On Bounds for Network Revenue Management

Kalyan Talluri *

May 25, 2009

Abstract

The Network Revenue Management problem can be formulated as a stochastic dynamic programming problem (DP or the "optimal" solution V^*) whose exact solution is computationally intractable. Consequently, a number of heuristics have been proposed in the literature, the most popular of which are the deterministic linear programming (DLP) model, and a simulation based method, the randomized linear programming (RLP) model. Both methods give upper bounds on the optimal solution value (DLP and $PHLP$ respectively). These bounds are used to provide control values that can be used in practice to make accept/deny decisions for booking requests. Recently Adelman [1] and Topaloglu [18] have proposed alternate upper bounds, the affine relaxation (AR) bound and the Lagrangian relaxation (LR) bound respectively, and showed that their bounds are tighter than the DLP bound. Tight bounds are of great interest as it appears from empirical studies and practical experience that models that give tighter bounds also lead to better controls (better in the sense that they lead to more revenue). In this paper we give tightened versions of three bounds, calling them sAR (strong Affine Relaxation), sLR (strong Lagrangian Relaxation) and $sPHLP$ (strong Perfect Hindsight LP), and show relations between them. Specifically, we show that the $sPHLP$ bound is tighter than sLR bound and sAR bound is tighter than the LR bound. The techniques for deriving the sLR and $sPHLP$ bounds can potentially be applied to other instances of weakly-coupled dynamic programming.

Key words. revenue management, bid-prices, relaxations, bounds.

1 Introduction

Revenue management is the control of the sale of a limited quantity of a resource (hotel rooms for a night, airline seats, advertising slots etc.) to a heterogenous population with different valuations for a unit of the resource. The resource is perishable, and for simplicity sake, we assume that it perishes at a fixed point of time in the future. Customers are independent of each other and arrive randomly during a sale period, and demand one unit of resource each. Sale is online, so the firm has to decide at the time of each customer's arrival if it wishes to sell (at a specific price) or not, the tradeoff being selling too much at too low a price early and running out of capacity, or, rejecting too many low valuation customers and ending up with excess unsold inventory. That is a brief description of revenue management. The reader should consult the books by Talluri and van Ryzin [15] or Phillips [10] or the survey articles of McGill and van Ryzin [9], Elmaghraby and Keskinocak [5], and

*Kalyan Talluri, ICREA and UPF, Universitat Pompeu Fabra, Ramon Trias Fargas 25-27, 08005 Barcelona, Spain, email: kalyan.talluri@upf.edu

Bitran and Caldentey [4] for a background on the theory and a survey of applications of revenue management.

In industries such as hotels and airlines the products consume bundles of different resources (multi-night stays, multi-leg itineraries) and the decision to accept or reject a particular product at a certain price depends on the future demands and revenues for all the resources used by the product (and also indirectly, all the resources in the network). Network revenue management (network RM) is control based on the demands for the entire network. Chapter 3 of Talluri and vanRyzin [15] contains all the necessary background on network RM.

The network revenue management problem, under certain assumptions on the demand process, can be formulated as a stochastic dynamic programming problem. The controls of this dynamic program are subsets of products to open up for sale at any given point in time. The state is time remaining and the remnant capacities on the resources. This dynamic program is practically impossible to solve for even small networks. So a number of heuristics have been devised that generate controls to make the accept/reject decisions, of which the oldest and most well-known and widely used are the the Deterministic Linear Programming (DLP) , proposed by Simpson [11] and analyzed in Williamson [19], the Randomized Linear Programming (RLP) method in Talluri and van Ryzin [13], and the Displacement Adjusted Virtual Nesting ($DAVN$) heuristic of Belobaba [3]. The mathematical programs on which DLP and RLP are based can be shown to form upper bounds on the optimal value.

Recently, Adelman [1] and Topaloglu [18] have proposed new heuristics for network RM, based on an affine approximation to the linear programming formulation of the dynamic program, and a Lagrangian relaxation of the dynamic program respectively. Both show that their formulations give provably tighter upper bounds on the optimal solution than the DLP upper bound.

Tight bounds are of great interest as it appears from empirical studies and practical experience that models that give tighter bounds also lead to better controls (better in the sense that they lead to more revenue). In this paper we give tightened versions of three bounds, calling them sAR (strong Affine Relaxation), sLR (strong Lagrangian Relaxation) and $sPHLP$ (strong Perfect Hindsight LP), and show relations between them. Specifically, we show that the $sPHLP$ bound is tighter than sLR bound and sAR bound is tighter than the LR bound. The technique for deriving the sLR and $sPHLP$ bounds can potentially be applied to other instances of weakly-coupled dynamic programming.

2 The optimal dynamic program

2.1 Notation

A product is a specification of a combination of resources and a price. (For example an itinerary-fare class combination for an airline network.) We assume that the booking horizon begins at time 0 and all the resources are consumed instantaneously at time τ. Time is discrete and assumed to consist of τ intervals, indexed by t. We make the standard assumption that the intervals are fine enough so that at most one customer arrives in each period.

The underlying network has m resources and n products. The current capacity on resource i at time t is $r_{i,t}$ and the vector of capacities r_t, so the initial set of capacities at time 0 is r_0. Let $\bar{r}_0 = \max_i(|r_{i,0}|$. Products are indexed by j and resources by i. The revenue from product j is f_j.

A resource i is said to be in product j ($i \in j$) if j uses resource i. We represent this by $a_{i,j} = 1$ if $i \in j$, and $a_{i,j} = 0$ if $i \notin j$, or alternately with the 0-1 incidence vector a_j of product j. For both r and a, the number of indices and context should make it clear if it is a scalar or a vector. An arrival is a purchase request for a product in a specific interval of time.

We represent a mathematical program or a dynamic program by a label that also serves as the value of the program. For example, (DLP) represents the deterministic linear program (described below) and DLP represents the objective function value of the optimal solution.

2.2 Demand Model

The future demand-to-come for each product is a random variable with a known distribution. The distribution and the parameters of this distribution are assumed to be known (statistically speaking) by some process of estimation and forecasting before the optimization.

In period t, a request for product j appears with probability $p_{j,t}$. Our assumption of at most one arrival per unit time translates to $\sum_j p_{j,t} \ll 1$. We will assume that the demands for the different products are independent of each other, independent across time and across products.

As in [18], to make the notation simpler, we add a dummy product $\jmath$ in each period with $a_{i,\jmath} = 0$ for all resources i and revenue 0 with an arrival probability of $1 - \sum_j p_{j,t}$ in period t. So from now on, we assume $\sum_j p_{j,t} = 1$ for all periods t.

The customer behavior assumption is simple: customer who wants product j, if it is unavailable (sale closed by the firm), will simply not purchase anything and disappear, rather than purchasing another available product. Note moreover that demand is specified at the product level. Together, these assumptions go under the name of the *independent class assumption* in the RM community.

We shall be discussing simulation-based methods in this paper. The idea is to simulate the future using our forecasts—the implicit assumption being demand does in fact follows our distributional assumptions and, statistically, the forecasts. When put this way, it might seem a stretch, but it is actually no different from optimizing the expected value given certain distributional assumptions.

An *instance* (sample path) is a single set of simulated realizations of all the demands (a sample path) for all the products from 0 to τ. A large number of instances are generated to capture the variance in the distribution, and each instance-specific data will be indexed with a superscript k. For example: the demand for product j in time period t generated in instance k would be $p^k_{j,t}$, capacity on resource i, at time t in instance k (under some control) is represented by $r^k_{i,t}$, etc.

2.3 Controls

For a capacity vector r_t, define the set of all "acceptable" products as $\mathcal{A}_{r_t} = \{j | a_j \leq r_t\}$. The control at time t is a subset of acceptable products that are *open*; i.e., given the remnant capacity vector r_t at time t, we can accept a certain set of products but based on our forecasts of demand to come, we may *close* some of these products for sale and open the rest of the acceptable set for sale.

Let $\mathcal{U}_{r_t}$ denote the collection of all subsets of $\mathcal{A}_{r_t}$, i.e., the feasible controls. For a $u \in \mathcal{U}_{r_t}$, define u_{r_t} as the incidence vector of the elements of u (a vector of size n, the number of products). If the context is clear, we write $u_{j,t}$ as the j'th element of this vector and $u_{i,j,t} = a_{i,j} u_{j,t}$. We use a different notation when the incidence vector is a vector of decision variables — y instead of u ($y_{j,t}$

etc.).

As we deal with decomposition strategies, where the network problem is decomposed into single-resource problems, we define the controls at the level of resource i as follows: Define $\mathcal{A}_{r_{i,t}} = \{j | a_{i,j} \leq r_{i,t}\}$. Define $\mathcal{U}_{i,r_{i,t}}$ as a collection of all subsets of $\{j \in \mathcal{A}_{r_{i,t}} | i \in j\}$. $u_{i,j,t}$ are the elements of the incidence vector of $u \in \mathcal{U}_{i,r_{i,t}}$.

Note that the level of control we are defining is the most general possible, and not necessarily the RM controls used in practice, where a more restrictive nested structure or a threshold price structure called bid-price control is usually imposed[1], due to industry practice, tractability or system limitations.

2.4 Dynamic Program

The dynamic program to determine optimal controls $u^*_{r_t}$ is as follows:

Let $V_t(r_t)$ denote the maximum expected revenue to go, given remaining capacity r_t in period t. Then $V_t(r_t)$ must satisfy the Bellman equation

$$V_t(r_t) = \max_{u \in \mathcal{U}_{r_t}} \left\{ \sum_j p_{j,t} \{ f_j u_{j,t} + V_{t+1}(r_t - u_{j,t} a_j) \} \right\} \tag{1}$$

with the boundary condition

$$V_{\tau+1}(r) = 0, \ \forall r. \tag{2}$$

Let $V^*(r_0)$ denote the optimal value of this dynamic program from 0 to τ, for the given initial capacity vector r_0.

3 Classical bounds

The deterministic linear program and the randomized linear program reviewed here serve as benchmark methods in network revenue management.

3.1 Deterministic Linear Program

One of the earliest methods proposed for generating network bid-prices is a simple and compact linear program that generally goes by the name of Deterministic Linear Program (DLP). See Chapter 3.3.1 of [15] for a discussion of this method and its variants. The method consists in solving the following linear program

$$\max \quad \sum_{j,t} f_j y_{j,t} \tag{3}$$

[1] Roughly, a bid-price control is as follows: at any given time t, a bid-price, a non-negative real number is associated with each resource and a request for product j is accepted if the sum of the bid-prices on the resources that j consumes is less than the revenue from j.

$$(DLP) \quad \text{s.t.} \quad \sum_t \sum_j a_{i,j} y_{j,t} \leq r_{i,0} \ \ \forall i$$
$$0 \leq y_{j,t} \leq p_{j,t}$$

and using the dual prices as the bid-prices for control.

DLP is quite popular as it is very easy to program and can be solved quickly using any off-the-shelf LP software package. Its performance is quite reasonable, and often serves as the benchmark method in simulation comparisons.

3.2 Randomized Linear Program

Consider a simulation where we generate N instances from the demand data. Each such generated instance k leads to a perfect-hindsight linear program $PHLP^k$ as follows. Let $p^k_{j,t} = 1$ if j is generated at time t in instance k and 0 otherwise.

$$\begin{aligned} & \max && \sum_{j,t} f_j p^k_{j,t} y^k_{j,t} \\ (PHLP^k) \quad & \text{s.t.} && \sum_t \sum_{j \ni i} y^k_{j,t} \leq r_{i,0} \ \ \forall i \\ & && 0 \leq y^k_{j,t} \leq 1 \end{aligned} \tag{4}$$

Our formulation of $(PHLP^k)$ is slightly unconventional and different from the DLP formulation for instance k — notice that we have moved $p^k_{j,t}$ from the constraints to the objective function. As $p^k_{j,t}$ is either 0 or 1, this is equivalent to having the constraint $0 \leq y^k_{j,t} \leq p^k_{j,t}$ in the more conventional formulation.

The randomized linear programming method RLP takes the average of the dual prices of $PHLP^k$ corresponding to resource i as the bid-price for resource i.

Define $PHLP$ as the average of the objective values of the linear programs $(PHLP^{(k)})$. Define $PHIP$ as the average over the integer program versions of $(PHLP^{(k)})$ where we add the restriction $y^k_{j,t} \in \{0, 1\}$ (variables of the corresponding integer program that we call $(PHIP^{(k)})$).

We do not specify the choice of the number of generated sample paths, N, but just assume it is large enough and equal to $n_1 \times n_2 \times \cdots \times n_t \times \cdots \times n_\tau$, where n_t are samples of arrivals drawn in period t. We make this more precise at the beginning of Section 6

Also, in [15] the RLP method (and therefore $PHLP$) is presented in terms of taking expectation over a sample path. We explicitly index the generated instances because in a proof later on we need to link different instances by constraints.

The fact that DLP and RLP are upper bounds on $V^*(r_0)$ is quite easy to prove; see for instance [17].

4 New bounds

Recently Adelman [1] and Topaloglu [18] have proposed new bounds for the network RM problem based respectively on an affine approximation of the dynamic program (the AR bound) and a

Lagrangian relaxation approach to dynamic programming (the LR bound).

4.1 The LR bound

Let the decision variable $y_{i,j,t} = 1$ if we accept a request for product j on resource i in period t and 0 otherwise. Let e_i be a m dimensional vector with 1 in position i and 0 elsewhere. The optimality condition (1) can be written as

$$
\begin{aligned}
& V_t(r_t) = \max && \sum_j p_{j,t}\{f_j y_{\imath,j,t} + V_{t+1}(r_t - \sum_{i\in j} y_{i,j,t} e_i\} & \\
(DP) \quad & \text{s.t.} && y_{i,j,t} \le r_{i,t} \quad \forall j, \forall i \in j & (5)\\
& && y_{\imath,j,t} - y_{i,j,t} = 0 \quad \forall j, \forall i \in j & (6)\\
& && y_{i,j,t} \in \{0,1\}. &
\end{aligned}
$$

Topaloglu in [18], augments the set of legs by a dummy resource $\imath$ with infinite capacity, and sets all products to use one unit of this dummy resource $\imath$. So $y_{\imath,j,t}$ is an extra, dummy, unrestricted variable whose sole purpose is to impose the condition that we either accept a product on all resources, or reject it on all the resources that the product uses.

Notice that the constraints are constraints of a recursive dynamic program, so the variable $y_{i,j,t}$ has to be interpreted as a state-dependent variable and strictly speaking ought to be written as y_{i,j,t,r_t}, i.e., one for each possible state r_t at time t, i and j, as there are a set of constraints for all possible states in the future. If there is any reason for confusion, we explicitly write them so. The set of products j with $y_{i,j,t,r_t} = 1$ represents the set of products on resource i open for sale at time t given a state r_t.

Now relax the constraints of the form $y_{\imath,j,t} - y_{i,j,t} = 0$ with a set of Lagrange multipliers $\lambda = \{\lambda_{i,j,t}\}$ to obtain:

$$
\begin{aligned}
& V_t^\lambda(r_t) = \max \textstyle\sum_j \{p_{j,t} f_j y_{\imath,j,t} - \sum_{i\in j} \lambda_{i,j,t} y_{\imath,j,t} + \sum_{i\in j} \lambda_{i,j,t} y_{i,j,t} + \\
& \qquad\qquad p_{j,t} V_{t+1}^\lambda(r_t - \textstyle\sum_{i\in j} y_{i,j,t} e_i)\} \\
& \text{s.t.} \\
(LR) \quad & \qquad y_{i,j,t} \le r_{i,t} \quad \forall j, \forall i \in j \qquad (7)\\
& \qquad y_{i,j,t} \in \{0,1\}.
\end{aligned}
$$

Topaloglu ([18]) shows that (7) break up into resource-level dynamic programs. We simplify this result slightly by showing later (Proposition 4) that one can assume that the optimal multipliers satisfy $\sum_{i\in j} \lambda_{i,j,t} = p_{j,t} f_j$—which makes the decomposition transparent, as the objective function and the constraints separate by resource.

There are a few small differences from the original formulation of [18]:

- Instead of scaling the multipliers by $p_{j,t}$ as in [18] we use the unscaled version for clarity.
- $\lambda_{i,j,t}$ are defined only for $i \in j$.

Define the LR bound $LR = \min_\lambda V_0^\lambda(r_0)$. Topaloglu [18] shows:

$$DLP \ge LR \ge V^*.$$

The main advantage of Topaloglu's method is that a relatively small set of Lagrange multipliers are used to relax the dynamic program. Another interpretation is that a single set of state-independent but time and product-dependent Lagrange multipliers are used for all future time periods, reducing the state-space of the multipliers. And once relaxed, the problem decomposes into resource-level problems with much smaller state spaces.

4.2 The AR bound

Adelman's idea is to first formulate the dynamic program as a linear program and then use an affine relaxation (as a special case of a broader proposal of relaxations based on basis functions) of the dynamic programming value functions.

The dynamic program (1) can be formulated as a linear program, albeit with a prohibitively large set of variables and constraints, as follows. The decision variables are $V_t(r_t)$, one for each possible state vector r_t at time t.

$$\begin{aligned} V^*(r_0) = \min_{t,V_t(r_t)} \quad & V_0(r_0) \\ \text{s.t.} \quad & V_t(r_t) \geq \sum_{j \in u} p_{j,t} f_j u_{j,t} + \sum_j p_{j,t} V_{t+1}(r_t - a_j u_{j,t}) \ \ \forall t, r_t, u \in \mathcal{U}_{r_t} \\ & V_\tau(\cdot) = 0 \end{aligned} \tag{8}$$

The linear program (8) has an exponential number of variables and constraints. The AR relaxation imposes a specific affine functional form on the variables:

$$V_t(r_t) \approx \theta_t + \sum_i r_{i,t} v_{i,t}, \ \ \forall t, r_t. \tag{9}$$

So the number of variables at least are reduced to number of resources multiplied by the number of periods. The number of constraints however remains very large, but techniques such as column-generation can be brought to bear on a solution. A rough interpretation of $V_{i,t}$ is that it represents the marginal value of resource i at time t. Substituting (9) into (8) gives the linear program

$$\begin{aligned} & \min_{\theta, v_{i,t}} \quad \theta_0 + \sum_i r_{i,0} v_{i,0} \\ (AR) \quad & \text{s.t.} \\ & \theta_t - \theta_{t+1} + \sum_i \{ r_{i,t} v_{i,t} - (\sum_j p_{j,t}(r_{i,t} - u_{i,j,t})) v_{i,t+1} \} \geq \sum_{j \in u} p_{j,t} f_j \ \ \forall t, r_t, u \in \mathcal{U}_{r_t} \\ & v_{i,t} \geq 0 \end{aligned}$$

Adelman [1] shows:

$$DLP \geq AR \geq V^*.$$

5 sLR

In this section we tighten the Lagrangian bound and define informally the connection between the dynamic program, policies and $PHLP$.

5.1 Strong Lagrangian Bound (sLR)

As we mentioned at the end of Section 4.1 the Lagrangian multipliers used to decompose the DP are state-independent—for each product j and time t, the $\lambda_{i,j,t}, i \in j$ are constants across all states. One natural idea to try is to make these state-dependent somehow while keeping the number of multipliers to a reasonable level. To this end, relax the constraints of (6) by multipliers $\lambda_{i,j,t,r_{i,t}}$, where $r_{i,t}$ is the capacity on leg i at time t. Notice that the multipliers do not depend on the state of the network r_t, but on the capacities at the resource-level, so the number of multipliers is $mn\tau\bar{r}_0$. We label these (partially) state-dependent multipliers as λ^r. The relaxation becomes:

$$\begin{aligned} V_t^{\lambda^r}(r_t) = \max \textstyle\sum_j \{ & [p_{j,t} f_j - \textstyle\sum_{i \in j} \lambda_{i,j,t,r_{i,t}}] y_{\imath,j,t} + \textstyle\sum_{i \in j} \lambda_{i,j,t,r_{i,t}} y_{i,j,t} + \\ & p_{j,t} V_{t+1}^{\lambda^r}(r_t - \textstyle\sum_{i \in j} y_{i,j,t} e_i)\} \\ \text{s.t.} & \\ & y_{i,j,t} \le r_{i,t} \quad \forall j, \forall i \in j \\ & y_{i,j,t} \in \{0,1\} \end{aligned} \tag{10}$$

However, one quickly sees that (10) does not decompose by the resources because of the term $[p_{j,t} f_j - \sum_{i \in j} \lambda_{i,j,t,r_{i,t}}] y_{\imath,j,t}$. The summation is dependent on the state vector r_t, so even though it gives a stronger bound than LR, it is not computable. We describe below two approximations to (10) that make it computationally tractable.

5.1.1 Decomposing the problem

To make (10) tractable, one natural idea to try is to introduce new variables $w_{i,j,t}$ that satisfy the equation $f_j = \sum_{i \in j} w_{i,j,t}$ and substitute into $V_t^{\lambda^r}(r_t)$. So the term in the square brackets becomes $[p_{j,t} \sum_{i \in j} w_{i,j,t} - \sum_{i \in j} \lambda_{i,j,t,r_{i,t}}]$. As in Topaloglu [18] we can eliminate $y_{\imath,j,t}$ by replacing

$$[p_{j,t} \sum_{i \in j} w_{i,j,t} - \sum_{i \in j} \lambda_{i,j,t,r_{i,t}}] y_{\imath,j,t}$$

by

$$[p_{j,t} \sum_{i \in j} w_{i,j,t} - \sum_{i \in j} \lambda_{i,j,t,r_{i,t}}]^+,$$

where $[x]^+ = \max\{0, x\}$. Now observe that

$$[p_{j,t} \sum_{i \in j} w_{i,j,t} - \sum_{i \in j} \lambda_{i,j,t,r_{i,t}}]^+ \le [\sum_{i \in j} [p_{j,t} w_{i,j,t} - \lambda_{i,j,t,r_{i,t}}]^+]^+ = \sum_{i \in j} [p_{j,t} w_{i,j,t} - \lambda_{i,j,t,r_{i,t}}]^+$$

So we replace $[p_{j,t} f_j - \sum_{i \in j} \lambda_{i,j,t,r_{i,t}}] y_{i,j,t}$ by its upper bound $\sum_{i \in j} [p_{j,t} w_{i,j,t} - \lambda_{i,j,t,r_{i,t}}]^+$ and we get a bound which does decompose by resource, and hence is computable.

For a given $w_{i,j,t}, f_j = \sum_{i \in j} w_{i,j,t}, \forall j, t$, define

$$\begin{aligned} V_t^{w,\lambda^r}(r_t) = \max \textstyle\sum_j \{ & \textstyle\sum_{i \in j} [p_{j,t} w_{i,j,t} - \lambda_{i,j,t,r_{i,t}}]^+ + \textstyle\sum_{i \in j} \lambda_{i,j,t,r_{i,t}} y_{i,j,t} + \\ & p_{j,t} V_{t+1}^{w,\lambda^r}(r_t - \textstyle\sum_{i \in j} y_{i,j,t} e_i)\} \\ \text{s.t.} & \\ (LR(w, \lambda^r)) \quad & y_{i,j,t} \le r_{i,t} \quad \forall j, \forall i \in j \\ & y_{i,j,t} \in \{0,1\}. \end{aligned} \tag{11}$$

and then we find the best bound by calculating $\min_{\{\lambda^r, w | f_j = \sum_{i\in j} w_{i,j,t}, \forall j,t\}} V_0^{w,\lambda^r}(r_0)$; let λ^{*r}, w^* denote the minimizers. Unfortunately, we do not improve over LR.

Proposition 1 $LR = \min_{\{\lambda^r, w | f_j = \sum_{i\in j} w_{i,j,t}, \forall j,t\}} V_0^{w,\lambda^r}(r_0)$.

<u>Proof</u>
As we show in Proposition 4 in the next section, the optimal Lagrangian multipliers of (LR) can be assumed to satisfy $\sum_{i\in j} \lambda_{i,j,t} = f_j p_{j,t},\ \forall j,t$. So LR corresponds to a specific choice $w_{i,j,t} = \frac{\lambda_{i,j,t}}{p_{j,t}}$ and $\lambda_{i,j,t,r_{i,t}} = \lambda_{i,j,t}$, and hence $LR \leq V_0^{w^*,\lambda^{*r}}(r_0)$.

In the other direction, in LR, set $\lambda^*_{i,j,t}$ of LR equal to $p_{j,t} w^*_{i,j,t}$. For any given state r_t , in LR the revenue obtained from j from resource $i \in j$ in that state is $\lambda^*_{i,j,t} y_{i,j,t} = p_{j,t} w^*_{i,j,t} y_{i,j,t}$; whereas, in (11), the revenue from j from resource $i \in j$ in that state is

$$[p_{j,t} w^*_{i,j,t} - \lambda^*_{i,j,t,r_{i,t}}]^+ + \lambda^*_{i,j,t,r_{i,t}} y_{i,j,t} \geq p_{j,t} w^*_{i,j,t} y_{i,j,t}.$$

Q.E.D

Even though we could not improve over LR, formulation $(LR(w, \lambda^r))$ is useful as it shows a way to tighten LR bound.

5.1.2 Another attempt at decomposing the problem

LR can be tightened by obtaining the optimal separable approximation to the troublesome quantity $[p_{j,t} f_j - \sum_{i\in j} \lambda_{i,j,t,r_{i,t}}]^{[+]}$ as follows: Let variables $w_{i,j,t,r_{i,t}}$ satisfy $\sum_{i\in j} w_{i,j,t,r_{i,t}} \geq 0$ and $\sum_{i\in j} w_{i,j,t,r_{i,t}} \geq [p_{j,t} f_j - \sum_{i\in j} \lambda_{i,j,t,r_{i,t}}]$ for all states r_t; We replace $[p_{j,t} f_j - \sum_{i\in j} \lambda_{i,j,t,r_{i,t}}]^{[+]}$ by $\sum_{i\in j} w_{i,j,t,r_{i,t}}$.

This approximation is clearly an upper bound on (10) as in each state we are tallying a quantity strictly greater than in (10) while keeping all other parameters exactly the same. So fixing a policy and summing the transition and state-dependent rewards, we would end up with a higher value function in this approximation.

We define sLR (strengthened LR or optimal separable LR) as follows: Define variables $\lambda_{i,j,t,r_{i,t}}$ and $w_{i,j,t,r_{i,t}}$, such that

$$\sum_{i\in j} w_{i,j,t,r_{i,t}} \geq 0 \tag{12}$$

and

$$\sum_{i\in j} w_{i,j,t,r_{i,t}} \geq [p_{j,t} f_j - \sum_{i\in j} \lambda_{i,j,t,r_{i,t}}] \tag{13}$$

for all states r_t; and the dynamic program

$$V_t^{w^r,\lambda^r}(r_t) = \max \textstyle\sum_j \{\sum_{i\in j} w_{i,j,t,r_{i,t}} + \sum_{i\in j} \lambda_{i,j,t,r_{i,t}} y_{i,j,t} + p_{j,t} V_{t+1}^{w^r,\lambda^r}(r_t - \sum_{i\in j} y_{i,j,t} e_i)\}$$

s.t.

$$y_{i,j,t} \leq r_{i,t} \quad \forall j, \forall i \in j \tag{14}$$
$$y_{i,j,t} \in \{0,1\}.$$

Then, define $sLR = \min_{\{w_{i,j,t,r_{i,t}},\lambda_{i,j,t,r_{i,t}},t=0,\ldots,\tau,\forall r_{i,t}\}} V_t^{w^r,\lambda^r}(r_0)$, subject to (12) and (13).

The decomposition procedure of the previous section combined with Proposition 1 shows

Proposition 2 *$sLR \leq LR$.*

sLR decomposes by resource for a fixed $w_{i,j,t,r_{i,t}}, \lambda_{i,j,t,r_{i,t}}$. While the size of the problem is much larger than (LR), it has the same convexity properties as (LR) (convex in λ and w as $V_t^{w^r,\lambda^r}(r_t)$ can be formulated as a linear program where all the w's and λ's appear only on the right hand-side of the linear program). So it can be solved by using the same sub-gradient optimization techniques of Topaloglu [18].

5.2 Policy interpretation

We would like to compare sLR with $PHLP$. The former is a dynamic program, and the latter a heuristic that is based on generated sample paths. To make the comparison we reinterpret the sLR in terms of sample paths and policies.

For a fixed set of w^r's and λ^r's, sLR is a finite-period, finite-state dynamic program and hence the solution can be stated in terms of a policy—a specification of the optimal decision in each period at each state. We specify a policy π as a map from time and state (remaining capacity) to a subset of acceptable products from the set of feasible products.

$$\pi : (t, r_t) \rightarrow \mathcal{U}_{r_t}$$

Since the dynamic program of sLR decomposes by resource for a fixed set of w^r's and λ^r's, we define the policies at the resource-level as:

$$\pi_i : (t, r_{i,t}) \rightarrow \mathcal{U}_{i,r_{i,t}}$$

For a given set of w^r's, λ^r's and π_i, let $V_{i,0}^{w^r,\lambda^r}(\pi_i)$ be the expected revenue from applying policy π_i.

sLR can be rewritten in terms of policies as

$$sLR = \min_{w^r,\lambda^r} \sum_i \max_{\pi_i} V_{i,0}^{w,\lambda^r}(\pi_i) = \min_{\lambda^r} \max_{\pi_i, i=1,\ldots,m} \min_{w} \sum_i V_{i,0}^{w,\lambda^r}(\pi_i) \tag{15}$$

The value function $V_{i,0}^{w,\lambda^r}(\pi_i)$ can be interpreted in terms of sample paths as follows: Generate N sample paths and for the fixed w's, λ^r's and π_i's, observe the amount of revenue obtained on that sample path. The average of the observed revenue as $N \rightarrow \infty$ equals $V_{i,0}^{w,\lambda^r}(\pi_i)$.

We write it in this form and give this interpretation because we introduce policy constraints and compare sLR with $PHLP$ in the next section.

6 A joint framework for $PHLP$ and sLR

In this section we interpret $PHLP$ in the framework of LR and sLR, showing they are different relaxations of a dynamic program disaggregated over sample paths. Then we give a a stronger version of $PHLP$ called $sPHLP$ and show that it is a tighter bound than sLR.

6.1 Some properties of the Lagrange multipliers

First, we state Proposition 1 of [18] that shows how (7) breaks up into resource-level dynamic programs (modified to use unscaled multipliers).

Proposition 3 *[18] For a fixed λ, for resource i, define a resource-level dynamic program recursion as*

$$\vartheta_{i,t}^{\lambda}(r_{i,t}) = \max_{y_{i,j,t}\in\{0,1\}} \sum_{j\in\mathcal{A}_{i,r_{i,t}}} \lambda_{i,j,t} y_{i,j,t} + p_{j,t}\vartheta_{i,t+1}^{\lambda}(r_{i,t} - y_{i,j,t}).\} \tag{16}$$

Then,

$$V_t^{\lambda}(r_t) = \sum_{t'=t}^{\tau} \sum_j [f_j p_{j,t'} - \sum_{i\in j} \lambda_{i,j,t'}]^+ + \sum_i \vartheta_{i,t}^{\lambda}(r_{i,t}), \tag{17}$$

for all t, r_t.

We are interested in minimizing $V_0^{\lambda}(r_0)$ over λ. The following shows we can restrict ourselves to λ's that satisfy $\sum_{i\in j} \lambda_{i,j,t} = f_j p_{j,t}$.

Proposition 4 *The optimal Lagrangian multipliers can be assumed to satisfy $\sum_{i\in j} \lambda_{i,j,t} = f_j p_{j,t}, \ \forall j, t$.*

Proof
Suppose for product j and time $t' \geq t$ have $\sum_{i\in j} \lambda_{i,j,t'} > f_j p_{j,t'}$. Then for some $i \in j$, $\lambda_{i,j,t'} > 0$, and letting $\delta = \min\{\lambda_{i,j,t'}, \lambda_{i,j,t'} - f_j p_{j,t'}\}$ replace $\lambda_{i,j,t'}$ by $\lambda^{\delta}_{i,j,t'} = \lambda_{i,j,t'} - \delta$.

$\vartheta_{i,t}^{\lambda^{\delta}}(r_{i,t}) \leq \vartheta_{i,t}^{\lambda}(r_{i,t})$ as we are reducing the value of a product keeping all other product prices the same, and as the first part of the right hand side of (17) is unaffected by this change, we get

$$V_t^{\lambda^{\delta}}(r_t) \leq V_t^{\lambda}(r_t).$$

If after performing this step, $\sum_{i\in j} \lambda_{i,j,t'} > f_j p_{j,t'}$ still, we can repeat for another resource i of this product j in this period t' till we get equality.

Suppose for some product j and time t', $\sum_i \lambda_{i,j,t'} < f_j p_{j,t'}$ then for some $i \in j$, and $\delta = f_j p_{j,t'} - \sum_i \lambda_{i,j,t'}$ replace $\lambda_{i,j,t'}$ by $\lambda^{\delta}_{i,j,t'} = \lambda_{i,j,t'} + \delta$.

$$\vartheta_{i,t}^{\lambda^{\delta}}(r_{i,t}) \leq \vartheta_{i,t}^{\lambda}(r_{i,t}) + \delta,$$

as increasing the value of j in period t by δ keeping all other values same cannot increase the optimal value of resource i by more than δ.

$$[f_j p_{j,t} - \sum_i \lambda_{i,j,t}]^+$$

however decreases by exactly δ, so

$$V_t^{\lambda^{\delta}}(r_t) \leq V_t^{\lambda}(r_t).$$

Q.E.D

6.2 DP and $PHIP$

The RLP method originated as a more or less heuristic way to generate bid-prices, and the connection between the DP and the $PHLP$ and $PHIP$ bounds, as far as we are aware of, not explicitly formulated in the literature. In this section we show that $PHIP$ is a relaxation of DP in its way.

Instance k of the $PHIP$ bound, can be considered a multi-period network RM problem in its own right, The arrival probabilities in this case are $p^k_{j,t}$, equal to 1 if there is an arrival of j in period t in this instance k and 0 otherwise.

We can write ($PHIP$) in an aggregated form over the N generated instances, as

$$\begin{aligned} & \max && \frac{1}{N}\sum_{k=1}^{N}\sum_{j,t} f_j p^k_{j,t} y^k_{j,t} && (18) \\ (PHIP) \quad & \text{s.t.} && \sum_t \sum_{j \ni i} y^k_{j,t} \le r_{i,0} \ \ \forall i, \ \forall k \\ & && 0 \le y^k_{j,t} \le 1 \\ & && y^k_{j,t} \in \{0,1\}. \end{aligned}$$

For each generated sample path k, the ($PHIP^k$) can in fact be formulated as a dynamic program similar to (5) (that an integer program can be formulated as a dynamic program is nothing new, but here our formulation follows the format of (5)).

$$\begin{aligned} & V^k_t(r_t) = \max && \sum_j p^k_{j,t}\{f_j y^k_{i,j,t,r_t} + V^k_{t+1}(r_t - \sum_{i\in j} y^k_{i,j,t,r_{i,t}} e_i\} \\ (PHIP^k_{DP}) \quad & \text{s.t.} && y^k_{i,j,t,r_{i,t}} \le r_{i,t} \ \ \forall j, \forall i \in j && (19) \\ & && y^k_{i,j,t,r_t} - y^k_{i,j,t,r_{i,t}} = 0 \ \ \forall j, \forall i \in j \\ & && y^k_{i,j,t,r_{i,t}} \in \{0,1\}. \end{aligned}$$

We combine ($PHIP^k_{DP}$) for $k = 1, \ldots, N$ into a single dynamic program ($PHIP_{DP}$) that can be considered as solving N dynamic programs in parallel. Let $\mathbf{r}_t = [r^1_t, \ldots, r^N_t]$ represent the state vectors of all the N instances in a single vector (a vector of size $N \times m$).

$$\begin{aligned} & V_t(\mathbf{r}_t) = && \frac{1}{N}(\sum_{k=1}^{N}\{V^k_t(r^k_t) = \max \sum_j p^k_{j,t}[f_j y^k_{i,j,t,r^k_t} + V^k_{t+1}(r^k_t - \sum_{i\in j} y^k_{i,j,t,r^k_{i,t}} e_i)]\}) \\ (PHIP_{DP}) \quad & \text{s.t.} && y^k_{i,j,t,r_{i,t}} \le r_{i,t} \ \forall k, \ \forall j, \ \forall i \in j && (20) \\ & && y^k_{i,j,t,r^k_t} - y^k_{i,j,t,r^k_{i,t}} = 0 \ \forall j, \ \forall i \in j && (21) \\ & && y^k_{i,j,t,,r_{i,t}} \in \{0,1\}. \end{aligned}$$

with the state transition to $\mathbf{r}_{t+1}$ given by $r^k_{t+1} = r^k_t - \sum_{i\in j} y^k_{i,j,t,r^k_{i,t}} e_i$. Notice that the variables $y^k_{i,j,t,r_{i,t}}$ are defined for all possible $r_{i,t}$, but the constraints (21) are defined only for the state we are in at time t in instance k, namely, r^k_t.

Formulating $PHIP$ as ($PHIP_{DP}$) allows us to view $PHIP$ as a relaxation of (DP) and also brings out the connection to LR in Section 6.4.

To the formulation $(PHIP_{DP})$ add the following constraints (with new variables $y_{i,j,t,r_{i,t}}$) that we call policy constraints (akin to non-anticipative constraints in stochastic programming)

$$y^k_{i,j,t,r_{i,t}} = y_{i,j,t,r_{i,t}}, \ \ \forall k, j, t, r_{i,t}, i \in j. \tag{22}$$

The reason we can do this is because we have formulated $(PHIP^k)$ and $(PHIP^k_{DP})$ such that the instance-specific $p^k_{j,t}$ appears only in the objective function and not in the constraints (see discussion after the formulation of $PHLP^k$) (4).

Proposition 5 *$PHIP_{DP}$ with the constraints (22) has the same value as $V^*(r_0)$ from (5).*

Proof
Let y^*_{i,j,t,r_t} represent the optimal decision variables (namely, a policy) of the dynamic program (5). If we generate the sample paths and apply these controls, we get expected value of the dynamic program, which also coincides with the value of the $PHIP_{DP}$ with these variables and satisfies (22), and vice versa. *Q.E.D*

So $PHIP$ is a relaxation of the policy constraints. This observation can also be applied to a dynamic program at the resource level.

6.3 $PHLP$

$PHLP$ is a linear program but it too can be formulated as a dynamic program albeit on a continuous state space. The dynamic programming representation of $PHLP^k$ is

$$(PHLP^k_{DP}) \qquad \begin{aligned} V^k_t(r_t) = \max \quad & \sum_{j,t} p^k_{j,t}\{f_j y^k_{i,j,t,r_t} + V_{t+1}(r_t - \sum_{i\in j} y^k_{i,j,t,r_{i,t}} e_i\} \\ \text{s.t.} \quad & y^k_{i,j,t,r_{i,t}} \le r_{i,t} \ \ \forall j, \forall i \in j \\ & y^k_{i,j,t,r_t} - y^k_{i,j,t,r_{i,t}} = 0 \ \ \forall j, \ \forall i \in j \\ & 0 \le y^k_{i,j,t} \le 1. \end{aligned} \tag{23}$$

Using the same ideas as LR, we can relax the constraints $-y^k_{i,j,t} + y^k_{i,j,t} \le 0$. However, the variables $y^k_{i,j,t}$ are actually y^k_{i,j,t,r_t} so there are an infinite number of Lagrange multipliers (as $y^k_{i,j,t}$ are continuous the state r^k_t is continuous). Since LR uses a common set of state-independent Lagrange multipliers across all states, it is not clear that we get the same value as $PHLP^k$. However, since $PHLP^k_{DP}$ is a deterministic dynamic program, we expect that if we apply the optimal controls (or the optimal decomposition) the state evolution is deterministic, and we reach only one state vector at each time point, and the optimal Lagrange multipliers corresponding to the variables of these states will "work" for all states - i.e., prevent these other states from being ever reached. This is the intuition behind the next proposition, the proof of which, to avoid the technicalities of infinite linear programming, uses only (finite) linear programming duality.

Equation (16) was defined with 0-1 variables. Consider the relaxed version for instance k

$$\hat{\vartheta}^{\lambda,k}_{i,t}(r_{i,t}) = \max_{0\le y^k_{i,j,t,r_{i,t}}\le 1} \sum_{j\in\mathcal{A}_{i,r_{i,t}}} \lambda^k_{i,j,t} y^k_{i,j,t,r_{i,t}} + p^k_{j,t}\hat{\vartheta}^{\lambda,k}_{i,t+1}(r_{i,t} - y^k_{i,j,t,r_{i,t}}). \tag{24}$$

and the equivalent of (17) for instance k

$$\hat{V}_t^{\lambda,k}(r_0) = \sum_{t=0}^{\tau}\sum_j [f_j p_{j,t}^k - \sum_i \lambda_{i,j,t}^k]^+ + \sum_{i\in j} \hat{\vartheta}_{i,0}^{\lambda,k}(r_{i,0}), \tag{25}$$

Proposition 6 *There exist state-independent optimal Lagrange multipliers* $\lambda_{i,j,t}^k$ *for the constraints* $y_{i,j,t,r_t}^k - y_{i,j,t,r_{i,t}}^k = 0$ *of* $(PHLP_{DP}^k)$ *such that* $\sum_{i\in j}\lambda_{i,j,t}^k = f_j p_{j,t}^k$ *and* $PHLP_{DP}^k = \hat{V}_0^{\lambda,k}(r_0)$.

Proof

Consider the following (slightly different from DLP, but mimicking LR constraints), linear programming formulation of $PHLP^k$:

$$\begin{aligned}
& \max && \sum_{j,t} f_j p_{j,t}^k y_{i,j,t}^k && (26)\\
(PHLP2^k) \quad & \text{s.t.} && \sum_t \sum_{j\ni i} y_{i,j,t}^k \le r_{i,0} \ \ \forall i && (27)\\
& && y_{i,j,t}^k - y_{i,j,t}^k = 0 \ \ \forall j, \forall i \in j && (28)\\
& && y_{i,j,t}^k \le 1 && (29)\\
& && y_{i,j,t}^k \ge 0 &&
\end{aligned}$$

Let $y_{i,j,t}^{*,k}$ be the optimal primal variables and $\mu_i^{*,k}, \lambda_{i,j,t}^{*,k}, \gamma_{j,t}^{*,k}$ be the optimal dual variables corresponding to (27), (28) and (29) respectively. $\lambda_{i,j,t}^{*,k} \ge 0$, and $\sum_{i\in j}\lambda_{i,j,t}^{*,k} = f_j p_{j,t}^k$ as $y_{\imath,j,t}$ is unrestricted. Relax the constraints (28) using $\lambda_{i,j,t}^{*,k}$:

$$\begin{aligned}
& \max && \sum_i \sum_t \sum_{j\ni i} \lambda_{i,j,t}^{*,k} y_{i,j,t}^k && (30)\\
(PHLP2^{\lambda^*,k}) \quad & \text{s.t.} && \sum_t \sum_{j\ni i} y_{i,j,t}^k \le r_{i,0} \ \ \forall i && \\
& && y_{i,j,t}^k \le 1 && (31)\\
& && y_{i,j,t}^k \ge 0 &&
\end{aligned}$$

From linear programming duality theory, it can easily be verified that

$$PHLP2^k = PHLP2^{\lambda^*,k}$$

as $y_{i,j,t}^{*,k}$ and and $\mu_i^{*,k}, \gamma_{j,t}^{*,k}$ can serve as the optimal primal and dual variables of $PHLP2^{\lambda^*,k}$.

$PHLP2^{\lambda^*,k}$ is composed of resource-level linear programs, each of which has the same value as $\hat{\vartheta}_{i,0}^{\lambda^*,k}$ as can be seen by simple substitution of the $y_{i,j,t}^*$ variables. *Q.E.D*

Define the integer program version of $(PHLP2^{\lambda^*,k})$:

$$\begin{aligned}
& \max && \sum_i \sum_t \sum_{j\ni i} \lambda_{i,j,t}^{*,k} p_{j,t}^k y_{i,j,t}^k && (32)\\
(PHIP2^{\lambda^*,k}) \quad & \text{s.t.} && \sum_t \sum_{j\ni i} y_{i,j,t}^k \le r_{i,0} \ \ \forall i && \\
& && y_{i,j,t}^k \in \{0,1\}. &&
\end{aligned}$$

Let $(V_0^{\lambda^*,k})$ be $(V_0^{\lambda^*})$ of (17) applied to instance k (i.e., with $p^k_{j,t}$). Compare $(PHIP2^{\lambda^*,k})$ with $(V_0^{\lambda^*,k})$: as $\sum_{i\in j} \lambda^{*,k}_{i,j,t} = f_j p^k_{j,t}$, $\forall j$, and ignoring all j,t with $p^k_{j,t} = 0$, we see that they are the same.

The Lagrangian relaxation of a (maximization) integer program is a tighter upper bound than the linear programming relaxation, and an equivalent result for the dynamic program version of the integer program $PHIP^k_{DP}$, by adding the 0-1 variable restriction *after* relaxing (28) is as follows:

Proposition 7 $PHLP^k = \hat{V}_0^{\lambda^*,k}(r_0) \geq V_0^{\lambda^*,k}(r_0) \geq PHIP^k_{DP}$.

Define the perfect hindsight Lagrangian relaxation as $PHS_{LR} = \frac{1}{N}\sum_k V_0^{\lambda^*,k}(r_0)$. PHS_{LR} is the average of resource-level dynamic programs with product costs given by $\lambda^{*,k}_{i,j,t}$. By Proposition 7, $PHLP \geq PHS_{LR}$.

6.4 A joint view of $PHIP$ and sLR

In this section we show that both $PHIP$, LR and sLR are in fact different relaxations of the same dynamic program. This allows us to compare them in the next section.

In Section 6.2 we showed that $(PHIP_{DP})$ with additional constraints approaches (DP) for sufficiently large N. For N generated sample paths, we write (DP^N) for all possible state vectors $\mathbf{r}_t$ as follows:

$$
\begin{aligned}
& V_t(\mathbf{r}_t)^N = && \frac{1}{N}\Big(\sum_{k=1}^{N}\{V^k_t(r^k_t) = \max\sum_j p^k_{j,t}[f_j y^k_{\imath,j,t,r^k_t} + V^k_{t+1}(r^k_t - \sum_{i\in j} y^k_{i,j,t,r^k_{i,t}} e_i)]\}\Big) \\
(DP^N) \quad & \text{s.t.} && y^k_{i,j,t,r_{i,t}} \leq r_{i,t} \quad \forall k, \forall j, \forall i \in j && (33)\\
& && y^k_{i,j,t,r_{i,t}} - y_{i,j,t,r_{i,t}} = 0 \quad \forall k, \forall j, \forall i \in j && (34)\\
& && y^k_{\imath,j,t,r^k_t} - y^k_{i,j,t,r^k_{i,t}} = 0 \quad \forall k, \forall j, \forall i \in j && (35)\\
& && y^k_{\imath,j,t,r^k_t} - y_{\imath,j,t,r^k_t} = 0 \quad \forall k, \forall j && (36)\\
& && y^k_{\imath,j,t,r^k_t} - y_{\bar{\imath},j,t,r^k_t} = 0 \quad \forall k, \forall j && (37)\\
& && y_{\bar{\imath},j,t,r^k_t}, y_{\imath,j,t,r_t}, y^k_{i,j,t,r^k_{i,t}}, y_{i,j,t,r_{i,t}} \in \{0,1\}
\end{aligned}
$$

with state transitions as defined in $(PHIP_{DP})$. $y_{\bar{\imath},j,t,r_t}$ is a set of new dummy variables. Equations (37) are redundant, but will serve to interpret sLR. As in $(PHIP_{DP})$, constraints (33) and (34) are defined for all states r_t whereas constraints (35), (36) and (37) are defined only for the state r^k_t we are in in instance k at time t.

The constraints of (DP^N) implement the dynamic programming policies discussed in Section 5.2; namely, if for two instances we have the same capacity vector, we should be making the same decision. The variables $y_{i,j,t,r_{i,t}}$ and $y_{\imath,j,t,r_t}$ implement this common decision for state r_t.

The value of (DP^N) converges to the value of (DP) by considering the state vector $\mathbf{r}_t = [r_t,\ldots,r_t]$ by invoking a sample-path argument: fixing the variables $y_{i,j,t,r_{i,t}}$ and $y_{\imath,j,t,r_t}$ common to both, as $N \to \infty$ the value of (DP^N) converges to that of (DP).

Relaxations of the constraints of (DP^N) lead to $PHIP$ and LR: Relaxing equations (34), equations (36) and equations(37) with 0 (that is removing them) leads to $PHIP$. Relaxing equations

(35) with $\lambda_{i,j,t}$ and substituting $y_{i,j,t,r_{i,t}}$ and $y_{\imath,j,t,r_t}$ for $y^k_{i,j,t,r_{i,t}}$ and $y^k_{\imath,j,t,r_t}$ respectively leads to LR.

6.5 (sLR) from (DP^N)

We can derive sLR from (DP^N) as follows. Relax equations (35) with $\lambda_{i,j,t,r_{i,t}}$ and equations (37) with $w_{i,j,t,r_{i,t}}$ whenever $r^k_{i,t} = r_{i,t}$; with the multipliers satisfying $\sum_{i\in j} w_{i,j,t,r_{i,t}} \geq 0$ and $\sum_{i\in j}(w_{i,j,t,r_{i,t}} + \lambda_{i,j,t,r_{i,t}}) \geq f_j p_{j,t}$ for all states r_t. In the objective function, the coefficient of $y_{\bar{\imath},j,t,r^k_t}$ is $\sum_{i\in j} w_{i,j,t,r^k_{i,t}}$, the coefficient of $y_{i,j,t,r^k_{i,t}}$ is $\lambda_{i,j,t,r^k_{i,t}}$ and the coefficient of $y^k_{\imath,j,t,r^k_t}$ is $[p^k_{j,t} f_j - \sum_{i\in j}(\lambda_{i,j,t,r^k_{i,t}} + w_{i,j,t,r^k_{i,t}})]$. After relaxing equations (35) with $\lambda_{i,j,t,r_{i,t}}$ and equations (37) with $w_{i,j,t,r_{i,t}}$, (DP^N) becomes the following:

$$
\begin{aligned}
V_t^{N,w^r,\lambda^r}(\mathbf{r}_t) = \quad & \frac{1}{N}\Big(\sum_{k=1}^{N} V_t^{k,w^r,\lambda^r}(r^k_t) = \max\{\sum_j (p^k_{j,t} f_j - \sum_{i\in j}(\lambda_{i,j,t,r^k_{i,t}} + w_{i,j,t,r^k_{i,t}}))y^k_{\imath,j,t,r^k_t} \\
& + \sum_{i\in j} \lambda_{i,j,t,r^k_{i,t}} y^k_{i,j,t,r^k_{i,t}} + (\sum_{i\in j} w_{i,j,t,r^k_{i,t}}) y_{\bar{\imath},j,t,r^k_t} + p^k_{j,t} V^k_{t+1}(r^k_t - \sum_{i\in j} y^k_{i,j,t,r^k_{i,t}} e_i)\}\Big) \\
(DP_1^{N,w^r,\lambda^r}) \text{ s.t.} \quad & y^k_{i,j,t,r_{i,t}} \leq r_{i,t} \quad \forall k,\ \forall j, \forall i \in j \qquad (38) \\
& y^k_{i,j,t,r_{i,t}} - y_{i,j,t,r_{i,t}} = 0 \quad \forall k,\ \forall j, \forall i \in j \qquad (39) \\
& y^k_{\imath,j,t,r^k_t} - y_{\imath,j,t,r^k_t} = 0 \quad \forall k,\ \forall j \qquad (40) \\
& y_{\imath,j,t,r_t}, y_{\bar{\imath},j,t,r_t}, y^k_{\imath,j,t,r^k_t}, y^k_{i,j,t,r^k_{i,t}}, y_{i,j,t,r_{i,t}} \in \{0,1\}
\end{aligned}
$$

We can substitute $y_{i,j,t,r_{i,t}}$ for $y^k_{i,j,t,r^k_{i,t}}$ using (39) whenever $r_{i,t} = r^k_{i,t}$, and $y_{\imath,j,t,r_t}$ for $y^k_{\imath,j,t,r^k_t}$ by (40) whenever $r_t = r^k_t$. As $\sum_{i\in j} w_{i,j,t,r_{i,t}} \geq 0$, we can replace $(\sum_{i\in j} w_{i,j,t,r_{i,t}}) y_{\bar{\imath},j,t,r_t}$ by $(\sum_{i\in j} w_{i,j,t,r_{i,t}})$ (as $y_{\bar{\imath},j,t,r_t}$ does not appear in the constraints once we relax (37)). Replacing $y^k_{i,j,t,r_{i,t}}$ by $y_{i,j,t,r_{i,t}}$ and $y^k_{\imath,j,t,r^k_t}$ by $y_{\imath,j,t,r_t}$ leads to

$$
\begin{aligned}
V_t^{N,w^r,\lambda^r}(\mathbf{r}_t) = \quad & \frac{1}{N}\Big(\sum_{r_t}\sum_{k\in\mathcal{S}(r_t)} V_t^{k,w^r,\lambda^r}(r_t) = \max\{\sum_j (p^k_{j,t} f_j - \sum_{i\in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}}))y_{\imath,j,t,r_t} \\
& + \sum_{i\in j} \lambda_{i,j,t,r_{i,t}} y_{i,j,t,r_{i,t}} + (\sum_{i\in j} w_{i,j,t,r_{i,t}}) y_{\bar{\imath},j,t,r_t} + p^k_{j,t} V^k_{t+1}(r_t - \sum_{i\in j} y_{i,j,t,r_{i,t}} e_i)\}\Big) \\
(DP^{N,w^r,\lambda^r}) \text{ s.t.} \quad & y_{i,j,t,r_{i,t}} \leq r_{i,t} \quad \forall k,\ \forall j, \forall i \in j \qquad (41) \\
& y_{\imath,j,t,r_t}, y_{\bar{\imath},j,t,r_t}, y_{i,j,t,r_{i,t}} \in \{0,1\}
\end{aligned}
$$

Proposition 8 *For every $\epsilon > 0$, there is a N_ϵ such that for all $N > N_\epsilon$ the value of $|sLR - DP^{N,w^r,\lambda^r}| < \epsilon$, where w^r, λ^r are the optimal multipliers of sLR.*

Proof
From (14), the sLR subproblem for resource i is the following

$$
\begin{aligned}
V_{i,t}^{w^r,\lambda^r}(r_{i,t}) = \max \textstyle\sum_{j\ni i}\{ & w_{i,j,t,r_{i,t}} + \lambda_{i,j,t,r_{i,t}} y_{i,j,t,r_{i,t}} + \\
& p_{j,t} V_{i,t+1}^{w^r,\lambda^r}(r_{i,t} - y_{i,j,t,r_{i,t}})\}
\end{aligned}
$$

s.t.

$$y_{i,j,t,r_{i,t}} \leq r_{i,t} \quad \forall j \ni i \tag{42}$$
$$y_{i,j,t,r_{i,t}} \in \{0,1\}.$$

Assume as in *sLR* that $\lambda_{i,j,t,r_{i,t}}$ and $w_{i,j,t,r_{i,t}}$ satisfy, for all r_t

$$p_{j,t} f_j - \sum_{i \in j} (\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}}) \leq 0 \tag{43}$$

and

$$\sum_{i \in j} w_{i,j,t,r_{i,t}} \geq 0. \tag{44}$$

For any fixed solution $y_{i,j,t,r_{i,t}}$ (i.e., fixing a policy), define $\mathcal{S}(r_t) = \{k | r_t^k = r_t\}$; i.e., the subset of the instances $1, 2, \ldots, N$ where the state is r_t at time t. If no instance is in state r_t, then $\mathcal{S}(r_t)$ is defined as $\emptyset$. Likewise, define $\mathcal{S}(r_{i,t}) = \{k | r_{i,t}^k = r_{i,t}\}$; i.e., the subset of the instances $1, 2, \ldots, N$ where the state on resource i at time t is $r_{i,t}$. Let $n_{r_t} = |\mathcal{S}(r_t)|$ and $n_{r_{i,t}} = |\mathcal{S}(r_{i,t})|$. As under any policy we are at most in exactly one state per instance, we have $\sum_{r_t} n_{r_t} = N$.

Consider the co-efficient of $y_{\imath,j,t,r_t}$ in period t under a fixed policy

$$\frac{1}{N} \sum_{k \in \mathcal{S}(r_t)} (p_{j,t}^k f_j - \sum_{i \in j} (\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})) y_{\imath,j,t,r_t} = \frac{n_{r_t}}{N} (\frac{\sum_{k \in \mathcal{S}(r_t)} p_{j,t}^k}{n_{r_t}} f_j - \sum_{i \in j} (\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})) y_{\imath,j,t,r_t} \tag{45}$$

$y_{\imath,j,t,r_t} = 0$ if $\{\frac{\sum_{k \in \mathcal{S}(r_t)} p_{j,t}^k}{n_{r_t}} f_j - \sum_{i \in j} (\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})\} \leq 0$ and $y_{\imath,j,t,r_t} = 1$ otherwise. Likewise, the optimal solution of (DP^{N,w^r,λ^r}) will always have $y_{\bar{\imath},j,t,r_t} = 1$ because of our assumption (44).

First, as a sub-optimal solution to (DP^{N,w^r,λ^r}), set $y_{\imath,j,t,r_t} = 0$ irrespective of the value of its co-efficient. Once we substitute these values, the dynamic program $((DP^{N,w^r,\lambda^r}))$ decomposes into resource-level dynamic programs, as there are no variables or constraints in common between the different resources. Let $\mathbf{r}_{i,t}$ be the N-dimensional vector of state on resource i in instance k. The decomposed problem on resource i is

$$V_{i,t}^{N,w^r,\lambda^r}(\mathbf{r}_{i,t}) = \frac{1}{N} (\sum_{r_{i,t}} \sum_{k \in \mathcal{S}(r_{i,t})} V_{i,t}^{k,w^r,\lambda^r}(r_{i,t}^k) = \max\{\sum_{j \ni i} \lambda_{i,j,t,r_{i,t}} y_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}} \tag{46}$$
$$+ p_{j,t}^k V_{i,t+1}^k (r_{i,t} - y_{i,j,t,r_{i,t}})\})$$
$$(DP_i^{N,w^r,\lambda^r}) \quad \text{s.t.} \quad y_{i,j,t,r_{i,t}} \leq r_{i,t} \quad \forall j \ni i, \tag{47}$$
$$y_{i,j,t,r_{i,t}} \in \{0,1\}$$

and $\sum_{i=1}^m V_{i,t}^{N,w^r,\lambda^r}(\mathbf{r}_{i,t})$ is the value of the solution (DP^{N,w^r,λ^r}) with $y_{\imath,j,t,r_t}$ set to 0.

For N large, $V_{i,t}^{N,w^r,\lambda^r}(\mathbf{r}_{i,t})$ converges to the value of *sLR* on resource i: Fix the optimal policy of the dynamic program (42). The value of the dynamic program is approximated arbitrarily closely by generating sample paths and applying the policy on each sample path, and taking the average of the value for each sample path; in short, the solution applied to (DP_i^{N,w^r,λ^r}), and vice versa.

One final detail is the effect of setting $y_{\imath,j,t,r_t} = 0$ in (DP^{N,w^r,λ^r}). We argue that for N large, the effect of this is negligible, that is bounded by ϵ.

Call a state r_t *reachable* under a fixed policy if $r_t^k = r_t$ for some instance k applying the policy. If a state is reachable it is reachable an infinite number of times as $N \to \infty$ as we have a finite number of periods and random variables with discrete distributions (Bernoulli for each product) in each period. So, for all reachable states r_t under a fixed policy as $N \to \infty$, $n_{r_t} \to \infty$, and we expect $\frac{\sum_{k \in \mathcal{S}(r_t)} p^k_{j,t}}{n_{r_t}} \to p_{j,t}$.

Consider the optimal solution (in other words, policy) for a fixed N for the dynamic program (DP^{N,w^r,λ^r}). Since we have only a finitely many states and policies, as $N \to \infty$ we have some policy repeating as the optimal policy, and the value of this optimal policy is within (say) $\frac{\epsilon}{2}$ of the value that (DP^{N,w^r,λ^r}) converges to as $N \to \infty$. Fix such a policy. Assume N is large enough (and consequently, n_{r_t}) such that

$$|\frac{\sum_{k \in \mathcal{S}(r_t)} p^k_{j,t}}{n_{r_t}} - p_{j,t}| \le \frac{\epsilon}{2n\tau\bar{f}} \quad \forall t, j$$

for all reachable states r_t under our fixed policy, where $\bar{f} = \max_j f_j$. The difference between an optimal solution after we fix $y_{i,j,t,r_t} = 0$ and the optimal solution to (DP^{N,w^r,λ^r}) under the optimal policy, from Equation (45):

$$\begin{aligned}
\frac{1}{N}\sum_t\sum_j\sum_{r_t}[\sum_{k\in S(r_t)} p^k_{j,t}f_j &- \sum_{i\in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})]^+ \\
&= \frac{1}{N}\sum_t\sum_j\sum_{r_t}[\sum_{k\in S(r_t)} p^k_{j,t}f_j - f_j p_{j,t} + f_j p_{j,t} - \sum_{i\in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})]^+ \\
&\le \frac{1}{N}\sum_t\sum_j\sum_{r_t}[\sum_{k\in S(r_t)} p^k_{j,t}f_j - f_j p_{j,t}]^+ + [f_j p_{j,t} - \sum_{i\in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})]^+ \\
&= \frac{1}{N}\sum_t\sum_j\sum_{r_t}[\sum_{k\in S(r_t)} p^k_{j,t}f_j - f_j p_{j,t}]^+ \\
&= \sum_t\sum_j\sum_{r_t}\frac{n_{r_t}}{N} f_j[\frac{\sum_{k\in S(r_t)} p^k_{j,t}}{n_{r_t}} - p_{j,t}]^+ \\
&\le \sum_t\sum_j\sum_{r_t}\frac{n_{r_t}}{N} f_j|\frac{\sum_{k\in S(r_t)} p^k_{j,t}}{n_{r_t}} - p_{j,t}| \\
&\le \sum_t\sum_j\sum_{r_t}\frac{n_{r_t}}{N}\frac{\epsilon}{2n\tau} \\
&\le \sum_t\sum_j\frac{\epsilon}{2n\tau} \\
&= \frac{\epsilon}{2}
\end{aligned}$$

Q.E.D

6.6 *sPHLP*

The interpretation of sLR as a relaxation of $((DP^N))$ also shows a way to improve the upper bound sLR. What, if instead of a common $\lambda_{i,j,t,r_{i,t}}$ and $w_{i,j,t,r_{i,t}}$ for all instances as in sLR, we relax the constraints using instance-specific multipliers as in $PHIP$: using $\lambda^k_{i,j,t,r^k_{i,t}}$, 0, and $w^k_{i,j,t,r^k_{i,t}}$ to relax constraints (35),(36) and (37) respectively, with the restrictions that $\sum_{i \in j} w^k_{i,j,t,r_{i,t}} \geq 0$ and $[p^k_{j,t} f_j - \sum_{i \in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})] \leq 0$ for all states r_t. We call this relaxation $sPHIP$ and if we ignore the integer constraints, $sPHLP$.

We claim that $sPHIP$ is a tighter relaxation than sLR. Given the optimal $\lambda_{i,j,t,r_{i,t}}$ and $w_{i,j,t,r_{i,t}}$ from sLR, we define (assume $p_{j,t} > 0$)

$$w^k_{i,j,t,r_{i,t}} = \frac{p^k_{j,t} w_{i,j,t,r_{i,t}}}{p_{j,t}} \tag{48}$$

and

$$\lambda^k_{i,j,t,r_{i,t}} = \frac{p^k_{j,t} \lambda_{i,j,t,r_{i,t}}}{p_{j,t}} \tag{49}$$

Then, for all states, $\sum_{i \in j} w^k_{i,j,t,r_{i,t}} \geq 0$ and $[p^k_{j,t} f_j - (\sum_{i \in j}(\lambda^k_{i,j,t,r_{i,t}} + w^k_{i,j,t,r_{i,t}})] \leq 0$, as $[p_{j,t} f_j - (\sum_{i \in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}}))] \leq 0$ for the multipliers in sLR.

Relax the constraints (35),(36) and (37) with $\lambda^k_{i,j,t,r^k_{i,t}}$, 0, and $w^k_{i,j,t,r^k_{i,t}}$ to obtain:

$$\begin{aligned} V_t(\mathbf{r}_t)^N = \frac{1}{N}(\sum_{k=1}^{N}\{V^k_t(r^k_t) \quad &= \quad \max \sum_j [(p^k_{j,t} f_j - (\lambda^k_{i,j,t,r^k_{i,t}} + w^k_{i,j,t,r^k_{i,t}}))y^k_{\imath,j,t,r^k_t} + (\sum_{i \in j} w^k_{i,j,t,r^k_{i,t}}) y_{\bar{\imath},j,t,r^k_t} \\ & \quad + \sum_{i \in j} \lambda^k_{i,j,t,r^k_{i,t}} y^k_{i,j,t,r^k_{i,t}} + p^k_{j,t} V^k_{t+1}(r^k_t - \sum_{i \in j} y^k_{i,j,t,r^k_{i,t}} e_i)]\}) \\ (PHIP_1^{N,w^r,\lambda^r}) \quad \text{s.t.} \quad & y^k_{i,j,t,r_{i,t}} \leq r_{i,t} \quad k = 1, \ldots, N, \ \forall j, \forall i \in j && (50) \\ & y^k_{i,j,t,r^k_{i,t}} - y_{i,j,t,r^k_{i,t}} = 0 \quad \forall k, \ \forall j, \forall i \in j && (51) \\ & y_{\bar{\imath},j,t,r^k_t}, y_{\imath,j,t,r_t}, y^k_{i,j,t,r^k_{i,t}}, y_{i,j,t,r_{i,t}} \in \{0,1\} \end{aligned}$$

for which we can set $y_{\bar{\imath},j,t,r^k_t} = 1$ and $y^k_{\imath,j,t,r^k_t} = 0$ to obtain

$$\begin{aligned} V_t(\mathbf{r}_t)^N = \frac{1}{N}(\sum_{k=1}^{N}\{V^k_t(r^k_t) \quad &= \quad \max \sum_j [\sum_{i \in j} w^k_{i,j,t,r^k_{i,t}} + \sum_{i \in j} \lambda^k_{i,j,t,r^k_{i,t}} y^k_{i,j,t,r^k_{i,t}} \\ & \quad + p^k_{j,t} V^k_{t+1}(r^k_t - \sum_{i \in j} y^k_{i,j,t,r^k_{i,t}} e_i)]\}) \\ (PHIP_2^{N,w^r,\lambda^r}) \quad \text{s.t.} \quad & y^k_{i,j,t,r^k_{i,t}} \leq r^k_{i,t} \quad \forall k, \ \forall j, \forall i \in j && (52) \\ & y^k_{i,j,t,r^k_{i,t}} - y_{i,j,t,r^k_{i,t}} = 0 \quad \forall k, \ \forall j, \forall i \in j && (53) \\ & y^k_{i,j,t,r^k_{i,t}}, y_{i,j,t,r_{i,t}} \in \{0,1\} \end{aligned}$$

Substituting (48) and (49) into $(PHIP_2^{N,w^r,\lambda^r})$ and replacing $y^k_{i,j,t,r^k_{i,t}}$ by $y_{i,j,t,r_{i,t}}$ we get resource-level dynamic programs. For resource i, this is as follows:

$$\begin{aligned} V_{i,t}(\mathbf{r}_{i,t})^N = \quad & \frac{1}{N}(\sum_{k=1}^{N}\{V^k_t(r^k_{i,t}) = \max \sum_j [\frac{p^k_{j,t} w_{i,j,t,r_{i,t}}}{p_{j,t}} + \frac{p^k_{j,t} \lambda_{i,j,t,r_{i,t}}}{p_{j,t}} y_{i,j,t,r^k_{i,t}} \\ & \quad + p^k_{j,t} V^k_{t+1}(r^k_{i,t} - y_{i,j,t,r^k_{i,t}})]\}) \\ (PHIP_3^{N,w^r,\lambda^r}) \qquad \text{s.t.} \qquad & y_{i,j,t,r_{i,t}} \leq r_{i,t} \quad \forall k, \forall i \in j && (54) \\ & y_{i,j,t,r_{i,t}} \in \{0,1\} \end{aligned}$$

We claim

Proposition 9 *$\epsilon > 0$, there is a sufficiently large N_ϵ such that for all $N > N_\epsilon$ the value of $(PHIP_3^{N,w^r,\lambda^r})$ is within ϵ of the value of sLR.*

Proof
(Sketch) For each resource i, consider the optimal solution corresponding to the sub-problem for resource i in sLR. One can approximate the value for this subproblem by fixing the policy, generating sample-paths and taking the average over all the revenues obtained over the sample paths. This is essentially the revenue obtained by $(PHIP_3^{N,w^r,\lambda^r})$ for the same policy for large enough N. Likewise, one can fix the optimal policy (it is a policy as we use a common $y_{i,j,t,r_{i,t}}$ over all instances) for $(PHIP_3^{N,w^r,\lambda^r})$ and show that it approaches the revenue of sLR for each resource i. *Q.E.D*

In $(PHIP_3^{N,w^r,\lambda^r})$, the multipliers corresponding to a j,t and k are 0 whenever $p_{j,t}^k = 0$. One can assume this to be true in general. We claim

Proposition 10 *For $sPHIP$, whenever $p_{j,t}^k = 0$, we can assume the optimal multipliers satisfy $\lambda_{i,j,t,r_{i,t}}^k = w_{i,j,t,r_{i,t}}^k = 0$.*

Proof
As we require the multipliers to satisfy $\sum_{i\in j} w_{i,j,t,r_{i,t}}^k \geq 0$ and $[p_{j,t}^k f_j - \sum_{i\in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})] \leq 0$, whenever $p_{j,t}^k = 0$, $\sum_{i\in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})] \geq 0$. The multipliers affect only the value in period t where they appear in the objective function as $\sum_{i\in j} w_{i,j,t,r_{i,t}}^k + \lambda_{i,j,t,r_{i,t}}^k y_{i,j,t,r_{i,t}}$. When $p_{j,t}^k = 0$ both $\sum_{i\in j} w_{i,j,t,r_{i,t}}^k \geq 0$ and $\sum_{i\in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})] \geq 0$, the optimal multipliers would minimize the objective function for any fixed $y_{i,j,t,r_{i,t}} \in \{0,1\}$ by setting $\lambda_{i,j,t,r_{i,t}}^k = w_{i,j,t,r_{i,t}}^k = 0$. *Q.E.D*

Proposition (10) implies that the number of multipliers are a lot less than the maximum possible—two multipliers for every generated arrival in the simulation, rather than $2mnN\tau\bar{r}_0$).

While solving $sPHIP$ would be very difficult, $sPHLP$ can be formulated as a linear program with an exponential number of constraints, but with number of variables relatively small and linear in N; for small values of N (say 20 or 30) it is solvable even for large problems. The main advantage of $sPHLP$ is that even though there are exponential number of constraints, the separation problem can be solved trivially, so we can generate violated constraints on the fly very easily.

6.7 Solving $sPHLP$

For a fixed $\lambda_{i,j,t,r_{i,t}}^k$ and $w_{i,j,t,r_{i,t}}^k$ we formulate the maximization problem as a linear program—in fact, a set of network-flow problems, one each for resource i and instance k, linked by constraints (34). We then take its dual and combine it with the minimization over $\lambda_{i,j,t,r_{i,t}}^k$ and $w_{i,j,t,r_{i,t}}^k$.

The problem has an exponential set of constraints as the multipliers have to satisfy $[p_{j,t}^k f_j - \sum_{i\in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})] \leq 0$ and $\sum_{i\in j} w_{i,j,t,r_{i,t}}^k \geq 0$ for all states r_t. We generate violated constraints on the fly. For a given set of $\lambda_{i,j,t,r_{i,t}}^k$ and $w_{i,j,t,r_{i,t}}^k$'s, finding a violated constraint is trivial which makes the method appealing (say compared to the AR bound)

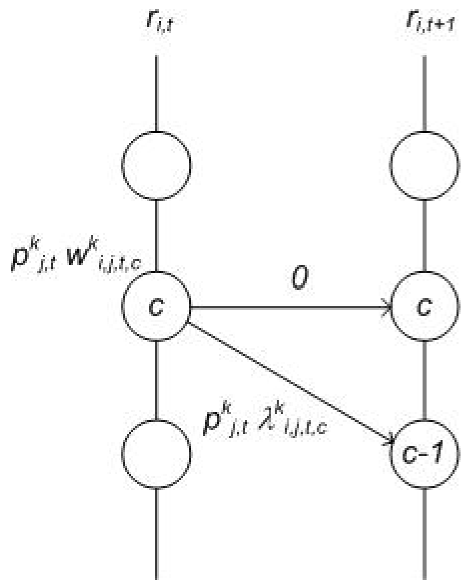

Figure 1: Network for resource i and instance k

First, for a fixed $\lambda^k_{i,j,t,r_{i,t}}$ and $w^k_{i,j,t,r_{i,t}}$, consider the dynamic program without constraints (34) linking the different instances for resource i and instance k:

$$\begin{aligned} V^k_t(r_{i,t}) \quad &= \quad \max \sum_j [w^k_{i,j,t,r_{i,t}} + \lambda^k_{i,j,t,r_{i,t}} y_{i,j,t,r_{i,t}} + p^k_{j,t} V^k_{t+1}(r^k_{i,t} - y_{i,j,t,r_{i,t}})]) \qquad (55) \\ (PHIP^{k,,w^r,\lambda^r}) \quad \text{s.t.} \quad & y_{i,j,t,r_{i,t}} \le r_{i,t} \quad \forall k,\ \forall j \ni i \\ & y_{i,j,t,r_{i,t}} \in \{0,1\} \end{aligned}$$

This can be viewed as network-flow problem on an acyclic network with nodes representing the states $r_{i,t}$, a node-weight of $\sum_{j \ni i} w^k_{i,j,t,r_{i,t}}$, and arcs from $r_{i,t}$ to $r_{i,t+1} = r_{i,t} - 1$ with revenues $\lambda^k_{i,j,t,r_{i,t}}$ for each $j \ni i$ with $p^k_{j,t} = 1$, and an arc from $r_{i,t}$ to $r_{i,t+1} = r_{i,t}$ with a revenue of 0 (see Figure 1). We want to find a maximum-value flow from the starting state to some terminal state. This is nothing more than the standard network representation of a *deterministic* finite-state, finite-period dynamic program.

There is a network for each resource i and instance k. The linear program formulation of this flow problem has variables $y^k_{i,j,t,r_{i,t}}$ and the constraints (34) are nothing more than setting $y^k_{i,j,t,r_{i,t}} = y^l_{i,j,t,r_{i,t}} = y_{i,j,t,r_{i,t}}$ for two instances k and l where $p^k_{j,t} = p^l_{j,t} = 1$, as by Proposition 10 whenever $p^k_{j,t} = 0$, we can assume $\lambda^k_{i,j,t,r^k_{i,t}} = w^k_{i,j,t,r^k_{i,t}} = 0$.

So for a fixed $\lambda^k_{i,j,t,r_{i,t}}$, $w^k_{i,j,t,r_{i,t}}$ we have a maximization linear programming problem. We want to minimize this over all $\lambda^k_{i,j,t,r_{i,t}}$, $w^k_{i,j,t,r_{i,t}}$ subject to the constraints $[p^k_{j,t} f_j - \sum_{i \in j}(\lambda_{i,j,t,r_{i,t}} + w_{i,j,t,r_{i,t}})] \le 0$ and $\sum_{i \in j} w^k_{i,j,t,r_{i,t}} \ge 0$. By taking the dual of the inner maximization problem, we obtain a single minimization problem.

So the solution for $sPHLP$ exploits the fact that for a deterministic dynamic program there is a compact linear programming formulation.

7 LR vs. AR bound

Computational results of [18] seem to indicate that the AR bound is weaker than the LR bound, but we don't know of any theoretical result that proves it is the case for all instances. It is quite possible of course that the two bounds are not comparable and there are instances where one dominates the other.

In this section we propose a minor variation of the AR bound that, at the expense of expanding the number of variables, leads to a formulation and a bound provably tighter than the LR bound.

7.1 The sAR bound

Our variation of the AR bound consists of the following relaxation of the dynamic programming value functions:

$$V_t(r_t) \approx \theta_t + \sum_i \sum_{r=1}^{r_{i,t}} v_{i,t,r}, \quad \forall t, r_t. \tag{56}$$

A few words on this variant are in order. The number of variables have increased. If R is the maximum capacity on the resources, then they have increased by R from the AR relaxation as now we specify a variable for all possible capacity points for each t and i. One can then think of these as capacity-dependent marginal values or bid-prices. For lack of a better alternative we call this sAR for strong AR bound or strengthened AR bound[2].

Since we are imposing a functional form on the value functions, this is clearly an upper bound on the dynamic program. Since the AR bound is a special case with the additional restriction $v_{i,t,r} = v_{i,t,s}$ for r, s, if one substitutes (56) into (1) and solve the resulting linear program, the objective value is less than or equal to the AR bound value (as it is a minimization problem).

So we have:

Proposition 11 *$AR \geq$sAR$\geq V^*(r_0)$.*

The interesting thing is that sAR can be shown to be stronger than the LR bound.

7.2 sAR vs. LR bound

Consider the linear program obtained after substituting the sAR approximation (56) for the optimal value functions:

$$\begin{aligned} & \min_{\theta, v_{i,t,r}} && \theta_0 + \sum_i \sum_{r=1}^{r_{i,0}} v_{i,0,r} \\ (sAR) \quad & \text{s.t.} && \theta_t - \theta_{t+1} + \sum_i \{ \sum_{r=1}^{r_{i,t}} v_{i,t,r} \\ &&& - \sum_j p_{j,t} \sum_{r=1}^{r_{i,t}-u_{i,j,t}} v_{i,t+1,r} \} \geq \sum_{j \in u} p_{j,t} f_j \quad \forall t, r_t, u \in \mathcal{U}_{r_t} \end{aligned} \tag{57}$$

[2]Some alternative names are piece-wise linear relaxation bound or capacity-dependent AR bound.

$$v_{i,t,r} \geq 0$$

Let $\lambda_{i,t,j}$ be a set of Lagrange multipliers to the LR relaxation satisfying $\sum_{i \in j} \lambda_{i,j,t} = f_j p_{j,t},\ \forall t, j,$ and let

$$v^{\lambda}_{i,t,r} = \vartheta^{\lambda}_{i,t}(r) - \vartheta^{\lambda}_{i,t}(r-1),$$

and therefore,

$$\vartheta^{\lambda}_{i,t}(r_{i,t}) = \sum_{r=1}^{r_{i,t}} v^{\lambda}_{i,t,r}.$$

Since $\vartheta^{\lambda}_{i,t}(r)$ is an optimal value of the dynamic program on resource i once the problem is decomposed using λ,

$$\vartheta^{\lambda}_{i,t}(r_{i,t}) \geq \sum_{j} \lambda_{i,j,t} u_{i,j,t} + p_{j,t} \vartheta^{\lambda}_{i,t+1}(r_{i,t} - u_{i,j,t}) \ \ \forall r_{i,t}, u \in \mathcal{U}_{i,r_{i,t}}. \tag{58}$$

We show that $\theta_t = 0$ and $v^{\lambda}_{i,t,r}$ are feasible solutions to (sAR) of (57). Equation (58) can be written in terms of $v^{\lambda}_{i,t,r}$ as

$$\sum_{r=1}^{r_{i,t}} v^{\lambda}_{i,t,r} - \sum_{j} p_{j,t} \sum_{r=1}^{r_{i,t} - u_{i,j,t}} v^{\lambda}_{i,t+1,r} \geq \sum_{j \ni i, j \in u} \lambda_{i,j,t} \ \ \forall r_{i,t}, u \in \mathcal{U}_{i,r_{i,t}}.$$

For each $u \in \mathcal{U}_{r_t}$, define the corresponding $\bar{u}_i \in \mathcal{U}_{i,r_{i,t}}$ of feasible subsets on i as follows:

$$\bar{u}_i = \{j | j \ni i, j \in u\}.$$

For a given $t, r_t, u \in \mathcal{U}_{r_t}$, summing over all i, and $\bar{u}_i$:

$$\sum_{i} \{ \sum_{r=1}^{r_{i,t}} v^{\lambda}_{i,t,r} - \sum_{j} p_{j,t} \sum_{r=1}^{r_{i,t} - u_{i,j,t}} v^{\lambda}_{i,t+1,r} \} \geq \sum_{i} \sum_{j \in \bar{u}_i} \lambda_{i,j,t} = \sum_{j \in u} p_{j,t} f_j.$$

as $\sum_{i \in j} \lambda_{i,j,t} = f_j p_{j,t}$, which shows:

Proposition 12 $LR \geq sAR$.

The sAR bound can be computed using the same column generation techniques of Adelman [1].

8 Some further comments

Our contributions in this paper are (i) we give a stronger Lagrangian relaxation bound sLR that conserves all the attractive computational properties of LR, (ii) we give a stronger simulation based perfect-hindsight bound $sPHLP$ by adding constraints at the resource level (iii) we compare the sLR bound with the $sPHLP$ bound and show that the former is always tighter as $N \to \infty$ (iv) we give a new strengthened AR bound sAR and show that it is stronger than the LR bound.

Comparing these seemingly disparate approaches also gives insights into the connections between the methods.

We note that the piecewise-linear control proposed here sAR has in fact been implemented in software developed by Garrett van Ryzin and the author in the mid-1990's (U.S. patent 6263315, [16]).

There are various interesting research directions we identify:

- It would be interesting to obtain a tighter (reasonably computable) bound than sAR.
- LR,sLR, AR, sAR and $sPHLP$ are time consuming and at the time of this writing, unlikely to be feasible for large problems that one sees in airline or e-commerce applications. So faster ways of computing them are of interest.
- The results of this paper can likely be extended to a choice model of customer behavior for a network with a model similar to the ones in ([12],[14]), as most of the models based on the independent class assumption have been extended to a multi-nomial logit type models of customer behavior: see Gallego et al. [6], Liu and van Ryzin [8], Kunnumkal and Topaloglu [7] and Adelman and Zhang [2]. The last two extend the LR and AR methods to network RM under the choice model.
- The relationship between sAR and sLR and $sPHIP$ is intriguing. It would be very interesting if one could prove that sAR is tighter or weaker than the others.

Finally, the sLR and $sPHIP$ bounding technique can potentially be applied to strengthen other applications of Lagrangian relaxation to dynamic programs.

References

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

[2] D Adelman and D. Zhang. An approximate dynamic programming approach to network revenue management with customer choice. Technical report, Graduate School of Business, University of Chicago, Chicago, 2007.

[3] P. P. Belobaba. Optimal vs. heuristic methods for nested seat allocation. Presentation at ORSA/TIMS Joint National Meeting, November 1992.

[4] G. R. Bitran and R. Caldentey. An overview of pricing models and revenue management. *Manufacturing and Service Operations Management*, 5:203–229, 2003.

[5] W. J. Elmaghraby and P. Keskinocak. Dynamic pricing: Research overview, current practices, and future directions. *Management Science*, 2003. (To appear).

[6] G. Gallego, G. Iyengar, R. Phillips, and A. Dubey. Managing flexible products on a network. Technical Report TR-2004-01, Dept of Industrial Engineering, Columbia University, NY, NY, 2004.

[7] S. Kunnumkal and H. Topaloglu. A new dynamic programming decomposition method for the network revenue management problem with customer choice behavior. Technical report, School of IEOR, Cornell University, Ithaca, NY, 2008.

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

[9] J. I. McGill and G. J. van Ryzin. Revenue management: Research overview and prospects. *Transportation Science*, 33:233–256, 1999.

[10] R Phillips. *Pricing and Revenue Optimization.* Stanford Business Books, Stanford, CA, 2005.

[11] R. W. Simpson. Using network flow techniques to find shadow prices for market and seat inventory control. Technical Report Memorandum, M89-1, Flight Transportation Laboratory, MIT, Cambridge, MA, 1989.

[12] K. T. Talluri. Airline revenue management with passenger routing control: A new model with solution approaches. *International Journal of Services Technology and Management*, 2:102–115, 2001.

[13] K. T. Talluri and G. J. van Ryzin. A randomized linear programming method for computing network bid prices. *Transportation Science*, 33:207–216, 1999.

[14] K. T. Talluri and G. J. van Ryzin. Revenue management under a general discrete choice model of consumer behavior. *Management Science*, January 2004.

[15] K. T. Talluri and G. J. van Ryzin. *The Theory and Practice of Revenue Management.* Kluwer, New York, NY, 2004.

[16] K.T. Talluri. Revenue management system and method. United States Patent, 6263315, July 2001.

[17] H. Topaloglu. On the asymptotic optimality of the randomized linear program for network revenue management. Technical report, School of IEOR, Cornell University, Ithaca, NY, 2007.

[18] H. Topaloglu. Using lagrangian relaxation to compute capacity-dependent bid prices in network revenue management. Technical report, School of IEOR, Cornell University, Ithaca, NY, 2007.

[19] E. L. Williamson. Comparison of optimization techniques for origin-destination seat inventory control. Master's thesis, Flight Transportation Laboratory, MIT, Cambridge, MA, 1988.