && \forall\, t,j, && (27) \\ & y_{t,i} = \begin{cases} c_i, & \text{if } t=1, \\ y_{t-1,i} - \sum_j a_{ij}\delta_{t-1,j}, & \text{if } t\ge 2, \end{cases} && \forall\, t,i, && (28) \\ & y_{t,i} \ge a_{ij} z_{t,j}, && \forall\, t,i,j, && (29) \\ & 0 \le z_{t,j} \le 1, && \forall\, t,j. && (30) \end{aligned}$$

Note that both $f_0(\delta,\mathbf{s})$ and $f_1(\delta,\mathbf{s})$ are optimal values of linear programs. Given the same $(\delta,\mathbf{s})$, $f_0(\delta,\mathbf{s}) = f_1(\delta,\mathbf{s})$. On one hand, since $f_1(\delta,\mathbf{s})$ is the variable aggregation program of $f_0(\delta,\mathbf{s})$, then any solution to $f_0(\delta,\mathbf{s})$ has a corresponding solution to $f_1(\delta,\mathbf{s})$. On the other hand, if we can prove any solution to $f_1(\delta,\mathbf{s})$ gives a solution to $f_0(\delta,\mathbf{s})$. It follows that $(\mathbf{D}1)^{A,R}$ and $(\mathbf{D}2)^{A,R}$ are equivalent.

In the program associated with $f_1(\delta,\mathbf{s})$, for each fixed t, let $\mathcal{P}_t = \{(\mathbf{y},\mathbf{z}) \in \mathbb{R}_+^{m+n} : (29)(30), y_{t,i} \le c_i\}$. It is obvious that $\mathcal{P}_t$ has integer extreme points because the constraint matrix is totally unimodular. Let L_t denote the index set of the extreme points for $\mathcal{P}_t$. Then, using the principle of Dantzig-Wolfe reformulation, the program associated with $f_1(\delta,\mathbf{s})$ can be reformulated as

$$f_1'(\delta,\mathbf{s}) = \quad \max_{\gamma} \quad \sum_{t,j}\left(\frac{\alpha_{t,j}}{\beta_{t,j}}\delta_{t,j} - \frac{1}{\beta_{t,j}}s_{t,j}\right)$$

$$
\begin{aligned}
\text{s.t.} \quad & \delta_{t,j}^2 \le s_{t,j} \sum_{e \in L_t} \gamma_e z_{t,j}^e, && \forall\ t, j,\\
& \sum_{e\in L_t} \gamma_e y_{t,i}^e = \begin{cases} c_i, & \text{if } t=1,\\ \sum_{e\in L_{t-1}} \gamma_e y_{t-1,i}^e - \sum_j a_{ij}\delta_{t-1,j}, & \text{if } t \ge 2, \end{cases} && \forall\ t, i,\\
& \sum_{e\in L_t} \gamma_e = 1, && \forall\ t,\\
& \gamma \ge 0.
\end{aligned}
$$

Each extreme point (y_t^e, z_t^e) of $\mathcal{P}_t$ is integral and $y_t^e \in \mathcal{X}_t$ from the definition of $\mathcal{P}_t$. In order to reproduce the program associated with $f_0(\delta, \mathbf{s})$, take

$$\mu_{t,\mathbf{x}} = \sum_{e\in L_t : \mathbf{y}_t^e = \mathbf{x}} \gamma_e, \qquad \forall t, \mathbf{x}.$$

It follows that

$$\mu_{t,\mathbf{x}} x_i = \sum_{e\in L_t : \mathbf{y}_t^e = \mathbf{x}} \gamma_e y_{t,i}^e, \qquad \forall t, \mathbf{x}, i.$$

For each t and j, we also have

$$\sum_{x\in\mathcal{X}_t : \mathbf{x}\ge A^j} \mu_{t,\mathbf{x}} = \sum_{x\in\mathcal{X}_t : \mathbf{x}\ge A^j} \sum_{e\in L_t : \mathbf{y}_t^e = \mathbf{x}} \gamma_e = \sum_{e\in L_t : \mathbf{y}_t^e \ge A^j} \gamma_e = \sum_{e\in L_t : z_{t,j}^e = 1} \gamma_e = \sum_{e\in L_t} \gamma_e z_{t,j}^e.$$

In other words, $f_0(\delta, \mathbf{s})$ can be produced from the problem associated with $f_1'(\delta, \mathbf{s})$ via variable aggregation. Thus any solution to $f_1'(\delta, \mathbf{s})$ has a corresponding solution to $f_0(\delta, \mathbf{s})$. This completes the proof. ■

The program $(\mathbf{D}2)^{A,R}$ admits an intuitive explanation. Since μ in $(\mathbf{D}1)^{A,R}$ is nonnegative and sums to one for each t, it can be interpreted as the probability of state $\mathbf{x}$ in period t. From (22), the variable $y_{t,i}$ can be interpreted as the expected resource level in period t for resource i. Similarly, from (23), $z_{t,j}$ is the probability that $\mathbf{x} \ge A^j$ in period t. $\delta_{t,j}$ can be seen as the expected sales for product j in period t. Since sales only exists in the situation $x \ge A^j$ when $z_{t,j} \ne 0$, the real demand is $\frac{\delta_{t,j}}{z_{t,j}}$ and the corresponding price is $(\alpha_{t,j} - \frac{\delta_{t,j}}{z_{t,j}})/\beta_{t,j}$ from demand function (8). When $z_{t,j} = \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} w_j(\mathbf{x}) = 0$, $\delta_{t,j} = 0$ from constraint (21). Then the revenue in period t is

$$\sum_{j:z_{t,j}\ne 0} \delta_{t,j} \frac{\alpha_{t,j} - \frac{\delta_{t,j}}{z_{t,j}}}{\beta_{t,j}} + \sum_{j:z_{t,j}=0} 0 = \sum_{t,j} \frac{\alpha_{t,j}}{\beta_{t,j}} \delta_{t,j} - \sum_{t,j:z_{t,j}\ne 0} \frac{\delta_{t,j}^2}{\beta_{t,j} z_{t,j}},$$

which interprets the objective function and constraint (24). $\sum_j a_{ij}\delta_{t-1,j}$ in constraint (25) can be interpreted as resource i consumed in period $t-1$ and the resource level $y_{t-1,i}$ is updated.

The quadratic constraint (24) in $(\mathbf{D}2)^{A,R}$ can be written as the following equivalent second order cone constraint

$$\left\| \begin{pmatrix} 2\delta_{t,j} \\ s_{t,j} - z_{t,j} \end{pmatrix} \right\| \le s_{t,j} + z_{t,j}, \qquad \forall t, j,$$

where $\|\cdot\|$ denotes the usual Euclidean norm. With the alternative representation of constraint (24), $(\mathbf{D}2)^{A,R}$ is a SOCP.

It is also instructional to consider the dual of $(\mathbf{D}2)^{A,R}$, which can be stated as

$$
\begin{aligned}
(\mathbf{P}2)^{A,R} \qquad & \min_{\psi,\mathbf{V},\omega,\zeta} \sum_i c_i V_{1,i} + \sum_{t,j} \zeta_{t,j} & \\
\text{s.t.} \qquad & 2\psi_{2,t,j} + \sum_i a_{ij} V_{t+1,i} \geq \frac{\alpha_{t,j}}{\beta_{t,j}}, & \forall t,j, \\
& \psi_{1,t,j} - \psi_{3,t,j} \leq \frac{1}{\beta_{t,j}}, & \forall t,j, \\
& V_{t,i} - V_{t+1,i} - \sum_j \omega_{t,i,j} \geq 0, & \forall t,i, \\
& \psi_{1,t,j} + \psi_{3,t,j} - \sum_i a_{ij}\omega_{t,i,j} - \zeta_{t,j} \leq 0, & \forall t,j, \\
& \left\| \begin{pmatrix} \psi_{2,t,j} \\ \psi_{3,t,j} \end{pmatrix} \right\| \leq \psi_{1,t,j}, & \forall t,j, \\
& \omega, \zeta \geq 0. &
\end{aligned}
$$

In the above, we again adopt the notational convention that $V_{T+1,i} = 0$ for all i. In our numerical study, we show that for a particular commercial solver, solving $(\mathbf{P}2)^{A,R}$ is sometimes quicker than solving $(\mathbf{D}2)^{A,R}$.

Before proceeding, we comment here that it is possible to discretize prices as a numerical strategy. Section 6 considers a discretized version of the problem and compares it with the formulation presented in this section.

5.3 Deterministic Nonlinear Programming Formulation and Its Connection to $(\mathbf{D}2)^{A,R}$

A useful computational benchmark for the dynamic pricing problem we discussed so far is the deterministic formulation that ignores demand uncertainty. Although we can formulate the deterministic problem for general demand functions, we focus again on the case with linear independent demand. The inverse demand function is given by

$$r_{t,j}(\lambda_{t,j}) = \frac{\alpha_{t,j} - \lambda_{t,j}}{\beta_{t,j}}, \qquad \forall t,j.$$

The deterministic program can be formulated as

$$
\begin{aligned}
(\mathbf{DeP}) \qquad & z^d = \max_{\lambda} \quad \sum_{t,j} \lambda_{t,j} \left(\frac{\alpha_{t,j} - \lambda_{t,j}}{\beta_{t,j}}\right), & \\
\text{s.t.} \qquad & \sum_{t,j} a_{ij}\lambda_{t,j} \leq c_i, & \forall\, i, \qquad (31)
\end{aligned}
$$

$$\lambda_{t,j} \geq 0, \qquad\qquad \forall\ t.$$

The program (**DeP**) is a nonlinear program because the objective function is quadratic. Constraints (31) are resource constraints, the dual variables of which can be interpreted as the marginal values of resources. We have the following result, for which we provide a proof in Appendix E.

PROPOSITION 4. $z^A \leq z^d \leq (1 + \frac{1}{\min_i c_i}) z^A$.

Since z^A is an upper bound on the optimal revenue, the first inequality in Proposition 4 immediately implies that the deterministic formulation offers an upper bound on the optimal revenue as well. This relationship between the deterministic formulation and the optimal revenue is a well known result in the literature for quite general demand functions (Gallego and van Ryzin 1997), for which Proposition 4 offers an alternative proof for the case of linear independent demand. The second inequality in Proposition 4 shows that the gap between the objective value of the deterministic formulation and that of the affine ALP is bounded by a factor inversely related to the minimum capacity on the network. This result parallels that of Kunnumkal and Talluri (2016), where they consider a capacity control problem under customer choice. Indeed, the bounding factor in Proposition 4 is the same as the one reported in Kunnumkal and Talluri (2016). This is another evidence that much of the results for capacity control problems in the existing ADP literature might be generalized to other settings. Future research could investigate to what extent these results for the capacity control problems can be generalized. We view our work as a small yet meaningful step in that direction.

6. Discretization as an Alternative Solution Strategy

So far, we have focused on the formulation with continuous prices. Under the linear independent model, the revenue function is a second order polynomial. Therefore, the optimization problem is nonlinear.

An alternative strategy is to formulate the problem with discrete price points. That is, the firm offers a price menu for each product. Customers are segmented based on prices, and will arrive with price-dependent probabilities. The firm will only offer one price for each product in each period. This setting is close to capacity control problems, in which customers belong to different classes with a predetermined price, and the firm decides which type of customers to accept. As discussed in the revenue management literature (Gallego and van Ryzin 1997), the pricing problems with a discrete set of prices and capacity control problems are intimately connected. It is reasonable to expect that theoretical results on the reduction of ALPs for the capacity control problem (Tong and Topaloglu 2013, Vossen and Zhang 2015) can be extended to the pricing problem with a discrete set of prices. In this section, we confirm that this is indeed the case by establishing compact ALPs for the formulation with discrete price sets.

6.1 The Formulation with a Discrete Price Set

For each product j, we consider a discrete price set with K price points in period t. Let the price set be $\{r_{t,j,0}, r_{t,j,1}, ..., r_{t,j,K-1}\}$ with corresponding arrival rates in the set $\{\lambda_{t,j,0}, \lambda_{t,j,1}, ..., \lambda_{t,j,K-1}\}$. Without loss of generality, we assume that $r_{t,j,1} > ... > r_{t,j,K-1}$ and $\lambda_{t,j,1} < ... < \lambda_{t,j,K-1}$. In addition, we include a null price $r_{t,j,0}$ with corresponding demand $\lambda_{t,j,0} = 0$. We require $\sum_j \lambda_{t,j,K-1} \leq 1$ for all t so that there is at most one customer arrives in each period.

The dynamic programming model with discrete price sets can be similarly formulated as in the continuous price case. Let $\hat{v}_t(\cdot)$ denote the value function in period t. The decision is the price point (including the null price) to be offered for each product. Let $u_{t,j.k} \in \{0, 1\}$ denote whether price $r_{t,j,k}$ is offered in period t for product j. Since only one price can be offered for each product, we must have

$$\sum_k u_{t,j,k} = 1, \quad \forall t, j.$$

Because of the capacity constraint, the null price must be offered when there is not sufficient capacity to offer a product. This constraint can be written as

$$u_{t,j,0} \geq a_{ij}(1 - x_i), \qquad \forall t, i, j.$$

Let $\hat{\mathcal{U}}_t(\mathbf{x})$ denote the feasible action set for state $\mathbf{x}$ in period t. Then the dynamic program in (1) becomes

$$\begin{aligned}
\hat{v}_t(\mathbf{x}) &= \max_{\mathbf{u}_t \in \hat{\mathcal{U}}_t(\mathbf{x})} \left\{ \sum_{j,k} u_{t,j,k}\lambda_{t,j,k} \left(r_{t,j,k} + \hat{v}_{t+1}(\mathbf{x} - A^j) \right) + \left(1 - \sum_{j,k} u_{t,j,k}\lambda_{t,j,k} \right) \hat{v}_{t+1}(\mathbf{x}) \right\} \\
&= \max_{\mathbf{u}_t \in \hat{\mathcal{U}}_t(\mathbf{x})} \left\{ \sum_{j,k} u_{t,j,k}\lambda_{t,j,k} \left(r_{t,j,k} + \hat{v}_{t+1}(\mathbf{x} - A^j) - \hat{v}_{t+1}(\mathbf{x}) \right) \right\} + \hat{v}_{t+1}(\mathbf{x}). \qquad (32)
\end{aligned}$$

Under the affine approximation, the approximate linear program of (32) can be written as

$$(\mathbf{P})_D^A \quad z_D^A = \min_{\theta, \mathbf{V}} \quad \theta_1 + \sum_i V_{1,i} c_i$$

$$\text{s.t.} \quad \theta_t - \theta_{t+1} + \sum_i \left(V_{t,i}x_i - \left(x_i - \sum_{j,k} u_{t,j,k}\lambda_{t,j,k}a_{ij} \right) V_{t+1,i} \right) \geq \sum_{j,k} u_{t,j,k}\lambda_{t,j,k}r_{t,j,k}, \quad \forall\ t, \mathbf{x}, \mathbf{u}_t \in \hat{\mathcal{U}}_t(\mathbf{x}).$$

We also adopt the notational convention that $\theta_{T+1} = 0$ and $V_{T+1,i} = 0$ for all i.

6.2 A Compact Formulation

Column generation can be applied to solve the dual to $(\mathbf{P})_D^A$, which we label as $(\mathbf{D})_D^A$. For a given primal solution $(\theta, \mathbf{V})$, the column generation subproblem for each period t is

$$
\begin{aligned}
(\mathbf{CG})_D^A \quad & \min_{\mathbf{x},\mathbf{u}_t} \quad \theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i}) x_i - \sum_{j,k} u_{t,j,k} \lambda_{t,j,k} \left(r_{t,j,k} - \sum_i a_{ij} V_{t+1,i} \right), \\
& \sum_k u_{t,j,k} = 1, && \forall j, \quad (33) \\
& u_{t,j,0} \geq a_{ij}(1 - x_i), && \forall\, i, j, \quad (34) \\
& \mathbf{x}, \mathbf{u}_t \text{ binary.} && (35)
\end{aligned}
$$

Observe that for each product j, only the null price or the revenue maximizing price corresponding to

$$
k_j^* \in \arg\max_k \lambda_{j,k} \left(r_{j,k} - \sum_i a_{ij} V_{t+1,i} \right)
$$

can be optimal. Therefore, the coefficient vector in constraint (33) can be reduced from a K-vector with all elements being 1 to a K-vector with only two elements being 1. Here we assume ties are broken arbitrarily. Thus the transpose of the constraint matrix has at most two nonzero entries in each column and the rows can be partitioned into two disjoint sets with nonzero entries in each set, where one set only have N rows correspondent to the coefficient of $u_{t,j,0}, \forall j$. The property of total unimodularity holds for the coefficient matrix of constraints (33)–(34) and the integer program $(\mathbf{CG})_D^A$ can be relaxed to an equivalent linear program by replacing constraints (35) with constraint $\mathbf{x}, \mathbf{u}_t \geq 0$.

Following a similar line of analysis in Vossen and Zhang (2015), $(\mathbf{D})_D^A$ admits the following compact representation:

$$
\begin{aligned}
(\mathbf{D2})^{A,D} \qquad & z_D^A = \max_{\mathbf{q},\mathbf{y}} \sum_{t,j,k} \lambda_{t,j,k} r_{t,j,k} q_{t,j,k} \\
\text{s.t.} \qquad & y_{t,i} = \begin{cases} c_i, & \text{if } t = 1, \\ y_{t-1,i} - \sum_{j,k} a_{ij} \lambda_{t-1,j,k} q_{t-1,j,k}, & \text{if } t \geq 2, \end{cases} && \forall i, \\
& a_{ij} y_{t,i} + q_{t,j,0} \geq a_{ij}, && \forall t, i, j, \\
& \sum_k q_{t,j,k} = 1, && \forall t, j, \\
& \mathbf{q}, \mathbf{y} \geq 0.
\end{aligned}
$$

The program $(\mathbf{D}2)^{A,D}$ is a linear program and the number of constraints is linear in the number of resource m, the number of products n, and the number of periods T. However, when the number of price points K increases, the number of variables $\mathbf{q}$ grows linearly in Tn, which can be very large; as a result, it can be computationally challenging to solve $(\mathbf{D}2)^{A,D}$. In our numerical study, we show that for a relatively small set of price points, solving $(\mathbf{D}2)^{A,D}$ can be faster than solving the continuous formulation, possibly with some solution quality degradation. While for relatively large sets of price points, solving $(\mathbf{D}2)^{A,D}$ is slow, or even infeasible, within a reasonable time frame.

7. Pricing Policies

In this section, we introduce the pricing policies we consider in our simulation study.

First, it is natural to use the dual values ω^* for the resource constraints (31) in (**DeP**) as static bid-prices. Note that since we are considering a pricing problem, classical bid-price control (Talluri and van Ryzin 1998) does not apply in our setting. Instead, the bid prices are used as approximations of marginal values in equation (1). That is, the price $\mathbf{r}_t$ is determined by solving (1) where the opportunity cost for accepting a class-j customer in period t, $v_{t+1}(\mathbf{x}) - v_{t+1}(\mathbf{x} - A^j)$, is approximated by $\sum_i a_{ij}\omega_i^*$. This policy is called static bid policy **SB** later in the paper.

Similarly, we can use the solution from the affine approximation to construct a bid-price policy. Let $(\theta^*, \mathbf{V}^*)$ be an optimal solution for the affine approximation. We can approximate $v_{t+1}(\mathbf{x}) - v_{t+1}(\mathbf{x} - A^j)$ by $\sum_i a_{ij} V^*_{t+1,i}$. This policy is called the dynamic bid-price policy (**DB** hereafter).

In addition, we also consider dynamic programming decomposition methods based on the solution to (**DeP**) and the affine approximation. In the NRM context, dynamic programming decomposition methods were considered by Liu and van Ryzin (2008), Miranda Bront et al. (2009), Zhang and Adelman (2009), and Zhang (2011), among others. Here, we adapt these methods to the dynamic pricing setting. The version that considers dynamic bid-prices is called **DBD**, while the one that considers static bid-prices is called **SBD**.

7.1 The Policy DBD

The policy **DBD** is based on the optimal affine solutions $(\theta^*, \mathbf{V}^*)$. For each i, consider the approximation

$$v_t(\mathbf{x}) \approx v_{t,i}(x_i) + \sum_{k \neq i} V^*_{t,k} x_k, \quad \forall t, x. \tag{36}$$

Using the above approximation in $(\mathbf{P})^{\phi}$ under affine approximation, we obtain

$$\begin{aligned}
(\mathbf{NLP}_i) \qquad & \min_{v_{t,i}(\cdot)} \quad v_{1,i}(c_i) + \sum_{k\neq i} V^*_{1,k} c_k \\
\text{s.t.} \qquad & v_{t,i}(x_i) + \sum_{k\neq i} V^*_{t,k} x_k \geq \sum_j \lambda_{t,j}(\mathbf{r}_t) \left(r_{t,j} + v_{t+1,i}(x_i - a_{ij}) + \sum_{k\neq i} V^*_{t+1,k}(x_k - a_{kj}) \right) \\
& + \left(1 - \sum_j \lambda_{t,j}(\mathbf{r}_t)\right) \left(v_{t+1,i}(x_i) + \sum_{k \neq i} V^*_{t+1,k} x_k \right), \ \forall\ t, \mathbf{x} \in \mathcal{X}_t, \mathbf{r}_t \in \mathcal{R}_t(\mathbf{x}).
\end{aligned}$$

The proof of the following results is analogous to the one for Proposition 2 in Zhang and Adelman (2009) and is therefore omitted.

Proposition 5. *For each i, let $v_{t,i}(\cdot)$ and $\widehat{v}_{t,i}(\cdot)$ be a feasible solution and an optimal solution to* $(\mathbf{NLP}_i)$*, respectively. We have*

$$(i) \quad \min_i \left\{ v_{t,i}(x_i) + \sum_{k\neq i} V^*_{t,k} x_k \right\} \geq v_t(\mathbf{x});$$

$$(ii) \quad z^d \geq z^A \geq \max_i \left\{ \widehat{v}_{1,i}(x_i) + \sum_{k\neq i} V^*_{1,k} x_k \right\} \geq \min_i \left\{ \widehat{v}_{1,i}(x_i) + \sum_{k\neq i} V^*_{1,k} x_k \right\} \geq v_1(\mathbf{c}).$$

Proposition 5 shows some interesting bounds. Part (i) shows that the objective value of $(\mathbf{NLP}_i)$ for each i gives an upper bound to the optimal dynamic programming value function. Part(ii) further shows that the objective value of $(\mathbf{NLP}_i)$ for each i is a tighter upper bound than the affine bound z^A, which in turn is tighter than the bound from the deterministic formulation z^d. From the discussion in Section 6, pricing problems are structurally similarly to capacity control problems; Proposition 5 studied in Zhang and Adelman (2009) naturally holds in our pricing problem. The second to last term in (ii) is called **DBD** bound in our numerical study.

One nice property of $(\mathbf{NLP}_i)$ is that an optimal solution can be obtained by solving the following dynamic programming equations

$$\begin{aligned} \widehat{v}_{t,i}(x_i) = \quad & \max_{\mathbf{x}_{-i},\mathbf{r}_t} \left\{ \sum_j \lambda_{t,j}(\mathbf{r}_t) \left[r_{t,j} + \widehat{v}_{t+1,i}(x_i - a_{ij}) - \widehat{v}_{t+1,i}(x_i) - \sum_{k\neq i} V^*_{t+1,k} a_{kj} \right] - \sum_{k\neq i} (V^*_{t,k} - V^*_{t+1,k}) x_k \right\} \\ & + \widehat{v}_{t+1,i}(x_i), \end{aligned} \tag{37}$$

with boundary conditions $\widehat{v}_{T+1,i}(x_i) = 0 \ \forall x_i, i$. Here $\mathbf{x}_{-i}$ is $\mathbf{x}$ without the i-th component x_i. Refer to Zhang and Adelman (2009) for relevant discussions. In Appendix F, we show that $\Delta \widehat{v}_{t,i}(x_i) = \widehat{v}_{t,i}(x_i) - \widehat{v}_{t,i}(x_i - 1)$ is monotone in both t and x_i.

For linear independent demand, (37) becomes

$$\begin{aligned} \widehat{v}_{t,i}(x_i) = \max_{x_{-i},\mathbf{r}_t} & \left\{ \sum_j (\alpha_{t,j} - \beta_{t,j} r_{t,j}) \left[r_{t,j} + \widehat{v}_{t+1,i}(x_i - a_{ij}) - \widehat{v}_{t+1,i}(x_i) - \sum_{k\neq i} V^*_{t+1,k} a_{kj} \right] \right. \\ & \left. - \sum_{k\neq i} (V^*_{t,k} - V^*_{t+1,k}) x_k \right\} + \widehat{v}_{t+1,i}(x_i). \end{aligned}$$

The maximization problem on the right side has the same structure as the column generation subproblem $(\mathbf{CG})^A_L$ except that x_i is fixed. Therefore, we can use the same solution method to solve it.

After $\{\widehat{v}_{t,i}(\cdot)\}_{\forall t,i}$ is obtained, the price for each product in period t and state $\mathbf{x}$ can be set by solving

$$\max_{\mathbf{r}_t} \left\{ \sum_j (\alpha_{t,j} - \beta_{t,j} r_{t,j}) \left[r_{t,j} - \sum_i \left(\widehat{v}_{t,i}(x_i) - \widehat{v}_{t,i}(x_i - a_{ij}) \right) \right] \right\}.$$

In the above, we approximate the marginal cost of accepting class-j customer with $\sum_i (\widehat{v}_{t,i}(x_i) - \widehat{v}_{t,i}(x_i - a_{ij}))$. Note that this policy is time and inventory dependent and is called **DBD**.

7.2 The Policy SBD

The policy **SBD** is based on the dual values ω^* for the resource constraints in (**DeP**). This policy is similar to **DBD**, except that instead of the approximation (36), we consider the approximation

$$v_t(\mathbf{x}) \approx v_{t,i}(x_i) + \sum_{k \neq i} \omega_k^* x_k, \quad \forall t, x. \tag{38}$$

It is straightforward to show that the analysis in Section 7.1 also works for this approximation with $V_{t,i}^*$ replaced by ω_i^* for each i. For example, the bounds and structural properties in Propositions 5 and 8 still hold. In particular, a bound analogous to **DBD** bound can be shown where $V_{1,i}^*$ is replaced by ω_i^* for each i. This bound is called **SBD** bound in our numerical study. We omit the details of the analysis for brevity. Like **DBD**, **SBD** is also a time and inventory dependent policy.

8. Numerical Study

We conduct a numerical study to investigate the computational performance and compare the performance of the different heuristic policies. We also compare the performance of continuous and discrete formulations.

8.1 Computational Setup

We consider hub-and-spoke networks with one or two hubs as illustrated in Figure 1. Each non-hub location has one connecting flight to the hub and one from the hub. For the one-hub network with S non-hub locations, there are $2S$ resources (flights) and $S^2 + 2S$ products ($2S$ local non-stop itineraries and S^2 connecting itineraries, including round trip between hub and non-hub locations). For the two-hub network with S non-hub locations, there are $2S + 2$ resources and $S^2 + 4S + 3$ products ($2S + 2$ local non-stop itineraries, 1 hub-hub connecting itinerary, $2S$ hub and non-hub locations connecting itineraries, and S^2 non-hub locations connecting itineraries).

We use the load factor defined as follows to measure the demand intensity:

$$\text{load factor} = \frac{\sum_{t,i,j} a_{ij} \lambda_{t,j}(\mathbf{r}_t^*)}{\sum_i c_i},$$

where $\mathbf{r}_t^*$ is the revenue maximizing price under unlimited resources. Therefore, the load factor is the total expected resource consumption under revenue-maximizing prices divided by the total available resource. This definition of load factor is similar to the one used in Adelman (2007),

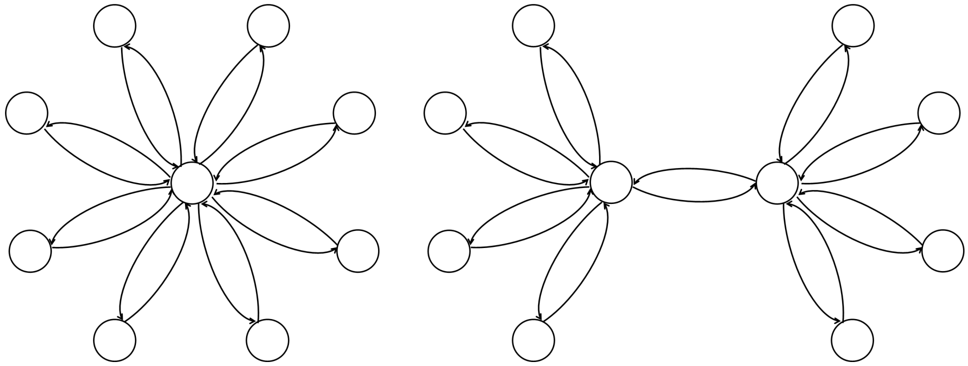

Figure 1 Hub-and-Spoke Network with one hub and two hubs

albeit for capacity control problems. For linear independent demand functions considered in our numerical study, $r^*_{t,j} = \frac{\alpha_{t,j}}{2\beta_{t,j}}$, $\lambda_{t,j}(r^*_{t,j}) = \frac{\alpha_{t,j}}{2}$ and the load factor is $\frac{\sum_{t,i,j} a_{ij}\alpha_{t,j}}{2\sum_i c_i}$.

We systematically change the problem size by varying the number of non-hub locations S in the set $\{4, 6, 8\}$ and the number of periods T in the set $\{100, 200, 400\}$. The number of variables and quadratic constraints for the SOCP formulation increases rapidly as problem size increases. Table 1 shows the problem sizes for different instances. The value of $\alpha_{t,j}$ are randomly generated and sum to 0.75 across all products in each period. The price sensitivity parameter $\beta_{t,j}$ is set to be $\alpha_{t,j}/\epsilon_j$, where ϵ_j is a randomly generated value from the uniform distribution $\mathcal{U}[1, 10]$ for the itineraries from non-hub location to the hub and $\mathcal{U}[10, 100]$ for the itineraries from the hub to non-hub location, respectively. For connecting itineraries, ϵ_j is taken to 0.95 times the sum of the corresponding local itineraries. We choose capacity such that the load factor is between 1 and 2. Each instance is represented with (H, S, T, LF), where H is the number of hubs and LF is the load factor.

8.2 Computational Results for the Continuous Formulation

In this subsection, we report computational results for the compact SOCP model as well as the column generation algorithm for the continuous-price model.

The programs $(\mathbf{D2})^{A,R}$-$(\mathbf{P2})^{A,R}$ are second order cone programming problems, for which there are many different solvers. In our numerical experiments, we tried several different solvers, including Gurobi, CVX (Grant et al. 2008), SDPT3 (Toh et al. 1999), and Knitro. In our initial test of the different solvers, we used MATLAB to call the different solvers using a computer server with Windows Server 2016 Standard 64-bit, Intel Xeon Gold 6140 CPU, and 192 GB RAM. Our test result shows that Gurobi is in general much faster compared with the other three solvers. For this reason, we only report results for Gurobi.

Table 1 Problem sizes for different test instances

Instance	Resources	Products	Periods	Variables	Quadratic Constraints	Linear Constraints
(1,4,200,1.3)	8	24	200	59208	4800	16008
(1,4,400,2.0)	8	24	400	118408	9600	32008
(1,4,800,1.9)	8	24	800	236808	19200	64008
(1,6,200,1.3)	12	48	200	156012	9600	31212
(1,6,400,1.6)	12	48	400	312012	19200	62412
(1,8,200,1.0)	16	80	200	323216	16000	51216
(2,4,200,1.5)	10	35	200	100010	7000	23010
(2,4,400,1.8)	10	35	400	200010	14000	46010
(2,4,800,2.0)	10	35	800	400010	28000	92010
(2,6,200,1.2)	14	63	200	229614	12600	40614
(2,6,400,1.2)	14	63	400	459214	25200	81214
(2,8,200,1.1)	18	99	200	439218	19800	63018

To illustrate the computational advantage of solving SOCP formulation compared with solving the original ALP formulation with column generation, we report the solution time of the column generation algorithm with 0.01% and 1% accuracy in Table 2, which was also implemented with Matlab and Gurobi. Monotonicity property of the affine ALP solution was shown to be important for the computational performance of the column generation algorithms for capacity control problems in Adelman (2007) and Zhang and Adelman (2009). We have shown that such monotonicity property also holds in our setting; that is, the optimal solution (θ, V) to the affine ALP is non-increasing in t. Like Adelman (2007) and Zhang and Adelman (2009), we observed that the column generation algorithm converged more quickly with monotonicity constraints. Therefore, we enforced monotonicity constraints in our numerical experiments. Perhaps unsurprisingly, solving the reduced programs $(\mathbf{D2})^{A,R}$-$(\mathbf{P2})^{A,R}$ is in general orders of magnitude faster than solving the original ALP with column generation for both 0.01% and 1% accuracy. For the same network, as the number of periods increases, the solution time increases nonlinearly.

The column z^d in Table 2 reports the deterministic bound. The second to last column in Table 2 reports the ratio between z^d and the objective value of the affine bound (z^A), and the last column reports the theoretical bound in Proposition 4. Note that the results here are consistent with Proposition 4.

Note that $(\mathbf{D2})^{A,R}$ and $(\mathbf{P2})^{A,R}$ can have substantially different solution time for different problem instances. Such variations in solution time for primal-dual pairs are not uncommon for nonlinear programs in general, and SOCPs in particular. Nevertheless, it would be interesting to understand the drivers of the varying numerical performance for different problem instances. We leave such considerations for future work.

Table 2 Computational comparison between reduced programs and column generation algorithm

Instance	Computational Time					z^d	z^d/z^A	$1+\frac{1}{\min_i c_i}$
	CG-0.01%	CG-1%	$(\mathbf{D2})^{A,R}$	$(\mathbf{P2})^{A,R}$	(**DeP**)			
(1,4,200,1.3)	952.82	325.51	2.44	1.84	0.11	11089.49	1.02	1.08
(1,4,400,2.0)	5097.80	1842.11	13.65	7.90	0.14	16105.55	1.02	1.07
(1,4,800,1.9)	83364.77	26594.65	130.99	55.11	0.13	36102.84	1.01	1.03
(1,6,200,1.3)	2767.74	733.48	24.67	3.36	0.16	11928.67	1.02	1.10
(1,6,400,1.6)	37054.32	8322.76	75.90	31.77	0.16	27231.30	1.02	1.06
(1,8,200,1.0)	2844.65	672.78	41.66	26.64	0.13	13030.09	1.00	1.10
(2,4,200,1.5)	1941.84	678.85	5.93	2.73	0.14	12291.53	1.02	1.10
(2,4,400,1.8)	9937.22	3655.17	61.01	40.33	0.13	20612.11	1.01	1.05
(2,4,800,2.0)	115997.72	34450.58	76.72	112.99	0.17	40908.93	1.01	1.03
(2,6,200,1.2)	1048.84	397.60	41.82	26.16	0.14	12726.02	1.01	1.10
(2,6,400,1.2)	6816.78	2851.85	39.63	138.48	0.09	23167.49	1.01	1.05
(2,8,200,1.1)	628.03	293.14	74.79	58.18	0.35	12530.84	1.01	1.10

8.3 Computational Comparison between Continuous and Discrete Formulations

Since $(\mathbf{D2})^{A,D}$ is a relatively small linear program, a reasonable question to ask is whether solving $(\mathbf{D2})^{A,D}$ is faster than solving the SOCPs we proposed for the continuous formulation. To investigate this question, we compare the computational times and optimal values of $(\mathbf{P2})^{A,R}$ and $(\mathbf{D2})^{A,D}$ for $K \in \{5, 10, 20, 40\}$ using test instances with linear independent demand. For $(\mathbf{D2})^{A,D}$, we use equally spaced intervals between $\frac{\alpha_{t,j}}{2\beta_{t,j}}$ and $\frac{\alpha_{t,j}}{\beta_{t,j}}$ for each product j in period t. Note that all prices between 0 and $\frac{\alpha_{t,j}}{2\beta_{t,j}}$ are weakly dominated as a price above $\frac{\alpha_{t,j}}{2\beta_{t,j}}$ can be chosen to produce the same revenue with lower demand. Therefore, we ignore all prices below $\frac{\alpha_{t,j}}{2\beta_{t,j}}$ in our numerical study.

Table 3 compares the solution times of $(\mathbf{P2})^{A,R}$ and $(\mathbf{D2})^{A,D}$ for different values of K. In general, solving $(\mathbf{D2})^{A,D}$ with $K = 5$ has shorter computational time compared with solving $(\mathbf{P2})^{A,R}$ for most problem instances. However, the objective values are still quite far from the optimal values reported by $(\mathbf{P2})^{A,R}$. As we increase the value of K, the objective values get closer to the optimal values from $(\mathbf{P2})^{A,R}$. However, the solution time is longer than those of $(\mathbf{P2})^{A,R}$ for many problem instances. In addition, for some problem instances, a solution cannot be obtained for $(\mathbf{D2})^{A,D}$ due to an out of memory error; these instances are indicated with a "$-$" in the table.

We comment here that the numerical behavior we observed is not unique. Indeed, all continuous optimization problems can be approximately solved via discretization. But discretization is not always pursued precisely because of the type of numerical behavior observed here. There is a trade-off between solution accuracy and solution time, which does not always favor discretization.

Table 3 Computational times for continuous and discrete formulations

Instance	K=5		K=10		K=20		K=40		$(\mathbf{P2})^{A,R}$	
	z_D^A	time	z_D^A	time	z_D^A	time	z_D^A	time	z^A	time
(1,4,200,1.3)	10805.60	0.74	10865.28	1.74	10876.70	2.69	10879.94	3.88	10877.25	1.84
(1,4,400,2.0)	15668.13	3.07	15751.01	3.41	15775.48	5.48	15778.80	9.28	15777.77	7.90
(1,4,800,1.9)	35442.06	6.33	35726.41	9.36	35766.90	13.63	-	-	35760.88	55.11
(1,6,200,1.3)	11565.26	2.23	11651.28	4.21	11660.62	7.36	11663.20	11.06	11661.49	3.36
(1,6,400,1.6)	26554.15	7.21	26664.26	10.36	26697.42	16.42	-	-	26693.80	31.77
(1,8,200,1.0)	12931.72	2.66	12953.51	7.36	12963.55	7.91	12966.77	13.48	12965.32	26.64
(2,4,200,1.5)	11937.58	1.81	11994.68	2.40	12005.31	3.76	12008.83	6.16	12005.12	2.73
(2,4,400,1.8)	20138.66	4.83	20302.34	4.95	20317.76	9.51	20320.75	15.24	20314.84	40.33
(2,4,800,2.0)	40288.96	9.74	40507.36	12.76	-	-	-	-	40527.64	112.99
(2,6,200,1.2)	12517.99	3.51	12544.61	3.56	12554.51	7.28	12556.72	11.33	12553.99	26.16
(2,6,400,1.2)	22972.52	5.13	23005.02	7.28	-	-	-	-	23017.93	138.48
(2,8,200,1.1)	12329.49	3.12	12351.09	4.27	12356.42	6.96	-	-	12356.38	58.18

8.4 Comparison of Pricing Policies

In this subsection, we compare the bounds and the pricing policies introduced in Section 7 on hub-and-spoke instances illustrated in Figure 1. Each pricing policy is simulated for 400 times using the same demand sequence. We also consider variants of the policies that uses discrete prices. These policies are prefixed with a "d." The policies **dDB** and **dDBD** are the versions of **DB** and **DBD** policies with discrete prices.

Table 4 compares the pricing policies from the deterministic nonlinear program and the affine ALP for the continuous-price case. For each instance, the first row reports the absolute value of policy bounds and policy performance. The second row reports the relative value to the **DBD** bound reported in the second column. We can see that the pricing policy **SBD** and **DBD** offer higher expected revenues than **SB** and **DB**. Therefore, dynamic programming decomposition generates higher revenue in general. The last column in Table 4 reports the confidence interval (CI) on the revenue difference between **DBD** and **SBD**, which shows that **DBD** offers higher revenue than **SBD**. On average, **DBD** improves the revenue by 0.53%. From the relative performance in Table 4, we can see that the difference of the expected revenue between dynamic program decomposition policies and bid price policies is larger than the difference of their bounds. This suggests that dynamic program decomposition should be adopted even when the decomposition does not significantly improve the bounds.

Table 5 compares the pricing policies from the continuous formulation and the discrete formulation with $K = 10$. As before, for each problem instance, the first row is the absolute value and the second row is the relative value to the **DBD** bound under the continuous formulation. The solution time for the discrete formulation is shorter than that of the continuous formulation as shown in Table 3, but the expected revenues from the policies of **DB** and **DBD** are higher than **dDB** and **dDBD** for most problem instances. The last column in Table 5 reports the confidence interval (CI) on the revenue difference between **DBD** and **dDBD**, which indicates that the **DBD** policy generates higher revenue. Our overall conclusion is that the continuous formulation tends to produce better heuristic policies for the instances we considered. From an algorithmic standpoint, even if the discrete formulation is preferred in some applications, the continuous formulation can still serve as a benchmark to evaluate the performance and refine the given menu of prices.

9. Conclusion

Dynamic pricing for NRM has been studied since the seminal work of Gallego and van Ryzin (1997). It is well known that the dynamic programming formulation for the problem suffers from the curse of dimensionality as the state space grows exponentially with the number of resources on the network. While the linear programming based ADP has been used to solve the capacity control problem for NRM by many authors in recent years, much less work was done for the dynamic pricing problem. From a solution perspective, dynamic pricing problems are fundamentally harder than capacity control problems as even the simplest deterministic formulation is a nonlinear problem. In this paper, we have tried to extend this framework to the dynamic pricing problem. Despite some technical obstacles, our overarching conclusion is that the ADP framework can and probably should be applied to dynamic pricing problems.

One practical difficulty is that we are dealing with semi-infinite linear programs, which are in general very challenging to solve. For general demand functions, we propose a column generation algorithm to solve the problem. However, our numerical study using linear independent demand functions suggests that column generation algorithm can be quite computationally intensive. Therefore, much work is needed to reduce the computational intensity. For a special class of demand functions, i.e., the linear independent demand, we show that the affine ALP can be stated compactly as a SOCP, which can then be solved using standard nonlinear programming solvers. We illustrate via a computational study that solving the SOCP can be orders of magnitude faster than solving the original ALP formulation using column generation.

There are many possible avenues for future research. First, despite significant recent literature on ADP approach for NRM, almost all research so far considers capacity control problems and the corresponding ALPs are regular linear programs and therefore can be solved with linear programming solvers. In contrast, we solve nonlinear programs in our numerical study. Our numerical experience suggests that nonlinear solvers have made sufficient progress to be a viable option and should be seriously considered for other ADP applications. We are encouraged to learn about some recent applications in other areas. For example, Sun et al. (2014) consider quadratic approximations in the context of inventory management problems. Second, our work considers the affine approximation. Stronger value function approximations, such as the piecewise linear approximation, should be considered in future work. Finally, the compact SOCP formulation we proposed is based on a linear independent demand function. An immediate next step is to consider general linear demand functions that explicitly model demand interactions across products. Our initial attempt to derive a similar reduction result is not successful, even though this case can still be handled using our general framework. Furthermore, while linear demand functions have some good theoretical justifications (Besbes and Zeevi 2015), there are certainly many other demand functions. One example is piecewise linear demand functions. The formulation with linear demand can be adapted to create an iterative procedure to solve a version of the problem with a piecewise linear demand as follows: Start with a selected linear piece for each product and solve the problem; if the optimal price for a given product falls outside of the chosen linear piece, replace the demand function with the linear piece that covers the optimal price; repeat for a set number of iterations or until convergence. The convergence property of the algorithm is not well-understood and left for future research. For general demand functions, our proposed method relies on a column generation algorithm, which is likely to be computationally intensive. Finding more compact formulations and more efficient solution procedures for general demand functions is an important direction for future research. Much additional work remains to be done.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 71871139 and 71371119), Shanghai Shu Guang Program (No. 17SG16), and Shanghai Excellent Academic Leader Program (No. 16XD1401700). The authors are grateful to the department editor Huseyin Topaloglu, the anonymous senior editor, and three anonymous reviewers for their helpful comments.

References

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

Alizadeh, Farid, Donald Goldfarb. 2003. Second-order cone programming. *Mathematical programming* **95**(1) 3–51.

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.

Besbes, Omar, Assaf Zeevi. 2015. On the (surprising) sufficiency of linear models for dynamic pricing with demand learning. *Management Science* **61**(4) 723–739.

Boyd, Stephen, Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Charnes, A., W.W. Cooper, K. Kortanek. 1963. Duality in semi-infinite programs and some works of Haar and Caratheodory. *Management Science* **9**(2) 209–228.

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

Cooper, W. L., T. Homem-de-Mello. 2007. Some decomposition methods for revenue management. *Transportation Science* **41**(3) 332–353.

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

Dong, L., P. Kouvelis, Z. Tian. 2009. Dynamic pricing and inventory control of substitute products. *Manufacturing and Service Operations Management* **11**(2) 317–339.

Erdelyi, A., H. Topaloglu. 2010. Using decomposition methods to solve pricing problems in network revenue management. *Journal of Revenue and Pricing Management* **10**(4) 325–343.

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

Farias, Vivek F, Benjamin Van Roy. 2007. An approximate dynamic programming approach to network revenue management. *Preprint* .

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.

Grant, Michael, Stephen Boyd, Yinyu Ye. 2008. CVX: Matlab software for disciplined convex programming.

Kirshner, Samuel Nathan, Mikhail Nediak. 2015. Scalable dynamic bid prices for network revenue management in continuous time. *Production and Operations Management* **24**(10) 1621–1635.

Kunnumkal, S., H. Topaloglu. 2010. A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. *Production and Operations Management* **19**(5) 575–590.

Kunnumkal, Sumit, Kalyan Talluri. 2016. A note on relaxations of the choice network revenue management dynamic program. *Operations Research* **64**(1) 158–166.

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

Lim, Yun Fong, Selvaprabu Nadarajah, Qing Ding. 2015. Dynamic pricing for hotel rooms when customers request multiple-day stays. Working paper, Singapore Management University.

Liu, Q., G. J. van Ryzin. 2008. Strategic capacity rationing to induce early purchases. *Management Science* **54**(6) 1115–1131.

Lobo, Miguel Sousa, Lieven Vandenberghe, Stephen Boyd, Hervé Lebret. 1998. Applications of second-order cone programming. *Linear Algebra and its Applications* **284**(1) 193 – 228.

Maglaras, C., J. Meissner. 2006. Dynamic pricing strategies for multi-product revenue management problems. *Manufacturing and Service Operations Management* **8**(2) 136–148.

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.

Miranda Bront, J., I. Mendez-Diaz, G. Vulcano. 2009. A column generation algorithm for choice-based network revenue management. *Operations Research* **57**(3) 769–784.

Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, vol. 842. John Wiley & Sons.

Schön, Cornelia, Simon Hohberger. 2018. Continuous pricing in a capacitated network under mixed multinomial logit demand. Presentation at INFORMS Revenue Management and Pricing Section Conference, Toronto, Canada.

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

Sun, Peng, Kai Wang, Paul Zipkin. 2014. Quadratic approximation of cost functions in lost sales and perishable inventory control problems. Working Paper. Fuqua School of Business, Duke University, Durham, NC.

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. *The Theory and Practice of Revenue Management*. Springer, New York, NY.

Toh, Kim-Chuan, Michael J. Todd, Reha H. Tütüncü. 1999. SDPT3—A MATLAB software package for semidefinite programming, version 1.3. *Optimization Methods and Software* **11**(1-4) 545–581.

Tong, Chaoxu, Huseyin Topaloglu. 2013. On the approximate linear programming approach for network revenue management problems. *INFORMS Journal on Computing* **26**(1) 121–134.

Vossen, T., D. Zhang. 2015. Reductions of approximate linear programs for network revenue management. *Operations Research* **63**(6) 1352–1371.

Wittman, Michael D., Peter Belobaba. 2018. The implications of dynamic pricing for airline revenue management. Presentation at AGIFORS Revenue Management Study Group Conference, Hong Kong.

Wolsey, Laurence A, George L Nemhauser. 1988. *Integer and combinatorial optimization*. John Wiley & Sons.

Zhang, D. 2011. An improved dynamic programming decomposition approach for network revenue management. *Manufacturing and Service Operations Management* **13**(1) 35–52.

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., Z. Lu. 2013. Assessing the value of dynamic pricing in network revenue management. *INFORMS Journal on Computing* **25**(1) 102–115.

Zhang, D., L. Weatherford. 2017. Dynamic pricing for network revenue management: A new approach and application in the hotel industry. *INFORMS Journal on Computing* **29**(1) 18–35.

Table 4 Performance of different pricing policies under the continuous formulation

	Bound				Policy Performance				
	DBD Bound	z^A	**SBD** Bound	z^d	**DBD**	**DB**	**SBD**	**SB**	CI **DBD−SBD**
(1,4,200,1.3)	10715.47	10877.25 1.51%	10863.58 1.38%	11089.49 3.49%	10129 -5.47%	9600 -10.41%	10073 -6.00%	9588 -10.52%	(5.50,107.56)
(1,4,400,2.0)	15578.58	15777.77 1.28%	15811.93 1.50%	16105.55 3.38%	14931 -4.15%	14124 -9.34%	14870 -4.55%	14106 -9.46%	(9.39,100.00)
(1,4,800,1.9)	35388.82	35760.88 1.05%	35621.83 0.66%	36102.84 2.02%	34318 -3.02%	32939 -6.92%	34227 -3.28%	32897 -7.04%	(21.24,162.29)
(1,6,200,1.3)	11486.73	11661.49 1.52%	11697.29 1.83%	11928.67 3.85%	10568 -8.00%	10094 -12.13%	10503 -8.57%	9944 -13.43%	(8.77,120.82)
(1,6,400,1.6)	26417.77	26693.80 1.04%	26850.28 1.64%	27231.30 3.08%	24645 -6.71%	23572 -10.77%	24536 -7.12%	23483 -11.11%	(29.68,187.21)
(1,8,200,1.0)	12870.32	12965.32 0.74%	12922.58 0.41%	13030.09 1.24%	12158 -5.53%	11414 -11.31%	12047 -6.39%	11034 -14.27%	(34.44,187.32)
(2,4,200,1.5)	11855.05	12005.12 1.27%	12066.65 1.78%	12291.53 3.68%	10956 -7.58%	10488 -11.53%	10890 -8.14%	10412 -12.17%	(9.22,122.73)
(2,4,400,1.8)	20104.28	20314.84 1.05%	20303.59 0.99%	20612.11 2.53%	19152 -4.74%	18203 -9.46%	19071 -5.14%	18135 -9.80%	(5.75,157.06)
(2,4,800,2.0)	40281.34	40527.64 0.61%	40558.85 0.69%	40908.93 1.56%	38821 -3.62%	37059 -8.00%	38740 -3.83%	36980 -8.20%	(4.60,158.34)
(2,6,200,1.2)	12340.24	12553.99 1.73%	12423.54 0.67%	12726.02 3.13%	11643 -5.65%	11096 -10.09%	11540 -6.49%	10963 -11.16%	(36.33,169.34)
(2,6,400,1.2)	22762.19	23017.93 1.12%	22826.60 0.28%	23167.49 1.78%	22155 -2.67%	21100 -7.30%	22061 -3.08%	21020 -7.65%	(10.68,177.13)
(2,8,200,1.1)	12206.63	12356.38 1.23%	12291.21 0.69%	12530.84 2.66%	11953 -2.08%	11622 -4.79%	11885 -2.63%	11562 -5.28%	(6.45,129.43)

Table 5 Performance of pricing policies under continuous and discrete formulations

	Bound				Policy Performance				
	Continuous Price		Discrete Price		Continuous Price		Discrete Price		
	DBD Bound	z^A	**dDBD** Bound	z_D^A	**DBD**	**DB**	**dDBD**	**dDB**	CI **DBD−dDBD**
(1,4,200,1.3)	10715.47	10877.25 1.51%	10709.81 -0.05%	10865.28 1.40%	10129 -5.47%	9600 -10.41%	10083 -5.90%	9617 -10.25%	(6.16,86.90)
(1,4,400,2.0)	15578.58	15777.77 1.28%	15585.22 0.04%	15751.01 1.11%	14931 -4.15%	14124 -9.34%	14877 -4.51%	13976 -10.29%	(5.07,117.19)
(1,4,800,1.9)	35388.82	35760.88 1.05%	35409.85 0.06%	35726.41 0.95%	34318 -3.02%	32939 -6.92%	34250 -3.22%	32877 -7.10%	(4.32,132.60)
(1,6,200,1.3)	11486.73	11661.49 1.52%	11490.68 0.03%	11651.28 1.43%	10568 -8.00%	10094 -12.13%	10529 -8.34%	10086 -12.20%	(3.34,74.36)
(1,6,400,1.6)	26417.77	26693.80 1.04%	26431.05 0.05%	26664.26 0.93%	24645 -6.71%	23572 -10.77%	24560 -7.03%	23607 -10.64%	(17.77,152.98)
(1,8,200,1.0)	12870.32	12965.32 0.74%	12876.82 0.05%	12953.51 0.65%	12158 -5.53%	11414 -11.31%	12055 -6.33%	11462 -10.94%	(36.92,168.67)
(2,4,200,1.5)	11855.05	12005.12 1.27%	11846.45 -0.07%	11994.68 1.18%	10956 -7.58%	10488 -11.53%	10888 -8.16%	10500 -11.43%	(22.56,113.28)
(2,4,400,1.8)	20104.28	20314.84 1.05%	20113.75 0.05%	20302.34 0.99%	19152 -4.74%	18203 -9.46%	19091 -5.04%	18218 -9.38%	(8.45,113.68)
(2,4,800,2.0)	40281.34	40527.64 0.61%	40322.54 0.10%	40507.36 0.56%	38821 -3.62%	37059 -8.00%	38748 -3.81%	37052 -8.02%	(1.50,145.59)
(2,6,200,1.2)	12340.24	12553.99 1.73%	12344.11 0.03%	12544.61 1.66%	11643 -5.65%	11096 -10.09%	11583 -6.14%	11053 -10.43%	(6.19,113.60)
(2,6,400,1.2)	22762.19	23017.93 1.12%	22765.91 0.02%	23005.02 1.07%	22155 -2.67%	21100 -7.30%	22061 -3.08%	21096 -7.32%	(31.35,156.02)
(2,8,200,1.1)	12206.63	12356.38 1.23%	12208.60 0.02%	12351.09 1.18%	11953 -2.08%	11622 -4.79%	11880 -2.67%	11661 -4.47%	(31.57,114.16)

Appendix A. Proof of Proposition 1

Proof. To show the result, we need to check the feasibility of both $(\mathbf{P})^{\phi}$ and $(\mathbf{D})^{\phi}$.

We first check the feasibility of $(\mathbf{P})^{\phi}$. For notational simplicity, let $\kappa_t = \max_{\mathbf{r}_t \in \mathcal{R}_t} \sum_j \lambda_{t,j}(\mathbf{r}_t) r_{t,j}$ for each t be a set of constants. Take

$$\theta_t = \sum_{l=t}^{T} \kappa_l, \qquad \forall t, \tag{39}$$

$$V_{t,b} = 0, \qquad \forall t, b. \tag{40}$$

It follows that

$$\theta_t - \theta_{t+1} + \sum_{b \in \mathcal{B}} (V_{t,b} - V_{t+1,b}) \phi_b(\mathbf{x}) + \sum_j \lambda_{t,j}(\mathbf{r}_t) \sum_{b \in \mathcal{B}} V_{t+1,b} \left(\phi_b(\mathbf{x}) - \phi_b(\mathbf{x} - \mathbf{A}^j) \right) = \kappa_t \geq \sum_j \lambda_{t,j}(\mathbf{r}_t) r_{t,j}.$$

Therefore, (θ, V) defined in (39)–(40) is a feasible solution to $(\mathbf{P})^{\phi}$.

For the dual program $(\mathbf{D})^{\phi}$, it is straightforward to show that charging the null price in each period is feasible; i.e.

$$p_{t,\mathbf{x},\mathbf{r}_t} = \begin{cases} 1, & \text{if} \quad \mathbf{x} = \mathbf{c}, \mathbf{r}_{t,j} = r_{\infty}, \forall j, \\ 0, & \text{otherwise}, \end{cases} \qquad \forall t, \mathbf{x}, \mathbf{r}_t.$$

It can then be checked that both sides takes the value $\phi_b(\mathbf{c})$ in constraint (4). This completes the proof. ■

Appendix B. Affine Approximation with Log-Linear Independent Demand

Our main analysis is carried out for the linear independent demand. Here, we show that the same line of analysis can also be used to analyze the log-linear independent demand. Due to the similarity in the analysis, we recycle notations from the analysis for the linear independent demand model whenever appropriate.

The log-linear independent demand function is

$$\lambda_{t,j}(\mathbf{r}) = \alpha_{t,j} e^{-\beta_{t,j} r_{t,j}}, \qquad \forall t, j.$$

The column generation subproblem for $(\mathbf{D})^{\phi}$ under affine approximation and log-linear independent demand function is

$$\begin{aligned} \min_{\mathbf{x} \in \mathcal{X}_t, \mathbf{r}_t} \quad & \theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i}) x_i - \sum_j \alpha_{t,j} e^{-\beta_{t,j} r_{t,j}} \left(r_{t,j} - \sum_i a_{ij} V_{t+1,i} \right), \\ & x_i \in \{0, 1, ..., c_i\}. \qquad \forall i. \end{aligned}$$

and the revenue function is

$$H_{t,j}(\mathbf{V}_{t+1}, \mathbf{x}) = \max \ \alpha_{t,j} e^{-\beta_{t,j} r_{t,j}} \left(r_{t,j} - \sum_i a_{ij} V_{t+1,i} \right).$$

Suppose $\mathbf{x} \geq A^j$, then the optimal price for $H_{t,j}(\mathbf{V}_{t+1}, \mathbf{x})$ is $r^*_{t,j}(\mathbf{V}_{t+1}) = \sum_i a_{ij} V_{t+1,i} + \frac{1}{\beta_{t,j}}$ and the optimal revenue is $h_{t,j}(\mathbf{V}_{t+1}) = \frac{\alpha_{t,j}}{\beta_{t,j} e^{\beta_{t,j} \sum_i a_{ij} V_{t+1,i} + 1}}$. Similarly the column generation subproblem can be solved by solving its linear programming relaxation because the coefficient matrix is totally unimodular. The reformulated column generation subproblem implies that the primal problem $(\mathbf{P})^\phi$ under affine approximation and the log linear independent demand function can be written as

$$
\begin{aligned}
(\mathbf{P})^{A,L} \quad & \min_{\theta, \mathbf{V}} \quad \theta_1 + \sum_i V_{1,i} c_i \\
\text{s.t.} \quad & \theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i}) x_i - \sum_j w_j h_{t,j}(\mathbf{V}_{t+1}) \geq 0 \quad \forall\, \mathbf{x} \in \mathcal{X}_t, \mathbf{w} \text{ binary} : a_{ij} w_j \leq x_i, \forall\, t, i, j.
\end{aligned}
$$

Introducing a new variable $\rho_{t,j} = \sum_i a_{ij} V_{t+1,i} + \frac{1}{\beta_{t,j}}$, $(\mathbf{P})^{A,L}$ becomes

$$
\begin{aligned}
& \min_{\theta, \mathbf{V}, \rho} \quad \theta_1 + \sum_i V_{1,i} c_i \\
\text{s.t.} \quad & \theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i}) x_i - \sum_j w_j(\mathbf{x}) \frac{\alpha_{t,j}}{\beta_{t,j}} e^{-\beta_{t,j} \rho_{t,j}} \geq 0, && \forall\, t, \mathbf{x} \in \mathcal{X}_t. \\
& \rho_{t,j} = \sum_i a_{ij} V_{t+1,i} + \frac{1}{\beta_{t,j}}, && \forall\, t, j.
\end{aligned}
$$

the Lagrangian function of $(\mathbf{P1})^{A,R}$ can be written as

$$
\begin{aligned}
L(\theta, \mathbf{V}, \rho, \mu, \delta) = \quad & \theta_1 + \sum_i V_{1,i} c_i - \sum_{t, \mathbf{x} \in \mathcal{X}_t} \mu_{t,\mathbf{x}} \left[\theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i}) x_i - \sum_j w_j(\mathbf{x}) \frac{\alpha_{t,j}}{\beta_{t,j}} e^{-\beta_{t,j} \rho_{t,j}} \right] \\
& + \sum_{t,j} \delta_{t,j} \left[\rho_{t,j} - \sum_i a_{ij} V_{t+1,i} - \frac{1}{\beta_{t,j}} \right].
\end{aligned}
$$

Notice that $L(\theta, \mathbf{V}, \rho, \mu, \delta)$ is linear in θ and $\mathbf{V}$. We must have

$$
1 - \mu_{1,\mathbf{c}} = 0, \tag{41}
$$

$$
c_i (1 - \mu_{1,\mathbf{c}}) = 0, \qquad \forall\, i, \tag{42}
$$

$$
\sum_{\mathbf{x}} (\mu_{t-1,\mathbf{x}} - \mu_{t,\mathbf{x}}) = 0, \qquad \forall\, t \geq 2,
$$

$$
\sum_{\mathbf{x}} (\mu_{t-1,\mathbf{x}} - \mu_{t,\mathbf{x}}) x_i - \sum_j \delta_{t-1,j} a_{ij} = 0, \qquad \forall\, i, t \geq 2.
$$

It follows that

$$
z^L = \min_{\rho_{t,j}} \left(\delta_{t,j} \rho_{t,j} + \sum_{\mathbf{x}} \mu_{t,\mathbf{x}} w_j(\mathbf{x}) \frac{\alpha_{t,j}}{\beta_{t,j}} e^{-\beta_{t,j} \rho_{t,j}} \right) = \begin{cases} \frac{\delta_{t,j}}{\beta_{t,j}} \left[\ln(\alpha_{t,j} \sum_{\mathbf{x}} \mu_{t,\mathbf{x}} w_j(\mathbf{x})) - \ln \delta_{t,j} + 1 \right], & \text{if } \delta_{t,j} > 0, \\ 0, & \text{if } \delta_{t,j} = 0, \\ -\infty, & \text{if } \delta_{t,j} < 0, \end{cases}
$$

Note that the first two cases above can be combined together. Let

$$
g(\mu, \delta) = \min_{\theta, \mathbf{V}, \rho \geq 0} \; L(\theta, \mathbf{V}, \rho, \mu, \delta) = \sum_{t,j} \frac{\delta_{t,j}}{\beta_{t,j}} \left[\ln \left(\alpha_{t,j} \sum_{\mathbf{x}} \mu_{t,\mathbf{x}} w_j(\mathbf{x}) \right) - \ln \delta_{t,j} + 1 - 1 \right].
$$

Then the dual program can be written as

$$
\begin{aligned}
(\mathbf{D})^{A,L} \qquad & \max_{\mu,\delta} \quad \sum_{t,j} \frac{\delta_{t,j}}{\beta_{t,j}} \left[\ln\left(\alpha_{t,j} \sum_{\mathbf{x}} \mu_{t,\mathbf{x}} w_j(\mathbf{x}) \right) - \ln \delta_{t,j} \right] \\
\text{s.t.} \qquad & 1 - \mu_{1,\mathbf{c}} = 0, && (43) \\
& \sum_{\mathbf{x}\in\mathcal{X}_t} (\mu_{t-1,\mathbf{x}} - \mu_{t,\mathbf{x}}) = 0, \qquad \forall\ t \geq 2, && (44) \\
& \sum_{\mathbf{x}\in\mathcal{X}_t} (\mu_{t-1,\mathbf{x}} - \mu_{t,\mathbf{x}}) x_i - \sum_j \delta_{t-1,j} a_{ij} \beta_{t-1,j} = 0, \qquad \forall\ t \geq 2, i, \\
& \mu, \delta \geq 0.
\end{aligned}
$$

Similarly, let

$$
y_{t,i} = \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} x_i, \qquad \forall t, i, \tag{45}
$$

$$
z_{t,j} = \sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} w_j(\mathbf{x}) = \sum_{\mathbf{x}\in\mathcal{X}_t : \mathbf{x} \geq A^j} \mu_{t,\mathbf{x}}, \qquad \forall t, j. \tag{46}
$$

Since $\sum_{\mathbf{x}\in\mathcal{X}_t} \mu_{t,\mathbf{x}} = 1$ for all t, $0 \leq z_{t,j} \leq 1, \forall\ t, j$. Furthermore, since $x_i \geq a_{ij} w_j(\mathbf{x})$ for all i, j, t, $y_{t,i} \geq a_{ij} z_{t,j}, \forall t, i$. Putting everything together and applying variable aggregation to $(\mathbf{D}1)^{A,L}$ gives

$$
\begin{aligned}
(\mathbf{D}2)^{A,L} \qquad & z_L^A = \max_{\delta,\mathbf{y},\mathbf{z}} \quad \sum_{t,j} \frac{\delta_{t,j}}{\beta_{t,j}} \ln\left(\frac{\alpha_{t,j} z_{t,j}}{\delta_{t,j}} \right) \\
\text{s.t.} \qquad & y_{t,i} = \begin{cases} c_i, & \text{if } t = 1, \\ y_{t-1,i} - \sum_j a_{ij} \delta_{t-1,j}, & \text{if } t \geq 2, \end{cases} \qquad \forall\ t, i, && (47) \\
& y_{t,i} \geq a_{ij} z_{t,j}, \qquad \forall\ t, i, j, && (48) \\
& 0 \leq z_{t,j} \leq 1, \qquad \forall\ t, j. \\
& \delta,\ \mathbf{y},\ \mathbf{z} \geq 0.
\end{aligned}
$$

The equivalence between $(\mathbf{D})^{A,L}$ and $(\mathbf{D}2)^{A,L}$ can be proven through defining a new variable $s_{t,j} = \frac{\delta_{t,j}}{\alpha_{t,j} \sum_{\mathbf{x}} \mu_{t,\mathbf{x}} w_j(\mathbf{x})}$. Then the objective function in $(\mathbf{D})^{A,L}$ can be written as $-\sum_{t,j} \frac{\delta_{t,j}}{\beta_{t,j}} \ln s_{t,j}$ and one more constraint $\delta_{t,j} \leq \alpha_{t,j} s_{t,j} \sum_{\mathbf{x}} \mu_{t,\mathbf{x}} w_j(\mathbf{x})$ can be added. Similar to the proof in proposition 3, the inner program is a linear program and Dantzig-Wolfe decomposition can be applied to show the equivalence.

Note that $(\mathbf{D}2)^{A,L}$ is a convex program since the constraints are linear and the Hessian matrix of $\frac{\delta_{t,j}}{\beta_{t,j}} \ln\left(\frac{\alpha_{t,j} z_{t,j}}{\delta_{t,j}} \right)$ in the objective function is

$$
H = -\frac{1}{\beta_{t,j} \delta_{t,j} z_{t,j}^2} \begin{bmatrix} z_{t,j}^2 & -\delta_{t,j} z_{t,j} \\ -\delta_{t,j} z_{t,j} & \delta_{t,j}^2 \end{bmatrix} = -\frac{1}{\beta_{t,j} \delta_{t,j} z_{t,j}^2} = -\frac{1}{\beta_{t,j} \delta_{t,j} z_{t,j}^2} [z_{t,j} \ \ -\delta_{t,j}] \begin{bmatrix} z_{t,j} \\ -\delta_{t,j} \end{bmatrix} \preceq 0.
$$

However, solvers like CVX can't solve program $(\mathbf{D}2)^{A,L}$ due to disciplined convex programming error in the objective function. We can induce a solvable dual program through Lagrangian duality.

$$L(\delta,\mathbf{y},\mathbf{z},\mathbf{V},\omega,\zeta) = \sum_{t,j}\frac{\delta_{t,j}}{\beta_{t,j}}(\ln\alpha_{t,j}+\ln z_{t,j}-\ln\delta_{t,j}) - \sum_i (y_{1,i}-c_i)V_{1,i} - \sum_i (y_{t,i}-y_{t-1,i}+\sum_j a_{ij}\delta_{t-1,j})V_{t,i} + \sum_{t,i,j}(y_{t,i}-a_{ij}z_{t,j})\omega_{t,i,j} + \sum_{t,j}(1-z_{t,j})\zeta_{t,j} + \sum_{t,j}\eta_{t,j}\delta_{t,j} + \sum_{t,j}\hat{\eta}_{t,j}z_{t,j}.$$

Notice that $L(\delta,\mathbf{y},\mathbf{z},\mathbf{V},\omega,\zeta)$ is linear in $y_{t,i}$ and $y_{t,i}\geq 0$, thus the coefficient of $y_{t,i}$ must satisfy

$$V_{t+1,i} - V_{t,i} + \sum_j \omega_{t,i,j} \leq 0.$$

Assume $L(\delta,\mathbf{y},\mathbf{z},\mathbf{V},\omega,\zeta)$ achieves its minimum at $\delta^*,\mathbf{z}^*$ and we have

$$\frac{\partial L}{\partial \delta^*_{t,j}} = \frac{1}{\beta_{t,j}}(\ln\alpha_{t,j}+\ln z^*_{t,j}-\ln\delta^*_{t,j}) + \frac{\delta^*_{t,j}}{\beta_{t,j}}(-\frac{1}{\delta^*_{t,j}}) - \sum_i a_{ij}V_{t+1,i} + \eta_{t,j} = 0, \qquad \forall t,j, \tag{49}$$

$$\frac{\partial L}{\partial z^*_{t,j}} = \frac{\delta^*_{t,j}}{\beta_{t,j}}\frac{1}{z^*_{t,j}} - \sum_i a_{ij}\omega_{t,i,j} - \zeta_{t,j} + \hat{\eta}_{t,j} = 0, \qquad \forall t,j. \tag{50}$$

For any $\mathbf{V},\omega\geq 0,\zeta\geq 0$, based on (49) and (50), we have

$$\begin{aligned}
\zeta_{t,j} + \sum_i a_{ij}\omega_{t,i,j} = {} & \hat{\eta}_{t,j} + \frac{\delta^*_{t,j}}{\beta_{t,j}}\frac{1}{z^*_{t,j}} \\
\geq {} & \frac{1}{\beta_{t,j}}\exp(\ln\delta^*_{t,j} - \ln z^*_{t,j}) \\
= {} & \frac{1}{\beta_{t,j}}\exp\left(\ln\alpha_{t,j} - 1 - \beta_{t,j}\sum_i a_{ij}V_{t+1,i} + \eta_{t,j}\right) \\
\geq {} & \frac{\alpha_{t,j}}{\beta_{t,j}}\exp\left(-\beta_{t,j}\sum_i a_{ij}V_{t+1,i} - 1\right).
\end{aligned}$$

The first inequality and the last inequality are satisfied due to $\hat{\eta}_{t,j},\eta_{t,j}\geq 0,\forall t,j$. On the other hand, the Lagrangian function can be written as

$$\begin{aligned}
L(\delta^*,\mathbf{y},\mathbf{z}^*,\mathbf{V},\omega,\zeta) = {} & \sum_{t,j}\zeta_{t,j} + \sum_i V_{1,i}c_i + \sum_{t,j}(\frac{\partial L}{\partial \delta^*_{t,j}} + \frac{1}{\beta_{t,j}})\delta^*_{t,j} + \sum_{t,j}(\frac{\partial L}{\partial z^*_{t,j}} - \frac{\delta^*_{t,j}}{\beta_{t,j}}\frac{1}{z^*_{t,j}})z^*_{t,j} \\
= {} & \sum_{t,j}\zeta_{t,j} + \sum_i V_{1,i}c_i.
\end{aligned}$$

Therefore the Lagrangian dual program of $(\mathbf{D}2)^{A,L}$ can be written as

$$\min_{V,\omega,\zeta,\rho} \qquad \sum_{t,j}\zeta_{t,j} + \sum_i V_{1,i}c_i$$

$$\text{s.t.} \quad \zeta_{t,j} + \sum_i a_{ij}\omega_{t,i,j} \geq \frac{\alpha_{t,j}}{\beta_{t,j}} \exp\left(-\beta_{t,j}\sum_i a_{ij}V_{t+1,i} - 1\right), \quad \forall t,j,$$
$$V_{t,i} - V_{t+1,i} - \sum_j \omega_{t,i,j} \geq 0, \quad \forall t,i,$$
$$\rho,\zeta,\omega \geq 0.$$

For convex problem $(\mathbf{D}2)^{A,L}$, there exists a strictly feasible solution $\delta_{t,j} = z_{t,j} = \epsilon$ and $y_{1,i} = c_i, \forall i, y_{t,i} = y_{t-1,i} - \sum_j a_{ij}\delta_{t-1,j}, \forall t,i$, where ϵ denote a small positive number like 10^{-10}. Thus strong duality holds for convex problem $(\mathbf{D}2)^{A,L}$ according to Slater's Theorem in Section 5.2.3, Boyd and Vandenberghe (2004).

Appendix C. Monotonicity of $(\theta, \mathbf{V})$ in the Affine Solution

Using (6), the program $(\mathbf{P})^{\phi}$ can be written as

$$(\mathbf{P})^A \quad z^A = \inf_{\theta,\mathbf{V}} \quad \theta_1 + \sum_i V_{1,i}c_i$$
$$\text{s.t.} \quad \theta_t - \theta_{t+1} + \sum_i \left(V_{t,i}x_i - \left(x_i - \sum_j \lambda_{t,j}(\mathbf{r}_t)a_{ij}\right)V_{t+1,i}\right) \geq \sum_j \lambda_{t,j}(\mathbf{r}_t)r_{t,j}, \quad \forall\ t,\mathbf{x},\mathbf{r}_t \in \mathcal{R}_t(\mathbf{x}).$$

The corresponding dual problem is

$$(\mathbf{D})^A \quad \sup_{\mathbf{p}} \sum_{t,\mathbf{x},\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} \left(\sum_j \lambda_{t,j}(\mathbf{r}_t)r_{t,j}\right)p_{t,\mathbf{x},\mathbf{r}_t}$$
$$\text{s.t.} \quad \sum_{\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} x_i p_{t,\mathbf{x},\mathbf{r}_t} = \begin{cases} c_i, & \text{if } t=1, \\ \sum_{\mathbf{x}\in\mathcal{X}_{t-1},\mathbf{r}_{t-1}\in\mathcal{R}_{t-1}(\mathbf{x})} \left(x_i - \sum_j \lambda_{t-1,j}(\mathbf{r}_{t-1})a_{ij}\right)p_{t-1,\mathbf{x},\mathbf{r}_{t-1}}, & \text{if } t>1, \end{cases} \quad \forall\ t,i, \tag{51}$$
$$\sum_{\mathbf{x}\in\mathcal{X}_t,\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} p_{t,\mathbf{x},\mathbf{r}_t} = 1, \quad \forall\ t,$$
$$\mathbf{p} \geq 0.$$

We show that $(\theta,\mathbf{V})$ in $(\mathbf{P})^A$ is non-increasing in t. For problem $(\mathbf{P})^A$-$(\mathbf{D})^A$, let

$$t_i^* = \arg\min\{t\in 1,...,T : \exists j,\mathbf{x},\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x}) \text{ with } a_{ij}=1, \lambda_{t,j}(\mathbf{r}_t)>0, p_{t,\mathbf{x},\mathbf{r}_t}>0\}, \quad \forall\ i.$$

Proposition 6. *Assume that t_i^* exists. Then*

(i) For all $t > t_i^*$, there exists $p_{t,\mathbf{x},\mathbf{r}_t} > 0$ such that $x_i < c_i$, and

(ii) For all $t \leq t_i^*$, $p_{t,\mathbf{x},\mathbf{r}_t} > 0$ only if $x_i = c_i$.

Proof. We prove (i) and (ii) by contradiction.

(i) Suppose $\forall\ t > t_i^*, \forall\ p_{t,\mathbf{x},\mathbf{r}_t} > 0, x_i = c_i$, then $\sum_{\mathbf{x},\mathbf{r}_{t+1}\in\mathcal{R}_{t+1}(\mathbf{x})} c_i p_{t+1,\mathbf{x},\mathbf{r}_{t+1}} = c_i$.

According to the definition of t_i^*, there exist $j', \mathbf{x}', \mathbf{r}'_{t_i^*}$ such that $a_{ij'} > 0, \lambda_{t_i^*,j'}(\mathbf{r}'_{t_i^*}) > 0, p_{t_i^*,\mathbf{x}',\mathbf{r}'_{t_i^*}} > 0$. Then $x'_i - \sum_j \lambda_{t_i^*,j}(\mathbf{r}_{t_i^*})\,(x'_i - (x'_i - a_{ij})) \le x'_i - \lambda_{t_i^*,j'}(\mathbf{r}_{t_i^*})\,(x'_i - (x'_i - a_{ij'})) < x'_i \le c_i$. Then for constraints (51) in period t_i^*+1 in dual problem $(\mathbf{D})^A$, we have

$$\begin{aligned}\sum_{\mathbf{x},\mathbf{r}_{t_i^*+1}\in\mathcal{R}_{t_i^*+1}(\mathbf{x})} x_i p_{t_i^*+1,\mathbf{x},\mathbf{r}_{t_i^*+1}} &= \sum_{\mathbf{x},\mathbf{r}_{t_i^*}\in\mathcal{R}_{t_i^*}(\mathbf{x})} \left[x_i - \sum_j \lambda_{t_i^*,j}(\mathbf{r}_{t_i^*})\,(x_i - (x_i - a_{ij}))\right] p_{t_i^*,\mathbf{x},\mathbf{r}_{t_i^*}} \\ &\le \left[x'_i - \sum_j \lambda_{t_i^*,j}(\mathbf{r}'_{t_i^*})\,(x'_i - (x'_i - a_{ij}))\right] p_{t_i^*,\mathbf{x}',\mathbf{r}'_{t_i^*}} + \sum_{\mathbf{x}\neq\mathbf{x}',\mathbf{r}_{t_i^*}\neq\mathbf{r}'_{t_i^*}} c_i p_{t_i^*,\mathbf{x},\mathbf{r}_{t_i^*}} < c_i\end{aligned}$$

which is a contradiction. Thus there exists at least one $p_{t,\mathbf{x},\mathbf{r}_t} > 0$ such that $x_i < c_i$.

(ii) Suppose there exists $\mathbf{x}', \mathbf{r}'_t$ such that $x'_i < c_i$ and $p_{t,\mathbf{x}',\mathbf{r}'_t} > 0$ in period $t \le t_i^*$. Then

$$\begin{aligned}\sum_{\mathbf{x},\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} x_i p_{t,\mathbf{x},\mathbf{r}_t} =& x'_i p_{t,\mathbf{x}',\mathbf{r}'_t} + \sum_{\mathbf{x}\neq\mathbf{x}',\mathbf{r}\neq\mathbf{r}'} x_i p_{t,\mathbf{x},\mathbf{r}_t} \\ \le& x'_i p_{t,\mathbf{x}',\mathbf{r}'_t} + c_i \sum_{\mathbf{x}\neq\mathbf{x}',\mathbf{r}\neq\mathbf{r}'} p_{t,\mathbf{x},\mathbf{r}_t} \\ <& c_i p_{t,\mathbf{x}',\mathbf{r}'_t} + c_i \sum_{\mathbf{x}\neq\mathbf{x}',\mathbf{r}\neq\mathbf{r}'} p_{t,\mathbf{x},\mathbf{r}_t} = c_i.\end{aligned}$$

However, according to the definition of t_i^*, we have $\lambda_{t,j}(\mathbf{r}_t)\,(x_i - (x_i - a_{ij}))\,p_{t,\mathbf{x},\mathbf{r}_t} = 0\ \forall t, i, \mathbf{x}, \mathbf{r}_t$ and then according to constraints (51)

$$\begin{aligned}\sum_{\mathbf{x},\mathbf{r}_t\in\mathcal{R}_t(\mathbf{x})} x_i p_{t,\mathbf{x},\mathbf{r}_t} &= \sum_{\mathbf{x},\mathbf{r}_{t-1}\in\mathcal{R}_{t-1}(\mathbf{x})} \left[x_i - \sum_j \lambda_{t-1,j}(\mathbf{r}_t)\,(x_i - (x_i - a_{ij}))\right] p_{t-1,\mathbf{x},\mathbf{r}_{t-1}} \\ &= \sum_{\mathbf{x},\mathbf{r}_{t-1}\in\mathcal{R}_{t-1}(\mathbf{x})} x_i p_{t-1,\mathbf{x},\mathbf{r}_{t-1}} = \ldots = \sum_{\mathbf{x},\mathbf{r}_1\in\mathcal{R}_1(\mathbf{x})} x_i p_{1,\mathbf{x},\mathbf{r}_1} = c_i,\end{aligned}$$

which is a contradiction. Thus for all $p_{t,\mathbf{x},\mathbf{r}_t} > 0$, $x_i = c_i$. ■

PROPOSITION 7. *For primal problem $(\mathbf{P})^A$ there exists an optimal solution ($\theta^*, \mathbf{V}^*$) such that*

$$\begin{aligned}&\theta_1^* \ge \theta_2^* \ge \ldots \ge \theta_T^* \ge \theta_{T+1}^* = 0, \\ &V_{1,i}^* \ge V_{2,i}^* \ge \ldots \ge V_{T,i}^* \ge V_{T+1,i}^* = 0, \qquad \forall\ i.\end{aligned}$$

Proof. In program $(\mathbf{P})^A$ since $\forall\ t, \mathbf{x}, \mathbf{r}_t \in \mathcal{R}_t(\mathbf{x})$ all constraints are satisfied, then for each period with $\mathbf{x} = \mathbf{0}$ and $\lambda_{t,j}(\mathbf{r}_t) = 0,\ \forall j$ all constraints are also satisfied. Therefore we have $\theta_t^* - \theta_{t+1}^* \ge 0,\ \forall t$.

Then we prove the monotonicity of $\mathbf{V}$ $\forall i'$. Suppose $t_{i'}^*$ exists.

(1) First we prove $V_{t,i'}^* \ge V_{t+1,i'}^*$ when $t > t_{i'}^*$.

Suppose the optimal solution for primal problem $(\mathbf{P})^A$ is V^*, θ^*. According to complementary slackness, for $p_{t,\mathbf{x},\mathbf{r}_t} > 0$ in dual problem $(\mathbf{D})^A$, we have

$$\theta_t^* - \theta_{t+1}^* + \sum_i (V_{t,i}^* - V_{t+1,i}^*) x_i + \sum_j \lambda_{t,j}(\mathbf{r}_t) \left[\sum_i V_{t+1,i}^* \left(x_i - (x_i - a_{ij}) \right) - r_{t,j} \right] = 0$$

According to (1) in proposition 6, there exists $x_{i'} < c_{i'}$. Thus we can set $x'_{i'} = x_{i'} + \varepsilon \le c_{i'}$ and for other $i \ne i'$, x_i stay the same, where $\varepsilon \le c_{i'} - x_{i'}$ is a small positive integer. The constraints in primal problem $(\mathbf{P})^A$ still holds, that's

$$\begin{aligned} 0 \le \quad & \theta_t^* - \theta_{t+1}^* + \sum_{i \ne i'} (V_{t,i}^* - V_{t+1,i}^*) x_i + (V_{t,i'}^* - V_{t+1,i'}^*) \left(x_{i'} + \varepsilon \right) \\ & + \sum_j \lambda_{t,j}(\mathbf{r}_t) \left[\sum_{i \ne i'} V_{t+1,i}^* \left(x_i - (x_i - a_{ij}) \right) + V_{t+1,i'}^* \left(x_{i'} + \varepsilon - (x_{i'} - a_{i'j} + \varepsilon) \right) - r_{t,j} \right] \\ = \quad & \theta_t^* - \theta_{t+1}^* + \sum_i (V_{t,i}^* - V_{t+1,i}^*) x_i + \sum_j \lambda_{t,j}(\mathbf{r}_t) \left(\sum_i V_{t+1,i}^* a_{ij} - r_{t,j} \right) + (V_{t,i'}^* - V_{t+1,i'}^*) \varepsilon \\ = \quad & (V_{t,i'}^* - V_{t+1,i'}^*) \varepsilon \end{aligned}$$

Because $\varepsilon \ge 0$, thus $V_{t,i'}^* - V_{t+1,i'}^* \ge 0$, $V_{t,i'}^* \ge V_{t+1,i'}^*$.

(2) Secondly we prove $V_{t,i'}^* \le V_{t+1,i'}^*$ when $t < t_{i'}^*$.

Because $\sum_{\mathbf{x},\mathbf{r}_t \in \mathcal{R}_t(\mathbf{x})} p_{t,\mathbf{x},\mathbf{r}_t} = 1$, there exists at least one $p_{t,\mathbf{x},\mathbf{r}_t} > 0$. Then there is one $x_{i'} = c_{i'}$, according to complementary slackless

$$\theta_t^* - \theta_{t+1}^* + \sum_i (V_{t,i}^* - V_{t+1,i}^*) x_i + \sum_j \lambda_{t,j}(\mathbf{r}_t) \left[\sum_i V_{t+1,i}^* \left(x_i - (x_i - a_{ij}) \right) - r_{t,j} \right] = 0$$

According to (2) in proposition 6, $x_{i'} = c_{i'}$. Thus we can set $x'_{i'} = c_{i'} - \varepsilon$ and for other $i \ne i'$, x_i stay the same, where $\varepsilon \le c_{i'}$ is a small positive integer. The constraints in primal problem $(\mathbf{P})^A$ still holds, that's

$$\begin{aligned} 0 \le \quad & \theta_t^* - \theta_{t+1}^* + \sum_{i \ne i'} (V_{t,i}^* - V_{t+1,i}^*) x_i + (V_{t,i'}^* - V_{t+1,i'}^*)(c_{i'} - \varepsilon) \\ & + \sum_j \lambda_{t,j}(\mathbf{r}_t) \left[\sum_{i \ne i'} V_{t+1,i}^* \left(x_i - (x_i - a_{ij}) \right) + V_{t+1,i'}^* \left(c_{i'} - \varepsilon - (c_{i'} - a_{ij} - \varepsilon) \right) - r_{t,j} \right] \\ = \quad & \theta_t^* - \theta_{t+1}^* + \sum_i (V_{t,i}^* - V_{t+1,i}^*) x_i + \sum_j \lambda_{t,j}(\mathbf{r}_t) \left(\sum_i V_{t+1,i}^* a_{ij} - r_{t,j} \right) - (V_{t,i'}^* - V_{t+1,i'}^*) \varepsilon \\ = \quad & -(V_{t,i'}^* - V_{t+1,i'}^*) \varepsilon \end{aligned}$$

Because $\varepsilon \geq 0$, thus $V^*_{t,i'} - V^*_{t+1,i'} \leq 0$, $V^*_{t,i'} \leq V^*_{t+1,i'}$.

(3) If $V^*_{t,i'} < V^*_{t+1,i'}$ for some $t \leq t^*_{i'}$, then we can raise $V^*_{t,i'}$ to $V^*_{t+1,i'}$ without loss of optimality.

$$V'_{t,i'} = V^*_{t+1,i'}, \quad \theta'_t = \theta^*_t + (V^*_{t,i'} - V^*_{t+1,i'})c_{i'},$$

For constraints with period t and $t+1$, according to proposition 6, there is

$$\begin{aligned}
&\pi_t(\theta'_t, \theta^*_{t+1}, \mathbf{V}'_t, \mathbf{V}^*_{t+1}) \\
= \quad & \theta'_t - \theta^*_{t+1} + \sum_{i \neq i'} (V^*_{t,i} - V^*_{t+1,i})x_i + (V'_{t,i'} - V^*_{t+1,i'})x_{i'} + \sum_j \lambda_{t,j}(\mathbf{r}_t) \left[\sum_i V^*_{t+1,i} \left(x_i - (x_i - a_{ij}) \right) - r_{t,j} \right] \\
= \quad & \theta^*_t + (V^*_{t,i'} - V^*_{t+1,i'})c_{i'} - \theta^*_{t+1} + \sum_{i \neq i'} (V^*_{t,i} - V^*_{t+1,i})x_i + (V^*_{t+1,i} - V^*_{t+1,i})x_i + \sum_j \lambda_{t,j}(\mathbf{r}_t) \left(\sum_i V^*_{t+1,i} a_{ij} - r_{t,j} \right) \\
= \quad & \theta^*_t - \theta^*_{t+1} + \sum_{i \neq i'} (V^*_{t,i} - V^*_{t+1,i})x_i + (V^*_{t,i} - V^*_{t+1,i})c_i + \sum_j \lambda_{t,j}(\mathbf{r}_t) \left(\sum_i V^*_{t+1,i} a_{ij} - r_{t,j} \right). \qquad (52)
\end{aligned}$$

Then (52) can be seen as the left hand side of a constraint in $(\mathbf{P})^A$ with $\mathbf{x} = (x_1, ..., c, ..., x_m)$. Thus $\pi_t(\theta'_t, \theta^*_{t+1}, \mathbf{V}'_t, \mathbf{V}^*_{t+1}) \geq 0$.

For constraints with period t and $t-1$, from constraints in primal problem $(\mathbf{P})^A$, we have

$$\begin{aligned}
&\theta^*_{t-1} - \theta^*_t + \sum_{i \neq i'} (V^*_{t-1,i} - V^*_{t,i})x_i + (V^*_{t-1,i'} - V^*_{t,i'})x_{i'} \\
&+ \sum_j \lambda_{t-1,j}(\mathbf{r}_{t-1}) \left[\sum_{i \neq i'} V^*_{t,i} \left(x_i - (x_i - a_{ij}) \right) + V^*_{t,i'} \left(x_{i'} - (x_{i'} - a_{i'j}) \right) - r_{t-1,j} \right] \geq 0.
\end{aligned}$$

Then

$$\begin{aligned}
& \theta^*_{t-1} - \theta'_t + \sum_{i \neq i'} (V^*_{t-1,i} - V^*_{t,i})x_i + (V^*_{t-1,i'} - V'_{t,i'})x_{i'} + \sum_j \lambda_{t-1,j}(\mathbf{r}_{t-1}) \left[\sum_i V'_{t,i} \left(x_i - (x_i - a_{ij}) \right) - r_{t-1,j} \right] \\
= \quad & \theta^*_{t-1} - \theta^*_t - (V^*_{t,i'} - V^*_{t+1,i'})c_{i'} + \sum_{i \neq i'} (V^*_{t-1,i} - V^*_{t,i})x_i + (V^*_{t-1,i'} - V^*_{t+1,i'})x_{i'} + \\
& \sum_j \lambda_{t-1,j}(\mathbf{r}_{t-1}) \left[\sum_{i \neq i'} V^*_{t,i} \left(x_i - (x_i - a_{ij}) \right) + V^*_{t+1,i'} \left(x_{i'} - (x_{i'} - a_{i'j}) \right) - r_{t-1,j} \right] \\
= \quad & \theta^*_{t-1} - \theta^*_t + \sum_{i \neq i'} (V^*_{t-1,i} - V^*_{t,i})x_i + (V^*_{t-1,i'} - V^*_{t,i'})x_{i'} \\
& + \sum_j \lambda_{t-1,j}(\mathbf{r}_{t-1}) \left(\sum_{i \neq i'} V^*_{t,i} a_{ij} + V^*_{t,i'} a_{i'j} - r_{t-1,j} \right) + (V^*_{t+1,i'} - V^*_{t,i'}) \left[c_{i'} - x_{i'} + \sum_j \lambda_{t-1,j}(\mathbf{r}_{t-1}) a_{i'j} \right]
\end{aligned}$$

$$\geq \quad (V^*_{t+1,i'} - V^*_{t,i'})\left(c_{i'} - x_{i'} + \sum_j \lambda_{t-1,j}(\mathbf{r}_{t-1})a_{i'j}\right) \geq 0$$

The last inequality is satisfied because besides $c_{i'} \geq x_{i'}$, $x_{i'} \geq x_{i'} - a_{i'j}$, $\lambda_{t-1,j}(\mathbf{r}_{t-1}) \geq 0,\ \forall\ t,j$, we know when $t \leq t^*$, $V^*_{t+1,i} \geq V^*_{t,i}$ from part (2).

To sum up, when $\theta^*_t, V^*_{t,i'}$ becomes $\theta'_t, V'_{t,i'}$, the problem has the same optimal solution.

If $t^*_{i'}$ does not exist, all $t \leq t^*_{i'}$, from parts (2) and (3) we can easily get $V^*_{1,i'} = V^*_{2,i'} = \ldots = V^*_{T,i'} = V^*_{T+1,i'} = 0\ \forall\ i$. ■

Appendix D. SOCP Duality

Constraint (13) can be written as

$$\left[\theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i})x_i + 1\right]^2 \geq \sum_j \beta_{t,j} w_j(\mathbf{x})\rho^2_{t,j} + \left[\theta_t - \theta_{t+1} + \sum_i (V_{t,i} - V_{t+1,i})x_i - 1\right]^2.$$

Then $(\mathbf{P}1)^{A,R}$ can be written as a SOCP

$$
\begin{array}{lll}
(\mathbf{P}1)^{A,R} & \displaystyle\min_{\theta,\mathbf{V},\rho,\bar{\rho},d^a,d^b} \quad \theta_1 + \sum_i V_{1,i}c_i & \\
\text{s.t.} & d^a_{t,\mathbf{x}} = \theta_t - \theta_{t+1} + \displaystyle\sum_i (V_{t,i} - V_{t+1,i})x_i + 1, & \forall t, \mathbf{x}, \\
& d^b_{t,\mathbf{x}} = \theta_t - \theta_{t+1} + \displaystyle\sum_i (V_{t,i} - V_{t+1,i})x_i - 1, & \forall t, \mathbf{x}, \\
& \theta_t - \theta_{t+1} + \displaystyle\sum_i (V_{t,i} - V_{t+1,i})x_i + 1 \geq 0, & \forall t, \mathbf{x}, \\
& \bar{\rho}_{t,j,\mathbf{x}} = \sqrt{\beta_{t,j} w_j(\mathbf{x})}\rho_{t,j}, & \forall t, \mathbf{x}, j, \\
& \rho_{t,j} + \displaystyle\sum_i a_{ij} V_{t+1,i} \geq \frac{\alpha_{t,j}}{\beta_{t,j}}, & \forall\ t, j, \\
& \begin{pmatrix} d^a_{t,\mathbf{x}} \\ d^b_{t,\mathbf{x}} \\ \bar{\rho}_{t,1,\mathbf{x}} \\ \cdots \\ \bar{\rho}_{t,n,\mathbf{x}} \end{pmatrix} \succeq_Q 0, & \forall t, \mathbf{x}, \\
& \rho_{t,j} \geq 0, & \forall\ t, j,
\end{array}
$$

where $\succeq_Q$ refers to second-order cone inequalities. The dual program of $(\mathbf{P}1)^{A,R}$ is

$$(\mathbf{DSOCP}) \quad \max_{\mu^a,\mu^b,\delta,\bar{\delta}} \sum_{t,\mathbf{x}} (\mu^b_{t,\mathbf{x}} - \mu^a_{t,\mathbf{x}}) + \sum_{t,j} \frac{\alpha_{t,j}}{\beta_{t,j}}\delta_{t,j} \tag{53}$$

$$\text{s.t.} \quad \sum_{\mathbf{x}} (\mu^a_{t,\mathbf{x}} + \mu^b_{t,\mathbf{x}}) = 1, \qquad \forall t, j, \tag{54}$$

$$\sum_{\mathbf{x}} (\mu^a_{t,\mathbf{x}} + \mu^b_{t,\mathbf{x}})x_i = \begin{cases} c_i, & \text{if } t = 1, \\ \sum_{\mathbf{x}} (\mu^a_{t-1,\mathbf{x}} + \mu^b_{t-1,\mathbf{x}})x_i - \sum_j a_{ij}\delta_{t-1,j}, & \text{if } t \geq 2, \end{cases} \quad \forall\ i, \tag{55}$$

$$\delta_{t,j} - \sum_{\mathbf{x}} \sqrt{\beta_{t,j} w_j(\mathbf{x})} \bar{\delta}_{t,j,\mathbf{x}} \leq 0, \qquad \forall t, j, \quad (56)$$

$$\begin{pmatrix} \mu^a_{t,\mathbf{x}} \\ \mu^b_{t,\mathbf{x}} \\ \bar{\delta}_{t,1,\mathbf{x}} \\ \cdots \\ \bar{\delta}_{t,n,\mathbf{x}} \end{pmatrix} \succeq_Q 0, \qquad \forall t, \mathbf{x}, \quad (57)$$

$$\delta_{t,j} \geq 0, \qquad \forall t, j. \quad (58)$$

At optimality, (56) is binding because we can always increase the objective value by increasing $\delta_{t,j}$ until the constraint is binding. At optimality, the objective function (53) is

$$\begin{aligned} \sum_{t,j} \frac{\alpha_{t,j}}{\beta_{t,j}} \delta_{t,j} - \sum_{t,\mathbf{x}} (\mu^a_{t,\mathbf{x}} - \mu^b_{t,\mathbf{x}}) &= \sum_{t,j} \frac{\alpha_{t,j}}{\beta_{t,j}} \delta_{t,j} - \sum_{t,\mathbf{x}:\mu^a_{t,\mathbf{x}}+\mu^b_{t,\mathbf{x}} \neq 0} (\mu^a_{t,\mathbf{x}} - \mu^b_{t,\mathbf{x}}) - 2 \sum_{t,\mathbf{x}:\mu^a_{t,\mathbf{x}}+\mu^b_{t,\mathbf{x}} = 0} \mu^a_{t,\mathbf{x}} \\ &= \sum_{t,j} \frac{\alpha_{t,j}}{\beta_{t,j}} \delta_{t,j} - \sum_{t,\mathbf{x}:\mu^a_{t,\mathbf{x}}+\mu^b_{t,\mathbf{x}} \neq 0} \frac{\sum_j \bar{\delta}^2_{t,j,\mathbf{x}}}{\mu^a_{t,\mathbf{x}} + \mu^b_{t,\mathbf{x}}} \qquad (53') \end{aligned}$$

At optimality, the second equality holds because the last term (i.e., $-2\sum_{t,\mathbf{x}:\mu^a_{t,\mathbf{x}}+\mu^b_{t,\mathbf{x}}=0} \mu^a_{t,\mathbf{x}}$) will equal to zero to achieve maximum, and constraint (57) is binding to maximize the second term (i.e., $-\sum_{t,\mathbf{x}:\mu^a_{t,\mathbf{x}}+\mu^b_{t,\mathbf{x}}\neq 0}(\mu^a_{t,\mathbf{x}} - \mu^b_{t,\mathbf{x}})$).

Define model $(DSOCP')$ as (53'),(54),(55),(56'),(57),(58), where (56') is the binding part of (56), i.e., $\delta_{t,j} - \sum_{\mathbf{x}} \sqrt{\beta_{t,j} w_j(\mathbf{x})} \bar{\delta}_{t,j,\mathbf{x}} = 0, \ \forall t, j$. Obviously, $(DSOCP')$ is equivalent to $(DSOCP)$.

We next show that $(DSOCP')$ is equivalent to $(D1)^{A,R}$. Let Z^{DS} and Z^{AR} stand for the optimal objective value of $(DSOCP')$ and $(D1)^{A,R}$, respectively. First, we show that any feasible solution of $(DSOCP')$ can be mapped to a feasible solution of $(D1)^{A,R}$, thus $Z^{DS} \leq Z^{AR}$. Suppose that $(\hat{\mu}^a_{t,\mathbf{x}}, \hat{\mu}^b_{t,\mathbf{x}}, \hat{\delta}_{t,j}, \hat{\bar{\delta}}_{t,j,\mathbf{x}})$ is a feasible solution of $(DSOCP')$, we can construct a feasible solution of $(D1)^{A,R}$ as the following.

$$\begin{aligned} \tilde{\mu}_{t,\mathbf{x}} &= \hat{\mu}^a_{t,\mathbf{x}} + \hat{\mu}^b_{t,\mathbf{x}} \\ \tilde{\delta}_{t,j} &= \hat{\delta}_{t,j} \\ \tilde{s}_{t,j} &= \beta_{t,j} \sum_{\mathbf{x}:w_j(\mathbf{x}),\hat{\mu}^a_{t,\mathbf{x}}+\hat{\mu}^b_{t,\mathbf{x}}\neq 0} \frac{\hat{\bar{\delta}}^2_{t,j,\mathbf{x}}}{\hat{\mu}^a_{t,\mathbf{x}} + \hat{\mu}^b_{t,\mathbf{x}}} \end{aligned}$$

It is easy to see that $\tilde{\mu}_{t,\mathbf{x}}$ and $\tilde{\delta}_{t,j}$ meet the second and third constraints of $(D1)^{A,R}$, which are equivalent to Constraints (54) and (55) of $(DSOCP')$.

By constraint (56') and (58), we can get

$$\tilde{\delta}^2_{t,j} = \hat{\delta}^2_{t,j} = \left(\sum_{\mathbf{x}} \sqrt{\beta_{t,j} w_j(\mathbf{x})} \hat{\bar{\delta}}_{t,j,\mathbf{x}} \right)^2 \qquad (59)$$

$$= \beta_{t,j}\left(\sum_{\mathbf{x}:w_j(\mathbf{x}),\tilde{\mu}_{t,\mathbf{x}}\neq 0}\sqrt{w_j(\mathbf{x})\tilde{\mu}_{t,\mathbf{x}}}\frac{\hat{\hat{\delta}}_{t,j,\mathbf{x}}}{\sqrt{\tilde{\mu}_{t,\mathbf{x}}}}\right)^2 \tag{60}$$

$$\leq \beta_{t,j}\left(\sum_{\mathbf{x}:w_j(\mathbf{x}),\tilde{\mu}_{t,\mathbf{x}}\neq 0} w_j(\mathbf{x})\tilde{\mu}_{t,\mathbf{x}}\right)\left(\sum_{\mathbf{x}:w_j(\mathbf{x}),\tilde{\mu}_{t,\mathbf{x}}\neq 0}\frac{\hat{\hat{\delta}}^2_{t,j,\mathbf{x}}}{\tilde{\mu}_{t,\mathbf{x}}}\right) \tag{61}$$

$$= \left(\sum_{\mathbf{x}} w_j(\mathbf{x})\tilde{\mu}_{t,\mathbf{x}}\right)\tilde{s}_{t,j} \tag{62}$$

(60) holds because that

$$\sum_j \hat{\hat{\delta}}^2_{t,j,\mathbf{x}} \leq (\hat{\mu}^a_{t,\mathbf{x}}+\hat{\mu}^b_{t,\mathbf{x}})(\hat{\mu}^a_{t,\mathbf{x}}-\hat{\mu}^b_{t,\mathbf{x}}) = \tilde{\mu}_{t,\mathbf{x}}(\hat{\mu}^a_{t,\mathbf{x}}-\hat{\mu}^b_{t,\mathbf{x}})$$

by the conic constraint (57), and if $\tilde{\mu}_{t,\mathbf{x}}=0$, $\hat{\hat{\delta}}_{t,j,\mathbf{x}}=0, \forall j$. (61) holds because of Cauchy-Buniakowsky-Schwarz Inequality. Therefore, any feasible solution of $(DSOCP')$ is feasible to $(D1)^{A,R}$, and $Z^{DS} \leq Z^{AR}$.

It is easy to see that the first constraint of $(D1)^{A,R}$ is binding at optimality. We then show that any feasible solution of $(D1)^{A,R}$ with the first constraint binding is feasible to $(DSOCP')$, and thus $Z^{DS} \leq Z^{AR}$. Suppose $(\tilde{\mu}_{t,x}, \tilde{\delta}_{t,j}, \tilde{s}_{t,j})$ is a feasible solution of $(D1)^{A,R}$ with the first constraint binding. We can construct a feasible solution $(\hat{\mu}^a_{t,\mathbf{x}}, \hat{\mu}^b_{t,\mathbf{x}}, \hat{\delta}_{t,j}, \hat{\hat{\delta}}_{t,j,\mathbf{x}})$ to $(DSOCP')$ as the following.

$$\begin{aligned}
\hat{\delta}_{t,j} &= \tilde{\delta}_{t,j},\\
\hat{\hat{\delta}}_{t,j,\mathbf{x}} &= \varphi_{t,j}\sqrt{\beta_{t,j}w_j(\mathbf{x})}\tilde{\mu}_{t,\mathbf{x}},\\
\hat{\mu}^a_{t,\mathbf{x}} &= \begin{cases}\dfrac{\left(\frac{\sum_j \hat{\hat{\delta}}^2_{t,j,\mathbf{x}}}{\tilde{\mu}_{t,\mathbf{x}}}+\tilde{\mu}_{t,\mathbf{x}}\right)}{2}, & \text{if } \tilde{\mu}_{t,\mathbf{x}}\neq 0,\\ 0, & \text{if } \tilde{\mu}_{t,\mathbf{x}}=0.\end{cases}\\
\hat{\mu}^b_{t,\mathbf{x}} &= \begin{cases}\dfrac{\left(\tilde{\mu}_{t,\mathbf{x}}-\frac{\sum_j \hat{\hat{\delta}}^2_{t,j,\mathbf{x}}}{\tilde{\mu}_{t,\mathbf{x}}}\right)}{2}, & \text{if } \tilde{\mu}_{t,\mathbf{x}}\neq 0,\\ 0, & \text{if } \tilde{\mu}_{t,\mathbf{x}}=0.\end{cases}
\end{aligned}$$

where

$$\varphi_{t,j} = \begin{cases}\frac{\tilde{\delta}_{t,j}}{\beta_{t,j}\sum_{\mathbf{x}} w_j(\mathbf{x})\tilde{\mu}_{t,\mathbf{x}}}, & \text{if } \sum_{\mathbf{x}} w_j(\mathbf{x})\tilde{\mu}_{t,\mathbf{x}}\neq 0\\ 0, & \text{Otherwise.}\end{cases}$$

It is easy to see that $\hat{\mu}^a_{t,\mathbf{x}}, \hat{\mu}^b_{t,\mathbf{x}}$ and $\hat{\delta}_{t,j}$ meet Constraints (54) and (55) of $(DSOCP')$. What's more, for $\sum_{\mathbf{x}} w_j(\mathbf{x})\tilde{\mu}_{t,\mathbf{x}}=0$, $\hat{\delta}_{t,j}=\tilde{\delta}_{t,j}=0$ according to the quadratic constraint in $(D1)^{A,R}$. For $\sum_{\mathbf{x}} w_j(\mathbf{x})\tilde{\mu}_{t,\mathbf{x}}\neq 0$,

$$\sum_{\mathbf{x}}\sqrt{\beta_{t,j}w_j(\mathbf{x})}\hat{\hat{\delta}}_{t,j,\mathbf{x}} = \sum_{\mathbf{x}}\sqrt{\beta_{t,j}w_j(\mathbf{x})}\frac{\tilde{\delta}_{t,j}}{\beta_{t,j}\sum_{\mathbf{x}} w_j(\mathbf{x})\tilde{\mu}_{t,\mathbf{x}}}\sqrt{\beta_{t,j}w_j(\mathbf{x})}\tilde{\mu}_{t,\mathbf{x}} = \tilde{\delta}_{t,j}.$$

For $\tilde{\mu}_{t,\mathbf{x}}=0, \hat{\mu}^a_{t,\mathbf{x}}=\hat{\mu}^b_{t,\mathbf{x}}=0, \hat{\hat{\delta}}_{t,j,\mathbf{x}}=0, \forall j$. For $\tilde{\mu}_{t,\mathbf{x}}\neq 0, (\hat{\mu}^a_{t,\mathbf{x}}+\hat{\mu}^b_{t,\mathbf{x}})(\hat{\mu}^a_{t,\mathbf{x}}-\hat{\mu}^b_{t,\mathbf{x}}) = \tilde{\mu}_{t,\mathbf{x}}\frac{\sum_j \hat{\hat{\delta}}^2_{t,j,\mathbf{x}}}{\tilde{\mu}_{t,\mathbf{x}}} = \sum_j \hat{\hat{\delta}}^2_{t,j,\mathbf{x}}$. Therefore (56'),(57) are satisfied. Therefore, $(DSOCP)$ is equivalent to $(D1)^{A,R}$ at optimality.

Appendix E. Proof of Proposition 4

Proof. We start by showing the first inequality. Let L denote a big number. The formulation $(\mathbf{D}2)^{A,R}$ can be written as

$$(\mathbf{D}3)^{A,R} \qquad z^A = \max_{\delta,\mathbf{y},\mathbf{z}} \quad \sum_{t,j} \frac{\alpha_{t,j}}{\beta_{t,j}}\delta_{t,j} - \sum_{t,j:z_{t,j}\neq 0} \frac{\delta_{t,j}^2}{\beta_{t,j} z_{t,j}}$$

$$\text{s.t.} \qquad y_{t,i} = \begin{cases} c_i, & \text{if } t=1, \\ y_{t-1,i} - \sum_j a_{ij}\delta_{t-1,j}, & \text{if } t \geq 2, \end{cases} \qquad \forall\ t,i, \tag{63}$$

$$\delta_{t,j} \leq L z_{t,j}, \qquad \forall\ t,j,$$

$$y_{t,i} \geq a_{ij} z_{t,j}, \qquad \forall\ t,i,j,$$

$$0 \leq z_{t,j} \leq 1, \qquad \forall\ t,j.$$

Let $(\delta^*, \mathbf{y}^*, \mathbf{z}^*)$ be an optimal solution to $(\mathbf{D}3)^{A,R}$. We can show that λ^* defined by $\lambda^*_{t,j} = \delta^*_{t,j}$ for all t,j is a feasible solution to $(\mathbf{DeP})$ with higher corresponding objective value. From constraint (63) in $(\mathbf{D}3)^{A,R}$, we have

$$\sum_{t,j} a_{ij}\lambda^*_{t,j} \leq c_i, \quad \forall\ i.$$

Therefore λ^* is feasible for $(\mathbf{DeP})$. To show the relationship between the objective value, let

$$\bar{z}_{t,j} = \begin{cases} 0, & \text{if } z^*_{t,j} = 0, \\ 1/z^*_{t,j}, & \text{otherwise,} \end{cases} \qquad \forall t,j. \tag{64}$$

Note that since $z^*_{t,j} \leq 1$, $\bar{z}_{t,j} \geq 1$ for all t,j such that $z^*_{t,j} \neq 0$. The objective function for $(\mathbf{D}3)^{A,R}$ can be written as

$$\sum_{t,j} \lambda^*_{t,j}\left(\frac{\alpha_{t,j} - \bar{z}_{t,j}\lambda^*_{t,j}}{\beta_{t,j}}\right),$$

which is smaller than the objective function for $(\mathbf{DeP})$ corresponding to λ^*.

Now, let's turn to the second inequality in the proposition. Suppose $\tilde{\lambda}$ is an optimal solution to the deterministic program $(\mathbf{DeP})$ and let

$$\hat{z}_{t,j} = \frac{1}{\epsilon}, \qquad \forall t,j,$$

$$\hat{\delta}_{t,j} = \frac{\tilde{\lambda}_{t,j}}{\epsilon}, \qquad \forall t,j,$$

where $\epsilon = 1 + \frac{1}{\min_i c_i}$. Furthermore, let $\hat{\mathbf{y}}$ be recursive defined by (63) and $\hat{\delta}$. We can show that $(\hat{\delta}, \hat{\mathbf{y}}, \hat{\mathbf{z}})$ is a feasible solution to $(\mathbf{D}3)^{A,R}$. Since $\hat{y}_{t,i}$ is monotone decreasing in t for each i, we need to show that $\hat{y}_{T,i} \geq a_{ij}\hat{z}_{t,j}$. To see this, note that we have

$$y_{T,i} = y_{1,i} - \sum_{t=2}^{T}\sum_j a_{ij}\tilde{\delta}_{t-1,j}$$

$$
\begin{aligned}
&= c_i - \sum_{t=1}^{T-1}\sum_j a_{ij}\tilde{\lambda}_{t,j}/\epsilon \\
&\geq c_i - \sum_{t=1}^{T-1}\sum_j \frac{a_{ij}\tilde{\lambda}_{t,j}}{\epsilon} - \frac{c_i - \sum_{t=1}^{T}\sum_j a_{ij}\tilde{\lambda}_{t,j}}{\epsilon} \\
&= c_i(1-\frac{1}{\epsilon}) + \frac{\sum_j a_{ij}\tilde{\lambda}_{T,j}}{\epsilon} \\
&\geq c_i(1-\frac{1}{\epsilon}) \\
&\geq 1/\epsilon \\
&\geq a_{ij}\hat{z}_{t,j}.
\end{aligned}
$$

Therefore, we have

$$
z^A \geq \sum_{t,j}\frac{\alpha_{t,j}}{\beta_{t,j}}\hat{\delta}_{t,j} - \sum_{t,j}\frac{\hat{\delta}^2_{t,j}}{\beta_{t,j}\hat{z}_{t,j}} = \sum_{t,j}\frac{\tilde{\lambda}_{t,j}}{\epsilon\beta_{t,j}}(\alpha_{t,j} - \epsilon\frac{\tilde{\lambda}_{t,j}}{\epsilon}) = \frac{1}{\epsilon}\sum_{t,j}\frac{\tilde{\lambda}_{t,j}}{\beta_{t,j}}(\alpha_{t,j} - \tilde{\lambda}_{t,j}) = \frac{1}{\epsilon}z^d.
$$

This completes the proof. ■

Appendix F. Monotonicity of $\Delta\widehat{v}_{t,i}(x_i)$

The dynamic program (37) is a one dimensional dynamic program, but is different from the dynamic programming model for single resource revenue management (Lautenbacher and Stidham 1999) because of the optimization over $\mathbf{x}_{-i}$ and the additional term on the right hand side. Nevertheless, we can still show some familiar structural properties of the value function akin to those for the single resource revenue management problems. The result is summarized in the following proposition. We include a proof in the Appendix F for completeness.

Proposition 8. *For each i, $\Delta\widehat{v}_{t,i}(x_i) = \widehat{v}_{t,i}(x_i) - \widehat{v}_{t,i}(x_i - 1)$ is non-increasing in x_i and t.*

Proof. The proof is by induction for each fixed i. We first show the monotonicity with respect to x_i.

For $t = T$, $\Delta\widehat{v}_{T,i}(x_i) - \Delta\widehat{v}_{T,i}(x_i+1) \geq 0$ because

$$
\begin{aligned}
\widehat{v}_{T,i}(x_i) = \max_{x_{-i},\mathbf{r}_T}\Bigg\{\sum_j \lambda_{T,j}(\mathbf{r}_T)\Bigg[r_{T,j} + \widehat{v}_{T+1,i}(x_i - a_{ij}) - \widehat{v}_{T+1,i}(x_i) - \sum_{k\neq i} V^*_{T+1,k}a_{kj}\Bigg] \\
- \sum_{k\neq i}(V^*_{T,k} - V^*_{T+1,k})x_k\Bigg\} + \widehat{v}_{T+1,i}(x_i) = \max_{x_{-i},\mathbf{r}_T}\Bigg\{\sum_j \lambda_{T,j}(\mathbf{r}_T)r_{T,j} - \sum_{k\neq i} V^*_{T,k}x_k\Bigg\},
\end{aligned}
$$

which does not depend on x_i.

Now suppose $\Delta\widehat{v}_{t+1,i}(x_i) - \Delta\widehat{v}_{t+1,i}(x_i+1) \geq 0$ for some $t < T$. Let $(\mathbf{x}^*_{-i}, \mathbf{r}^*_t)$, $(\mathbf{x}'_{-i}, \mathbf{r}'_t)$, $(\mathbf{x}''_{-i}, \mathbf{r}''_t)$ be optimal solutions for $\widehat{v}_{t,i}(x_i+1)$, $\widehat{v}_{t,i}(x_i)$, and $\widehat{v}_{t,i}(x_i-1)$, respectively. Then

$$
\Delta\widehat{v}_{t,i}(x_i) - \Delta\widehat{v}_{t,i}(x_i+1) = 2\widehat{v}_{t,i}(x_i) - \widehat{v}_{t,i}(x_i+1) - \widehat{v}_{t,i}(x_i-1)
$$

$$
\begin{aligned}
= \;& 2\left\{\sum_j \lambda_{t,j}(\mathbf{r}'_t)\left[r'_{t,j}+\widehat{v}_{t+1,i}(x_i-a_{ij})-\widehat{v}_{t+1,i}(x_i)-\sum_{k\neq i}V^*_{t+1,k}a_{kj}\right]-\sum_{k\neq i}(V^*_{t,k}-V^*_{t+1,k})x'_k\right\}+2\widehat{v}_{t+1,i}(x_i)\\
&-\left\{\sum_j \lambda_{t,j}(\mathbf{r}^*_t)\left[r^*_{t,j}+\widehat{v}_{t+1,i}(x_i+1-a_{ij})-\widehat{v}_{t+1,i}(x_i+1)-\sum_{k\neq i}V^*_{t+1,k}a_{kj}\right]-\sum_{k\neq i}(V^*_{t,k}-V^*_{t+1,k})x^*_k\right\}-\widehat{v}_{t+1,i}(x_i+1)\\
&-\left\{\sum_j \lambda_{t,j}(\mathbf{r}''_t)\left[r''_{t,j}+\widehat{v}_{t+1,i}(x_i-1-a_{ij})-\widehat{v}_{t+1,i}(x_i-1)-\sum_{k\neq i}V^*_{t+1,k}a_{kj}\right]-\sum_{k\neq i}(V^*_{t,k}-V^*_{t+1,k})x''_k\right\}-\widehat{v}_{t+1,i}(x_i-1)\\
\geq \;& \left\{\sum_j \lambda_{t,j}(\mathbf{r}^*_t)\left[\widehat{v}_{t+1,i}(x_i-a_{ij})-\widehat{v}_{t+1,i}(x_i)-\widehat{v}_{t+1,i}(x_i+1-a_{ij})+\widehat{v}_{t+1,i}(x_i+1)\right]\right\}+\widehat{v}_{t+1,i}(x_i)-\widehat{v}_{t+1,i}(x_i+1)\\
&+\left\{\sum_j \lambda_{t,j}(\mathbf{r}''_t)\left[\widehat{v}_{t+1,i}(x_i-a_{ij})-\widehat{v}_{t+1,i}(x_i)-\widehat{v}_{t+1,i}(x_i-1-a_{ij})+\widehat{v}_{t+1,i}(x_i-1)\right]\right\}+\widehat{v}_{t+1,i}(x_i)-\widehat{v}_{t+1,i}(x_i-1)\\
= \;& \sum_{j:a_{ij}=1}\lambda_{t,j}(\mathbf{r}^*_t)\left(\Delta\widehat{v}_{t+1,i}(x_i+1)-\Delta\widehat{v}_{t+1,i}(x_i)\right)-\Delta\widehat{v}_{t+1,i}(x_i+1)+\\
&\sum_{j:a_{ij}=1}\lambda_{t,j}(\mathbf{r}''_t)\left(\Delta\widehat{v}_{t+1,i}(x_i-1)-\Delta\widehat{v}_{t+1,i}(x_i)\right)+\Delta\widehat{v}_{t+1,i}(x_i)\\
\geq \;& \left(\Delta\widehat{v}_{t+1,i}(x_i)-\Delta\widehat{v}_{t+1,i}(x_i+1)\right)\left(1-\sum_{j:a_{ij}=1}\lambda_{t,j}(\mathbf{r}^*_t)\right)\geq 0.
\end{aligned}
$$

The first inequality is satisfied because $\widehat{v}_{t,i}(x_i)$ decreases if we use $(\mathbf{x}^*_{-i},\mathbf{r}^*_t)$ or $(\mathbf{x}_{-i}", \mathbf{r}_t")$ instead of the optimal solution $(\mathbf{x}'_{-i},\mathbf{r}'_t)$ in the right hand side of the dynamic programming equation. The second inequality is satisfied because $\Delta\widehat{v}_{t+1,i}(x_i-1)-\Delta\widehat{v}_{t+1,i}(x_i)\geq 0$ for $x_i>1$ according to the inductive assumption. The last inequality is satisfied because $\sum_{j:a_{ij}=1}\lambda_{t,j}(\mathbf{r}''_t)\leq\sum_j\lambda_{t,j}(\mathbf{r}''_t)\leq 1$ and $\Delta\widehat{v}_{t+1,i}(x_i)-\Delta\widehat{v}_{t+1,i}(x_i+1)\geq 0$ according to the inductive assumption. This shows the monotonicity with respect to x_i.

Next, we show the monotonicity in time. Again, fix i. For $t<T$ and $x_i>1$, we have

$$
\begin{aligned}
&\Delta\widehat{v}_{t,i}(x_i)=\widehat{v}_{t,i}(x_i)-\widehat{v}_{t,i}(x_i-1)\\
= \;& \left\{\sum_j \lambda_{t,j}(\mathbf{r}^*_t)\left[r^*_{t,j}+\widehat{v}_{t+1,i}(x_i-a_{ij})-\widehat{v}_{t+1,i}(x_i)-\sum_{k\neq i}V^*_{t+1,k}a_{kj}\right]-\sum_{k\neq i}(V^*_{t,k}-V^*_{t+1,k})x^*_k\right\}+\widehat{v}_{t+1,i}(x_i)-\\
&\left\{\sum_j \lambda_{t,j}(\mathbf{r}'_t)\left[r'_{t,j}+\widehat{v}_{t+1,i}(x_i-1-a_{ij})-\widehat{v}_{t+1,i}(x_i-1)-\sum_{k\neq i}V^*_{t+1,k}a_{kj}\right]-\sum_{k\neq i}(V^*_{t,k}-V^*_{t+1,k})x'_k\right\}-\widehat{v}_{t+1,i}(x_i-1)\\
\geq \;& \sum_j \lambda_{t,j}(\mathbf{r}'_t)\left[\widehat{v}_{t+1,i}(x_i-a_{ij})-\widehat{v}_{t+1,i}(x_i)-\widehat{v}_{t+1,i}(x_i-1-a_{ij})+\widehat{v}_{t+1,i}(x_i-1)\right]+\Delta\widehat{v}_{t+1,i}(x_i)\\
= \;& \Delta\widehat{v}_{t+1,i}(x_i)+\sum_j\lambda_{t,j}(\mathbf{r}'_t)\left(\Delta\widehat{v}_{t+1,i}(x_i-a_{ij})-\Delta\widehat{v}_{t+1,i}(x_i)\right)\\
= \;& \Delta\widehat{v}_{t+1,i}(x_i)+\left(\Delta\widehat{v}_{t+1,i}(x_i-1)-\Delta\widehat{v}_{t+1,i}(x_i)\right)\sum_{j:a_{ij}=1}\lambda_{t,j}(\mathbf{r}'_t).
\end{aligned}
$$

Because $\sum_{j:a_{ij}=1}\lambda_{t,j}(\mathbf{r}'_t)$ is non-negative and $\Delta\widehat{v}_{t+1,i}(x_i-1)\geq\Delta\widehat{v}_{t+1,i}(x_i)$, $\Delta\widehat{v}_{t,i}(x_i)-\Delta\widehat{v}_{t+1,i}(x_i)\geq 0$. This completes the proof. ■