Les Cahiers du GERAD ISSN: 0711–2440

Time-Dependent Bid Prices for Multi-Period Network Revenue Management Problems

M. Bijvank, A. Haensel,
P. L'Ecuyer, P. Marcotte

G–2012–38

July 2012

Time-Dependent Bid Prices for Multi-Period Network Revenue Management Problems

Marco Bijvank

Rotterdam School of Management
Erasmus University
3062 PA, Rotterdam, The Netherlands
mbijvank@rsm.nl

Alwin Haensel

Department of Mathematics
VU University Amsterdam
1081 HV Amsterdam, The Netherlands
alwin.haensel@vu.nl

Pierre L'Ecuyer*
Patrice Marcotte

DIRO, Université de Montréal
Montréal (Québec) Canada, H3C 3J7
** and GERAD*
lecuyer@iro.umontreal.ca, marcotte@iro.umontreal.ca

July 2012

Les Cahiers du GERAD

G–2012–38

Abstract

In this paper, we propose new mathematical programming approaches for computing time-dependent bid prices in network revenue management problems. In contrast with previous models, ours can accommodate more than one customer request between two successive bid price updates, as frequently occurs in practice. As a first approach, we introduce a simplified version of our time-dependent bid price model in which the random demand is replaced by its expectation in each time period. Next, we propose three heuristic scenario-based stochastic programming methods, whose aim is to improve robustness to uncertainty in the case of stochastic demand. The time-dependent bid prices are found by solving mixed-integer linear programs. In our numerical experiments, our proposed methods outperform the previously-proposed techniques by 2 to 3% on average, in terms of average revenue under the dynamic strategies computed by the methods.

Key Words: Pricing and revenue management; time-dependent bid prices; scenario-based stochastic programming; simulation.

Résumé

Dans cet article, nous proposons une nouvelle approche de programmation mathématique permettant d'établir une politique de prix critiques ("bid prices") en gestion du revenu. En contraste avec les modèles précédents, le nôtre peut tenir compte d'un nombre arbitraire de requêtes entre deux mises à jour des prix critiques, et donc mieux s'adapter au contexte réel. En première approximation, nous considérons un modèle dynamique où la demande stochastique est remplacée par son espérance. Par la suite, et dans le but d'améliorer la robustesse de la solution face aux aléas, nous présentons trois algorithmes heuristiques basés sur la simulation de scénarios. Dans chaque cas, les prix critiques sont déterminés en résolvant un programme linéaire mixte. Notre approche est validée par des tests numériques qui montrent une augmentation de 2 ou 3% du revenu par rapport aux algorithmes classiques.

Mots clés : tarification et gestion du revenu; prix critiques dynamiques; programmation stochastique; simulation.

Acknowledgments: This research has been supported by grants from NSERC-Canada and FQRNT-Québec to the third and fourth authors, and a Canada Research Chair to the third author. The authors would like to thank Huseyin Topaloglu for discussions on the adaptation of the Lagrangian Relaxation with Time-Dependent Bid Prices (LRP) model to include arbitrary time periods between bid price updates.

1 Introduction

Revenue management (RM) involves the optimal allocation of limited resources to stochastic demand with the objective of maximizing revenue. Traditional application areas include airlines, hotels and advertising industries. In this context, each product consumes a certain amount of perishable resources. Demand for the products occurs at random before the expiration date, according to some probability distribution. In the case of airline tickets, the resources are the seats available on the different flight segments and their expiration dates correspond to the departure dates of the associated flights. Each request for a product can be either accepted or rejected, depending on the price, the available resources, and the time that remains before their expiration date, for example. An optimal strategy to make those accept/reject decisions is usually much too hard to compute, so one must rely on heuristics.

In network environments, heuristic strategies based on bid prices are widely used to control the availability of products that consume one or more resources. These strategies specify threshold values, called *bid prices*, associated with each resource, and a request for a product is accepted if and only if its revenue exceeds the sum of the bid prices of the resources involved. One of the traditional approaches proposed in the literature to compute bid prices is to formulate the RM problem as a static model where the total stochastic demand for each product is replaced by its expected value. The optimal dual prices of this simple *deterministic linear program* (DLP) are then used as bid prices. A limitation of this approach is that the distributional information about the demand is ignored and the bid prices are *static*, in the sense that they are assumed to be fixed and constant throughout the booking horizon, whereas the system under control is dynamic. Given that the requests for products arrive according to a stochastic process, intuition suggests that "optimal" bid prices should change dynamically over time, to account for newly available information, such as the remaining capacity for each resource. One simple way of updating the bid prices dynamically is to re-solve the static bid-price model at several epochs in time over the booking horizon, given the expected future demand and the remaining available resources. Williamson (1992) obtained good results with this heuristic.

Even if the demand process is assumed to be deterministic, optimal bid prices are not necessarily constant over time. They are typically decreasing when demand is stationary. Near the end of the time horizon, the bid prices become smaller because (intuitively) we are running out of time to use the resources. Adelman (2007) has shown this for a particular discrete-time dynamic programming (DP) model formulation of the network RM problem, where he assumes at most one customer request in each period, and he replaces the DP value function by an approximation defined as an affine function of the amount of resources available. He formulates the resulting approximate DP problem, which has finite state and decision spaces, as a linear program (LP) whose decision variables represent the time-dependent bid prices. The dual LP problem, which involves much fewer constraints than the primal, is then solved via column generation. For the same discrete-time model, Topaloglu (2009) relaxes the capacity constraints in the DP formulation through the use of Lagrange multipliers. This results in a separable network problem for which he computes time- and capacity-dependent bid prices by focusing on a single resource at a time. In Kunnumkal and Topaloglu (2010), a different relaxation of the capacity constraints is proposed.

All these methods operate on the DP formulation and assume at most one customer request within each time period. While this makes sense if time is divided into very small intervals and bid prices can change continuously, or at a very high frequency, this is not realistic in many applications, where updates occur daily, for instance. Furthermore, this assumption does not allow group requests. For instance, Figure 1 shows the number of seats sold for a direct flight from Amsterdam to a Spanish airport performed by Transavia over a booking horizon of 73 days. We only depict those days during which the product was available to customers (the product was not for sale on days 17 and 18). In this example, if the bid prices are updated on a daily basis, it is clear that multiple requests for this product occur between updates. In general, a model should allow an arbitrary number of requests between bid price updates. It is also important that the specific epochs at which bid prices are updated are taken into account when determining the control policy, because requests can only be accepted if sufficient remaining capacity is available for each of the resources used by the products. Restricting the approximation of the DP value function to an affine function, as in the method of Adelman

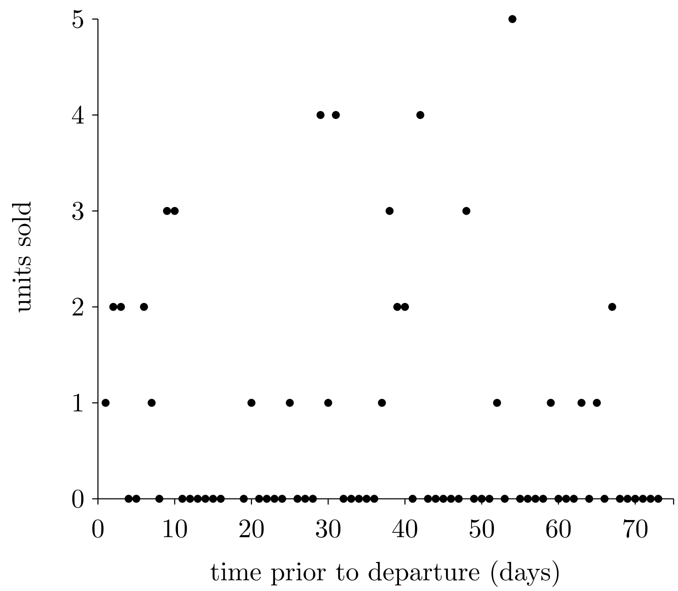

Figure 1: The daily sales for a flight from Amsterdam to Spain.

(2007), is a second source of error that may provide solutions with a significant optimality gap. Another drawback of the approaches of Adelman (2007) and Topaloglu (2009) is their substantial requirements in terms of computation times and implementation. The CPU times reported by Adelman (2007) increase rapidly with the number of time periods and the size of the network. Thus, even though the simple DLP formulation does not model the RM problem realistically, its simplicity and computational efficiency are appealing.

The aim of this paper is to propose and compare new methods to compute time-dependent bid prices that achieve a good trade-off between computational efficiency on the one hand, and quality and robustness of the solutions on the other hand. We consider a setting where the booking horizon is partitioned into T time *periods* of arbitrary lengths, not necessarily equal. These periods can be large and may contain multiple booking and group requests. The bid prices must be constant within each period, but are allowed to vary across periods. The problem of selecting bid prices that maximize the total expected revenue can be formulated as a stochastic DP problem, where the vector of bid prices determined at the beginning of each period may depend on both the period number and the amount of resources currently available. One could conceivably solve this problem by approximating the DP value function (e.g., by a linear combination of a fixed set of basis functions; see Adelman (2007)) and by approximating numerically the expectation in the DP equation, at each time step. However, an accurate approximation becomes impractical from the computational viewpoint as soon as the number of resources exceeds a few units, due to the high-dimensional state space and the difficulty of computing the expectation. One must therefore either settle for a crude approximation of the value function, such as the affine approximation proposed by Adelman (2007), for which the approximation error can be significant, or for an alternative approximation scheme or heuristic approach.

The approximate solution methods proposed in this paper are based on novel mathematical programming formulations that build over the classical DLP formulation, in which we introduce additional variables to account for time-dependent bid prices. In the first approach, the random demand in each time period is simply replaced by its expectation. The bid prices are optimized under the pessimistic assumption that requests for a given product in a given period can be accepted only if the aggregate demand for the product in that period can be satisfied. This is a crude approximation because the resulting expected revenue may deviate significantly from its true value when demands are replaced by their expectations. For this reason, we propose three methods to improve robustness by taking into account the stochastic nature of the demand.

These methods generate a finite set of demand scenarios, via simulation, and perform the optimization on a sample version of the problem. Although their common aim is to compute efficient bid prices corresponding to the scenarios generated, they differ in the way they deal with the capacity constraints. The first method associates a null revenue to demand allocations that violate the capacity constraint, the second method incorporates conditional expectation constraints on any overbookings, and the third method penalizes the overbookings in the objective function. Note that overbookings (selling more than the capacity) are only considered when computing the bid prices, whereas demand that cannot be met is actually lost in our setting.

The bid prices can be updated at the beginning of a preselected set of time periods, taking into account the current state of demand and the availability of resources. The optimization problem is then reformulated and solved only for the demand and bid prices over the future periods. It is also possible, as a further simplification, to compute the time-dependent bid prices only once, at the beginning, and use them without any updating. We will assess the performance of both these strategies through simulations, and compare them with (adaptations of) previously proposed methods.

The rest of the paper is organized as follows. In Section 2, we define the basic model and state the problem. An overview on related literature is presented in Section 3, to position our work. In Section 4 we propose a new mixed integer programming formulation to compute time-dependent bid prices based on deterministic but time-varying demand. The stochastic nature of the demand is taken into account in our formulations of Section 5, where we develop new scenario-based stochastic programming methods to derive robust bid prices. In Section 6, the performance of all these methods is tested and compared to previously proposed ones that compute time-dependent bid prices.

2 Problem Formulation and a Time-Dependent Bid Price Model

Consider a network that has m *resources* indexed by i, each with an initial capacity of c_i units. There are n *products* indexed by $j \in \{1, \ldots, n\}$, the set of all products, and r_j represents the (fixed) *revenue* obtained when one unit of product j is sold. Let $\mathbf{c} = (c_1, \ldots, c_m)^\top$ and $\mathbf{r} = (r_1, \ldots, r_n)^\top$. The *incidence matrix* is $\mathbf{A} = [a_{ij}] \in \mathbb{R}^{m \times n}$, where a_{ij} defines the number of units of resource i required by product j. Thus, $a_{ij} = 0$ if product j does not require any unit of resource i, and $a_{ij} > 0$ otherwise. This notation allows group requests, where each request is either accepted or rejected in total (no partial acceptance).

Throughout the paper, we use lowercase bold symbols for vectors and capital bold symbols for matrices. We use a subscript to represent the column vector of a matrix, and a superscript for a row vector. For example, the j-th column vector $\mathbf{A}_j$ indicates the subset of resources used by product j, and the i-th row vector $\mathbf{A}^i$ indicates the subset of products that consume resource i. The notation $i \in \mathbf{A}_j$ means that resource i is used by product j, and $j \in \mathbf{A}^i$ means that product j uses resource i. The state of the network is described by a vector $\mathbf{x} = (x_1, \ldots, x_m)^\top$, which indicates the amount of resource capacities still available. When a unit of product j is sold, the state of the network changes to $\mathbf{x} - \mathbf{A}_j$.

Demands for products arrive during the time interval $[0, T_\text{B}]$, for a fixed *booking horizon* T_B, according to some stochastic process. Each request is either accepted or declined upon its arrival. After time T_B, any unsold capacity becomes worthless. The accept/reject decisions should be made in a way that maximizes the total expected revenue. If we assume that the time intervals between successive requests are independent, the problem can be formulated as a dynamic program (more specifically, a semi-Markov decision process) for which the state at time $\tau \in [0, T_\text{B}]$ is $(\tau, \mathbf{x})$, where $\mathbf{x}$ denotes the vector of resources available at time τ, and the value function $V(\tau, \mathbf{x})$ represents the optimal expected revenue from time τ onward, provided that a single request arises at instant τ and that the current state is $(\tau, \mathbf{x})$. This value function satisfies the DP recurrence equation

$$V(\tau, \mathbf{x}) = \begin{cases} \mathbb{E}[\max\{V(\tau+\xi, \mathbf{x}),\ (r_\iota + V(\tau+\xi, \mathbf{x} - \mathbf{A}_\iota)) \cdot \mathbb{I}[\tau + \xi < T_\text{B} \text{ and } \mathbf{x} \geq \mathbf{A}_\iota]\}] & \text{if } \tau < T_\text{B} \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $\mathbb{I}$ denotes the indicator function, $\tau + \xi$ is the time of the next request, which is for product ι, and the expectation $\mathbb{E}$ is with respect to the random pair (ξ, ι). An optimal strategy consists in a binary function which to each $(\tau + \xi, \mathbf{r}, \iota)$, accepts a request for product ι when the system is in state $(\tau + \xi, \mathbf{r})$ if and only if the maximum in (1) is achieved by the right-hand term in the expectation.

In principle, we could define a parametric approximation for $V(\tau, \mathbf{x})$, plug it into the recurrence equation, and "optimize" the parameters, for example via least squares, after computing the values at a finite number of points. This is generally difficult to implement and very time-consuming. Moreover, even if the optimal strategy could be computed, it would be too complicated to implement in applications. For these reasons, practitioners restrict themselves a priori to simpler classes of suboptimal strategies, and this is what we shall do here. More specifically, we assume that the accept/reject decisions are made on the basis of bid prices, and that those bid prices can only be updated at a finite number of fixed epochs. This type of strategy is very common in real-life applications, for example in airlines and rail.

In our model, we assume that the interval $[0, T_{\mathrm{B}}]$ is partitioned into T time periods, numbered by $t = 1, \ldots, T$. Period 1 starts at time 0 and period T ends at time T_{B}. The demand for product j during period t is a random variable D_{jt} with known distribution, and we assume that each demand arrives at a time uniformly distributed over this period, independently across demands. We assume that the D_{jt}'s are mutually independent.

Customer' requests are either accepted or rejected, immediately upon their arrival, on the basis of bid prices. In period t, the policy is given by a vector of *bid prices* $\boldsymbol{\pi}_t = (\pi_{1t}, \ldots, \pi_{mt})^\top$, where product j is said to be *open for bookings* in period t when it generates a revenue at least as large as the sum of bid prices for the resources it needs. That is,

$$r_j \geq \sum_{i \in \mathbf{A}_j} \pi_{it} = \mathbf{A}_j^\top \boldsymbol{\pi}_t. \tag{2}$$

A customer request for product j in period t is *accepted* if (2) holds and if all the required resources for this product are available when the request arrives.

Under an open-loop control, bid prices for all periods are selected at the beginning of the first period and cannot be modified subsequentely. Under a dynamic (closed-loop) control, bid prices can be re-optimized at the beginning of any period. An optimal strategy to set these bid prices can be derived from the solution of a stochastic discrete-time DP problem. This requires to compute (perhaps approximately) the DP value function $V_t(\mathbf{x})$ for each t, where $V_t(\mathbf{x})$ represents the optimal expected future revenue at the beginning of period t when the remaining capacity is $\mathbf{x}$. This computation quickly becomes impractical when m gets large. This leads us to heuristic ways of selecting the bid prices.

The methods proposed in this paper do not attempt to solve the DP problem directly nor to approximate the DP value function. They are heuristics which solve an LP problem to compute all the future bid prices $\boldsymbol{\pi}_t, \ldots, \boldsymbol{\pi}_T$ based on the remaining capacity $\mathbf{x}_t$ at the beginning of period t and on specific values for the future demands. These demand values are obtained either by replacing each future demand by its expectation, or by generating scenarios (samples) for the demands via simulation and solving an LP problem that maximizes the revenues over these scenarios. In the case of a closed-loop control, this LP is re-solved at the beginning of each period t, based on the new information $\mathbf{x}_t$, whereas in an open-loop control, it is solved only for $t = 1$ with $\mathbf{x}_1 = \mathbf{c}$. In our numerical experiments, we will illustrate and compare the performance of our heuristic methods for the closed-loop and open-loop situations.

3 Related Literature

Bid-price control policies were introduced and studied extensively by Simpson (1989) and Williamson (1992). A theoretical foundation for this control strategy and the use of deterministic linear programming (DLP) to compute bid prices is provided by Talluri and van Ryzin (1998), who proved that a bid-price policy is asymptotically optimal as the demand and resource capacities grow large. In the DLP approach the following

LP is solved to determine the amount of capacity that should be assigned to each product in period τ (denoted by $\mathbf{Y}_\tau$):

$$
\begin{aligned}
V_t^{\mathrm{DLP}}(\mathbf{x}) := \quad \max \quad & \mathbf{r}^\top \cdot \sum_{\tau=t}^{T} \mathbf{Y}_\tau && \text{(3a)}\\
\text{s.t.} \quad & \mathbf{A} \cdot \sum_{\tau=t}^{T} \mathbf{Y}_\tau \le \mathbf{x} && \text{(3b)}\\
& \sum_{\tau=t}^{T} \mathbf{Y}_\tau \le \sum_{\tau=t}^{T} \mathbf{D}_\tau && \text{(3c)}\\
& \mathbf{Y} \in \mathbb{R}^{n\times(T-t+1)} && \text{(3d)}
\end{aligned}
$$

where $\mathbf{D}_\tau = (d_{1\tau}, \ldots, d_{n,\tau})^\top \in \mathbb{R}^n$ represents the expected demand for each product in time period τ. The bid prices are represented by the dual variables of the capacity constraints (3b).

Besides this DLP method, alternative methods that incorporate information on the demand distribution have been proposed. Talluri and van Ryzin (1998) reformulate the DLP as a *probabilistic non-linear program* (PNLP) where demand randomness is captured in the objective function. That is, the objective function in (3a) is replaced by $\sum_{j=1}^{n} r_j \cdot \sum_{\tau=t}^{T} \mathbb{E}[\min\{y_{j\tau}, D_{j\tau}\}]$ and the constraints (3c) are removed. Based on simulation studies it has been shown by Williamson (1992), Talluri and van Ryzin (1998) and de Boer et al. (2002), that simple probabilistic methods like PNLP produce bid prices that consistently underperform with respect to the DLP bid prices, because PNLP ignores a so-called nesting property where high-fare classes have access to all the capacity reserved for lower fare classes. As a result, the mathematical programming models like DLP tend to overprotect capacities for high-fare classes. Since PNLP suffers from this deficiency (de Boer et al. 2002), more sophisticated stochastic optimization techniques are required to incorporate the dynamics and randomness of the booking process.

Talluri and van Ryzin (1999) propose to replace the expected demand in DLP by the actual stochastic random variable, where demand samples are randomly generated and the DLP method is solved for each sample, and the average of the dual prices is being used as bid prices. For this *randomized linear programming* (RLP) method, numerical experiments indicate that RLP provides a small improvement over DLP. Topaloglu (2009) shows that the bid prices based on RLP are asymptotically optimal, similar to the results of Talluri and van Ryzin (1998) for DLP.

Based on sample path derivatives, Topaloglu (2008) implemented a stochastic gradient algorithm to compute static bid prices. Similar techniques have been proposed by Bertsimas and de Boer (2005) and van Ryzin and Vulcano (2008) to find good parameter values for a booking limit policy. In such a policy, booking limits are considered on each resource-fare-class combination and a request is accepted if the number of units sold remains within the associated limits in contrast to making decisions on the price in a bid-price policy. A policy closely related to bid-price control is introduced by Bertsimas and Popescu (2003), where a threshold price is imposed on each product instead of assuming the additive structure of the bid prices for each resource. The authors compare this policy to the traditional bid-price policy, and show that it converges to the optimum, using arguments similar to those of Talluri and van Ryzin (1998). We refer to Talluri and van Ryzin (2004) for a comprehensive overview of the different types of control policies and techniques to set the control parameters.

An alternative approach widely used in the literature is stochastic programming (SP), where demand is represented by a scenario tree based on some probability distribution. de Boer et al. (2002) reformulate the PNLP as a two-stage stochastic program, Higle (2007) proposes an improved two-stage model, whereas Chen and Homem-de Mello (2010) provide theoretical insights for this model. Multistage stochastic programming models have been developed for booking limits by Möller et al. (2008) and for bid prices by DeMiguel and Mishra (2008). In this context, we also mention Cooper and Homem-de Mello (2007), who propose a two-stage policy in which simple decision rules are used first and, as time progresses, an optimal policy is derived at the end of the booking horizon.

Multistage stochastic programming directly incorporates the dynamic behavior of the RM problem since the policy parameter values can change over time. Traditionally, static solution models are resolved frequently over the booking horizon to revise the parameters, where actual demand realizations during the booking process are taken into consideration. In practice, these revisions of the control policy occur most often in overnight runs of the optimization algorithm. However, it is well known that re-solving the DLP model does not always improve the policy (Cooper 2002), whereas re-solving an SP model can only result in improvements (Chen and Homem-de Mello 2010). Sufficient conditions under which re-solving does not worsen the performance of the control policy is provided by Secomandi (2008).

Akan and Ata (2008) also study adapted bid-price control and establish that optimal (adapted) bid prices form a martingale for the discrete-time model. Furthermore, they quantify the relationship between the updating frequency of the bid prices and their performance, which provides insights on how to choose the updating frequency. The martingale property is extended to a continuous-time model by Akan and Ata (2009). The usefulness of re-solving is also illustrated for control policies other than bid-price control (see Reiman and Wang (2008), Jasin (2012)).

Besides re-solving a static model, one can capture the dynamics of the RM model through a single model that relies on time-dependent bid prices of a particular form. Klein (2007) proposes a simple linear function, with basic bid prices that decrease linearly over time (either directly or based on expected future demands) and increase linearly when a product is sold. The principles underlying this approach are the perishability and scarcity effects, respectively (Berman et al. 2010). Adelman (2007) validates these results with a new LP model based on an affine approximation of the optimal stochastic dynamic programming value function. A drawback of this approach is that the number of constraints grows exponentially with the number of resources. In a related model by Farias and van Roy (2007), the number of constraints grows linearly. Kirshner and Nediak (2012) consider the continuous-time version of the model studied by Adelman (2007). Tong and Topaloglu (2011) revise the approach of Adelman (2007) and reformulate the problem as a minimum-cost network flow problem. As a result, the model of Adelman (2007) reduces to a linear program in which the number of constraints and decision variables increases linearly with the number of resources and products.

Alternatively, Topaloglu (2009) dualizes the capacity constraints that link decisions for different resources in the original DP formulation, through the use of Lagrange multipliers. As a result, the network problem decomposes by resources and time-dependent bid prices can easily be computed. Kunnumkal and Topaloglu (2010) follow a similar approach, but use a different relaxation of the capacity constraints. Tong and Topaloglu (2011) show that this approach produces the same upper bound on the original DP formulation as that of Adelman (2007).

In contrast with the works previously described, we propose an extension of DLP where available resources are assigned to time-varying demand, and where bid prices are primal rather than dual decision variables. To relax the deterministic demand assumption, we propose new scenario-based stochastic programming methods where bid prices maximize the total expected revenue over a finite set of demand scenarios, the latter being generated through simulation.

4 Time-Dependent Bid Prices Based on Expected Demand (TDED)

The first heuristic that we consider is quite simple: We replace all (random) future demands D_{jt} by their expectation d_{jt}, and then we solve an LP that selects all future bid prices to maximize the future revenue, given the current state $\mathbf{x}$. At the beginning of period t, the primary decision variables of this LP are the bid-price vectors $\boldsymbol{\pi}_t, \ldots, \boldsymbol{\pi}_T$. We collect the expected demands in the matrix $\mathbf{D} = [d_{jt}] \in \mathbb{R}^{n \times T}$, where the t-th column vector $\mathbf{D}_t$ is the expected demand for each product in time period t. The allocation of products to customer demands is captured by the matrix $\mathbf{Y} = [y_{jt}] \in \mathbb{R}^{n \times T}$, where y_{jt} denotes the number of products j assigned in period t, which is $y_{jt} = d_{jt}$ when (2) is satisfied (product j is open for bookings in period t), and $y_{jt} = 0$ otherwise. We introduce assignment variables $\mathbf{Z} = [z_{jt}] \in \mathbb{R}^{n\times T}$, where

$$z_{jt} = \begin{cases} 1 & \text{if product } j \text{ is open for bookings in period } t, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

and slack variables $\mathbf{U} = [u_{jt}]$ and $\mathbf{V} = [v_{jt}] \in \mathbb{R}^{n\times T}$ that represent the difference between the price for a product j and the sum of the bid prices for the involved resources, in period t:

$$u_{jt} = \begin{cases} r_j - \mathbf{A}_j^\top \boldsymbol{\pi}_t & \text{if } z_{jt} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

and

$$v_{jt} = \begin{cases} \mathbf{A}_j^\top \boldsymbol{\pi}_t - r_j & \text{if } z_{jt} = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

For a given time period t and remaining capacity $\mathbf{x}$, our problem formulation based on the expected demand is

$$V_t^{\text{TDED}}(\mathbf{x}) := \quad \max \quad \mathbf{r}^\top \cdot \sum_{\tau=t}^{T} \mathbf{Y}_\tau \tag{7a}$$

$$\text{s.t.} \quad \mathbf{A} \cdot \sum_{\tau=t}^{T} \mathbf{Y}_\tau \leq \mathbf{x} \tag{7b}$$

$$\mathbf{Y}_\tau = \mathbf{D}_\tau \cdot (\mathbf{Z}_\tau)^\top \tag{7c}$$

$$\mathbf{r} = \mathbf{A}^\top \cdot \boldsymbol{\pi}_\tau + \mathbf{U}_\tau - \mathbf{V}_\tau \tag{7d}$$

$$\mathbf{U}_\tau \leq K \cdot \mathbf{Z}_\tau \tag{7e}$$

$$\mathbf{V}_\tau \leq K \cdot (\mathbf{1} - \mathbf{Z}_\tau) \tag{7f}$$

$$\mathbf{Z}_\tau \cdot \mathbf{V}_\tau \neq \mathbf{0} \tag{7g}$$

$$\boldsymbol{\pi}_\tau, \mathbf{U}_\tau, \mathbf{V}_\tau \geq \mathbf{0} \tag{7h}$$

$$\mathbf{Y} \in \mathbb{R}^{n\times(T-t+1)}, \quad \mathbf{Z} \in \{0,1\}^{n\times(T-t+1)} \tag{7i}$$

where K is a large constant, and the constraints (7c) to (7h) that depend on τ must hold for $t \leq \tau \leq T$. In the objective function (7a), the total revenue is maximized. Constraints (7b) specify that no more capacity is allocated than is available, whereas constraints (7c) state that all (expected) demand $d_{j\tau}$ is allocated if $z_{j\tau} = 1$ and nothing is allocated otherwise. This is somewhat pessimistic, because in reality, part of the demand in a given period could be satisfied, given the available capacity. In Section 5.1, we relax this assumption. The bid-price constraints are specified in (7d)–(7h), where (7d) specifies the bid prices and (7e)–(7h) ensure that if $z_{j\tau} = 1$ then $u_{j\tau} \geq 0$ and $v_{j\tau} = 0$, and if $z_{j\tau} = 0$ then $u_{j\tau} = 0$ and $v_{j\tau} > 0$. Note that (7g) is required to exclude that $z_{j\tau}$ and $v_{j\tau}$ are both zero, this would violate (2); if $z_{j\tau}$ equals zero then the sum of the bid prices corresponding to the resources $\mathbf{A}_j$ should be strictly larger than r_j, i.e.. $v_{j\tau}$ should have a positive value. For implementation purposes, constraint (7g) can be rewritten as $\mathbf{Z}_\tau + \mathbf{V}_\tau \geq \boldsymbol{\epsilon}$ where $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^\top$ is a vector of small positive constants (e.g., in our experiments, we used $\epsilon_j = 10^{-6}$).

The idea behind TDED is related to DLP. Both problem formulations attempt to assign the expected demand to the resources available. In TDED, the demand is split into time periods, whereas DLP only considers the total demand over the entire booking horizon. Furthermore, TDED incorporates the time-dependent bid prices into the model, whereas DLP uses duality to compute the static bid prices. As a result, the TDED model requires the extra variables and constraints in (7d)–(7h).

5 Time-Dependent Bid Prices Based on Simulated Demand Scenarios

In this section, we propose three heuristics that account for the stochastic nature of the demand when selecting the bid prices. The main idea underlying these methods is to simulate demand samples and to find one set of bid prices that maximize the future revenue over all sample scenarios under some conditions to promote "robustness" of the solutions. This differs from RLP of Talluri and van Ryzin (1999), where the bid prices are computed for each realization of the demand and those bid prices are averaged to obtain the final set of static bid prices. We generate S demand scenarios, $\mathbf{D}^{(s)} = [d_{jt}^{(s)}] \in \mathbb{R}^{n\times T}$ for $s = 1, \dots, S$, based on the probability distribution of the demand process. The allocations $\mathbf{Y}^{(s)} = [y_{jt}^{(s)}] \in \mathbb{R}^{n\times T}$ have to be specified per scenario s as well. The positive integer S is a parameter that must be selected by the user. Using a larger S generally provides more robust solutions, but is computationally more expensive.

Since the realized demand will generally differ from the demand scenarios sampled with simulation, it is not optimal to select the bid prices so that allocations exactly satisfy the demand in all scenarios whenever a product is open for bookings. This means that the constraints (7c) have to be modified to deal with overbookings. In a first approach, we introduce a notion of virtual overbookings, whereas in the second approach we add constraints on actual overbookings in the sample scenarios, and in the third approach we penalize the overbookings. These three approaches are discussed in Section 5.1 to 5.3, respectively.

5.1 Virtual Overbookings with Time-dependent Bid Prices (VOTD)

When a requested product j is open for bookings but the available resources are no longer sufficient to accept the customer request in scenario s, i.e., when there is a resource $i \in \mathbf{A}_j$ such that $(\mathbf{A}^i)^\top \cdot \sum_{\tau\le t} \mathbf{D}_\tau^{(s)}\mathbf{Z}_\tau > x_i$, the request cannot be satisfied. To account for this possibility in our LP formulation, we introduce variables $w_{jt}^{(s)}$ that represent *virtual overbookings* on product j in period t, for scenario s. That is, in scenario s, we have $y_{jt}^{(s)} + w_{jt}^{(s)} = d_{jt}^{(s)} z_{jt}$ while $\mathbf{A}_j \sum_{\tau\le t} y_{j\tau}^{(s)} \le \mathbf{x}$. Virtual overbookings are assignments to customer demand that do not result in a sale, and therefore they do not generate revenue nor do they reduce the available capacity. Once a resource i is virtually overbooked, it cannot be allocated anymore to satisfy demand. To handle this situation, we introduce binary variables $\overline{w}_{it}^{(s)}$ that indicate whether a resource i requires virtual overbookings to satisfy demand ($\overline{w}_{it}^{(s)} = 1$), or whether it has sufficient capacity remaining to satisfy the demand in time period t for scenario s ($\overline{w}_{it}^{(s)} = 0$):

$$\overline{w}_{it}^{(s)} = \begin{cases} 0 & \text{if } (\mathbf{A}^i)^\top \cdot \sum_{\tau<t} \mathbf{D}_\tau^{(s)}\mathbf{Z}_\tau < x_i \\ 1 & \text{otherwise.} \end{cases}$$

This results in the following *Virtual Overbookings with Time-Dependent Bid Prices* (VOTD) formulation (here and in the other problem formulations in this paper, the constraints that depend on τ and/or s must hold for $t \le \tau \le T$ and/or $1 \le s \le S$, unless stated otherwise as in (8e) and (8f)):

$$rcllV_t^{\text{VOTD}}(\mathbf{x}) = \max \quad \frac{1}{S}\mathbf{r}^\top \cdot \sum_{s=1}^{S}\sum_{\tau=t}^{T} \mathbf{Y}_\tau^{(s)} \tag{8a}$$

$$\text{s.t.} \quad \mathbf{A} \cdot \sum_{\tau=t}^{T} \mathbf{Y}_\tau^{(s)} \le \mathbf{x} \tag{8b}$$

$$\mathbf{Y}_\tau^{(s)} + \mathbf{W}_\tau^{(s)} = \mathbf{D}_\tau^{(s)} \cdot (\mathbf{Z}_\tau)^\top \tag{8c}$$

$$\mathbf{W}_\tau^{(s)} \le L \cdot \mathbf{A}^\top \cdot \overline{\mathbf{W}}_\tau^{(s)} \tag{8d}$$

$$\overline{\mathbf{W}}_{\tau-1}^{(s)} \le \overline{\mathbf{W}}_\tau^{(s)} \qquad t < \tau \le T, 1 \le s \le S \tag{8e}$$

$$\mathbf{A} \cdot \mathbf{Y}_\tau^{(s)} \le L(\mathbf{1} - \overline{\mathbf{W}}_{\tau-1}^{(s)}) \qquad t < \tau \le T, 1 \le s \le S \tag{8f}$$

$$\mathbf{r} = \mathbf{A}^\top \cdot \boldsymbol{\pi}_\tau + \mathbf{U}_\tau - \mathbf{V}_\tau \tag{8g}$$

$$\mathbf{U}_\tau \le K \cdot \mathbf{Z}_\tau \tag{8h}$$

$$\mathbf{V}_\tau \le K \cdot (\mathbf{1} - \mathbf{Z}_\tau) \tag{8i}$$

$$\mathbf{Z}_\tau \cdot \mathbf{V}_\tau \neq \mathbf{0} \tag{8j}$$

$$\boldsymbol{\pi}_\tau, \mathbf{U}_\tau, \mathbf{V}_\tau \geq \mathbf{0} \tag{8k}$$

$$\mathbf{Z} \in \{0,1\}^{n\times(T-t+1)}, \quad \overline{\mathbf{W}}^{(s)} \in \{0,1\}^{n\times(T-t+1)} \tag{8l}$$

$$\mathbf{Y}_\tau^{(s)}, \mathbf{W}_\tau^{(s)} \geq \mathbf{0} \tag{8m}$$

where L is a sufficiently large real number, for instance $L = \max_{i,s}(\mathbf{A}^i)^\top \sum_{\tau=t}^T \mathbf{D}_\tau^{(s)}$, and $\mathbf{0}$ is a zero vector of the appropriate dimension. The objective function (8a) and the capacity constraints (8b) are similar to those in the basic TDED model. The demand in scenario s is satisfied with the assignment variables $\mathbf{Y}^{(s)} = [y_{jt}^{(s)}]$ and $\mathbf{W}^{(s)} = [w_{jt}^{(s)}]$ in (8c). Constraints (8d) allow virtual overbooking assignments $w_{j\tau}^{(s)}$ in (8c) if there is at least one resource $i \in \mathbf{A}_j$ that is virtually overbooked (i.e., $\overline{w}_{i\tau}^{(s)} = 1$), whereas (8e) ensures that once a resource is virtually overbooked it retains this status throughout the remainder of the booking horizon. Constraints (8f) enforce that no product is sold if one of its resources is virtually overbooked (i.e., $y_{j,\tau}^{(s)} = 0$ if $\overline{w}_{i,\tau-1}^{(s)} = 1$ for some $i \in \mathbf{A}_j$). Note that this formulation first assigns the demand in scenario s to $\mathbf{Y}^{(s)}$ before the demand is assigned to the virtual overbookings $\mathbf{W}^{(s)}$, because the virtual overbookings do not result in allocations that generate revenue. The constraints (8g)–(8k) set the bid prices and the slack variables similar to TDED.

5.2 Restricted Overbookings with Time-Dependent Bid Prices (ROTD)

In the VOTD approach, virtual overbookings do not result in sales. We now introduce an alternative approach where actual overbookings are allowed and generate revenue. To keep the overbookings under control, we impose an upper bound on the *conditional Value-at-Risk* (CVaR) associated with the number of overbookings in our sample scenarios. The CVaR (also called the *expected shortfall* and the *average value-at-risk* in the literature) is a well-known risk measure in portfolio management and is defined as follows (Rockafellar and Uryasev 2000, 2002). Let X be a random variable that represents a *loss* of a certain form. In our context, X is the number of overbookings. The *Value-at-Risk* (VaR) at confidence level α, for $0 < \alpha < 1$, is defined as the α quantile of the distribution of X, written as

$$\mathrm{VaR}_{1-\alpha}(X) \quad = \quad F^{-1}(\alpha) \;:=\; \inf\{x | F(x) \geq \alpha\}, \tag{9}$$

where $F(\cdot)$ is the cumulative distribution function of X, and the CVaR is the expected loss conditional on the event that the loss exceeds the VaR:

$$\mathrm{CVaR}_{1-\alpha}(X) \quad = \quad \mathbb{E}[X | X \geq \mathrm{VaR}_\alpha(X)]. \tag{10}$$

In our case, $\mathrm{VaR}_{1-\alpha}(X)$ is the smallest number of overbookings that is exceeded with probability no larger than $1-\alpha$, and $\mathrm{CVaR}_{1-\alpha}(X)$ is the expected number of overbookings when this number exceeds the VaR. A graphical illustration of the VaR and CVaR is given in Figure 2. The concept of VaR has been used in revenue management by Koenig and Meissner (2010), but the use of CVaR is new in this field.

To incorporate the CVaR constraints into our mathematical programming formulation, we will use a formulation developed by Rockafellar and Uryasev (2000, 2002), who have expressed the VaR and CVaR for empirical (or discrete) distributions as the solution of a linear program. To state this linear program in our context, where we are interested in the VaR and CVaR of the empirical distribution of overbooking realizations, we introduce the following notation. Consider the allocations $\mathbf{Y}^{(s)}$ of the remaining capacity $\mathbf{x}$ for the demand scenarios $\mathbf{D}^{(s)}$, for $s = 1, \ldots, S$. The total number of overbookings in periods t to T for resource i in scenario s is

$$O_{t,i}^{(s)} = \max\left\{0, \sum_{\tau=t}^{T} \sum_{j\in\mathbf{A}_i} y_{j\tau}^{(s)} - x_i\right\}.$$

For α and t fixed, we consider the empirical distribution of $O_{t,i}^{(1)}, \ldots, O_{t,i}^{(S)}$, which assigns probability $1/S$ to each value, and let X_i be the associated discrete random variable. Let $w_i = \mathrm{VaR}_{1-\alpha}(X_i)$ and $\eta_i^{(s)} = \max[0, O_{t,i}^{(s)} - w_i]$, where the latter is the overbooking realization in excess of the VaR when this excess is

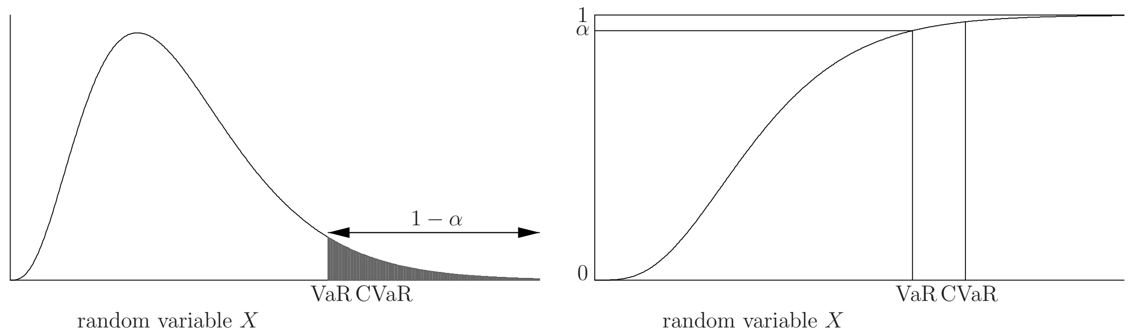

Figure 2: The probability density or mass function (left) and the cumulative distribution function (right) of a random variable X, where VaR and CVaR illustrate the concept of conditional Value-at-Risk.

positive, and 0 otherwise. Since a fraction $1-\alpha$ of the realizations $O_{t,i}^{(s)}$ exceeds the VaR , $\text{CVaR}_{1-\alpha}(X_i)$ is the average of $\eta_i^{(1)},\ldots,\eta_i^{(S)}$ divided by $(1-\alpha)S$, plus w_i. Let $\mathbf{w}=(w_1,\ldots,w_m)^\top$, $\boldsymbol{\eta}^{(s)}=(\eta_1^{(s)},\ldots,\eta_m^{(s)})^\top$ for $s=1,\ldots,S$, and $\boldsymbol{\gamma}=(\text{CVaR}_{1-\alpha}(X_1),\ldots,\text{CVaR}_{1-\alpha}(X_m))^\top$. Rockafellar and Uryasev (2000, 2002) show that the vectors $\mathbf{w}$ and $\boldsymbol{\gamma}$ of VaR's and CVaR's can be computed by solving the following linear program (the constraints are for $1\le s\le S$):

$$\boldsymbol{\gamma}= \quad \min \quad \mathbf{w}+\frac{1}{1-\alpha}\cdot\frac{1}{S}\sum_{s=1}^{S}\boldsymbol{\eta}^{(s)} \tag{11a}$$

$$\text{s.t.} \quad \mathbf{w}+\boldsymbol{\eta}^{(s)}\ge \mathbf{A}\cdot\sum_{\tau=t}^{T}\mathbf{Y}_\tau^{(s)}-\mathbf{x} \tag{11b}$$

$$\mathbf{w},\boldsymbol{\eta}^{(s)}\ge\mathbf{0}. \tag{11c}$$

In our model, we will impose that this empirical CVaR vector $\boldsymbol{\gamma}$ does not exceed a predefined vector of bounds $\mathbf{b}=(b_1,\ldots,b_m)$. This means that we are limiting the average number of overbookings within the $1-\alpha$ worst overbooking realizations. This is easily incorporated in the mathematical programming formulation by adding a linear constraint that bounds the expresssion (11a), together with the constraints (11b). This results in the following *Restricted Overbookings with Time-Dependent Bid Prices* (ROTD) model formulation (the constraints that depend on τ and/or s hold for $t\le\tau\le T$ and/or $1\le s\le S$):

$$V_t^{\text{ROTD}}(\mathbf{x})= \quad \max \quad \frac{1}{S}\mathbf{r}^\top\cdot\sum_{s=1}^{S}\sum_{\tau=t}^{T}\mathbf{Y}_\tau^{(s)} \tag{12a}$$

$$\text{s.t.} \quad \mathbf{A}\cdot\sum_{\tau=t}^{T}\mathbf{Y}_\tau^{(s)}-\boldsymbol{\eta}^{(s)}-\mathbf{w}\le\mathbf{x} \tag{12b}$$

$$\mathbf{w}+\frac{1}{1-\alpha}\cdot\frac{1}{S}\sum_{s=1}^{S}\boldsymbol{\eta}^{(s)}\le\mathbf{b} \tag{12c}$$

$$\mathbf{Y}_\tau^{(s)}=\mathbf{D}_\tau^{(s)}\cdot(\mathbf{Z}_\tau)^\top \tag{12d}$$

$$\mathbf{r}=\mathbf{A}^\top\cdot\boldsymbol{\pi}_\tau+\mathbf{U}_\tau-\mathbf{V}_\tau \tag{12e}$$

$$\mathbf{U}_\tau\le K\cdot\mathbf{Z}_\tau \tag{12f}$$

$$\mathbf{V}_\tau\le K\cdot(\mathbf{1}-\mathbf{Z}_\tau) \tag{12g}$$

$$\mathbf{Z}_\tau\cdot\mathbf{V}_\tau\ne\mathbf{0} \tag{12h}$$

$$\boldsymbol{\pi}_\tau,\mathbf{U}_\tau,\mathbf{V}_\tau\ge\mathbf{0} \tag{12i}$$

$$\mathbf{Z}^{(s)}\in\{0,1\}^{n\times(T-t)} \tag{12j}$$

5.3 Minimized Overbookings and Time-Dependent Bid Prices (MOTD)

In contrast to the ROTD model formulation, let us penalize the overbookings into the objective function instead of imposing the constraints (12c). To this aim we associate a penalty cost λ_i to each unit of resource i that is overbooked. The total future overbookings for scenario s are represented by the vector $\mathbf{w}^{(s)} = (w_1^{(s)}, \dots, w_m^{(s)})^\top$, where $w_i^{(s)}$ represents the overbookings of resource i for periods t to T in scenario s. These overbookings are assigned only in the scenarios where they are needed. This results in the following *Minimized Overbookings and Time-Dependent Bid Prices* (MOTD) model formulation (the constraints that depend on τ and/or s hold for $t \le \tau \le T$ and/or $1 \le s \le S$):

$$
\begin{aligned}
V_t^{\text{MOTD}}(\mathbf{x}) = \quad \max \quad & \frac{1}{S}\left[\mathbf{r}^\top \cdot \sum_{s=1}^{S}\sum_{\tau=t}^{T} \mathbf{Y}_\tau^{(s)} - \boldsymbol{\lambda}^\top \cdot \sum_{s=1}^{S} \mathbf{w}^{(s)}\right] && \text{(13a)}\\
\text{s.t.} \quad & \mathbf{A} \cdot \sum_{\tau=t}^{T} \mathbf{Y}_\tau^{(s)} - \mathbf{w}^{(s)} \le \mathbf{x} && \text{(13b)}\\
& \mathbf{Y}_\tau^{(s)} = \mathbf{D}_\tau^{(s)} \cdot (\mathbf{Z}_\tau)^\top && \text{(13c)}\\
& \mathbf{r} = \mathbf{A}^\top \cdot \boldsymbol{\pi}_\tau + \mathbf{U}_\tau - \mathbf{V}_\tau && \text{(13d)}\\
& \mathbf{U}_\tau \le K \cdot \mathbf{Z}_\tau && \text{(13e)}\\
& \mathbf{V}_\tau \le K \cdot (\mathbf{1} - \mathbf{Z}_\tau) && \text{(13f)}\\
& \mathbf{Z}_\tau \cdot \mathbf{V}_\tau \ne \mathbf{0} && \text{(13g)}\\
& \boldsymbol{\pi}_\tau, \mathbf{U}_\tau, \mathbf{V}_\tau \ge \mathbf{0} && \text{(13h)}\\
& \mathbf{Z}^{(s)} \in \{0,1\}^{n\times(T-t)} && \text{(13i)}
\end{aligned}
$$

6 Numerical Experiments

In this section, we illustrate and compare the performances of the different bid-price selection methods discussed earlier. Our aim is to demonstrate the gain in average total revenue with these methods that compute time-dependent bid prices.

6.1 Benchmark methods

We compare the performance of the bid prices against three methods from the literature. The first two methods are DLP and RLP (see Section 3). Both methods compute static bid prices. DLP is a standard technique, whereas RLP is a simulation-based extension for which we used 100 independent demand samples. The third method is based on the Lagrangian Relaxation with Time-Dependent Bid Prices (LRD) model proposed by Kunnumkal and Topaloglu (2010). This technique assumes that the booking horizon is divided in such small periods that at most one customer arrives in a time period. Therefore, we reformulated their solution approach to incorporate the expected demand $\mathbf{D}_t$ for each product during time period t, where each time period has an arbitrary length. We reformulate the original DP formulation of the network RM problem in Kunnumkal and Topaloglu (2010) as

$$
\begin{aligned}
V_t(\mathbf{x}) = \quad \max_{\mathbf{u}_t} \quad & \sum_{j=1}^{n} \mathbb{E}\left[r_j D_{jt} u_{jt} + V_{t+1}\left(\mathbf{x} - D_{jt} u_{jt} \sum_{i=1}^{m} a_{ij} \mathbf{e}_i\right)\right] \\
\text{s.t.} \quad & \sum_{j \in \mathbf{A}_i} D_{jt} u_{jt} \le x_i \qquad 1 \le i \le m && \text{(14)}\\
& u_{jt} \in \{0,1\} \qquad\quad 1 \le j \le n,
\end{aligned}
$$

for $t = 1, \dots, T-1$, where $\mathbf{e}_i$ is the ith unit vector and the decision variable $u_{jt} = 1$ if product j is open for bookings in period t, $u_{jt} = 0$ otherwise. This problem translates to the following formulation with non-negative Lagrange multipliers α_{ijt} associated with the capacity constraints in (14):

$$\begin{aligned} V_t^{\alpha}(\mathbf{x}) = \quad & \max_{\mathbf{u}_t} \quad \sum_{j=1}^{n} \Big\{ u_{jt}\big[r_j - \sum_{i=1}^{n} a_{ij}\alpha_{ijt}\big] + \sum_{i=1}^{m} \alpha_{ijt}x_{it} + V_{t+1}^{\alpha}(\mathbf{x} - u_{jt}\sum_{i=1}^{m} a_{ij}\mathbf{e}_i)\Big\} \\ & \text{s.t.} \quad 0 \le u_{jt} \le d_{jt} \qquad\qquad 1 \le j \le n. \end{aligned} \tag{15}$$

Similar to Kunnumkal and Topaloglu (2010), we compute the bid prices as

$$\pi_{i,t}^{\alpha} = \begin{cases} \sum_{j=1}^{n} d_{j,t+1}\alpha_{ij,t+1} + \ldots + \sum_{j=1}^{n} d_{j,T}\alpha_{ijT} & \text{for } 1 \le t < T \\ 0 & \text{for } t = T, \end{cases} \tag{16}$$

and we use the same LP formulation to set the Lagrange multipliers with $\min_{\alpha \ge 0}\{V_1^{\alpha}(\mathbf{c})\}$, where we replace p_{jt} by the expected demand d_{jt}. We do not have to compare to the technique proposed by Adelman (2007) since the latter provides the same objective value as that provided by the LRD model of Kunnumkal and Topaloglu (2010) (see Tong and Topaloglu (2011) for more details).

6.2 Selected Examples and Numerical Setup

Our test examples are similar to those used by Adelman (2007) and Kunnumkal and Topaloglu (2010). The network includes one hub and L spokes, where $L \in \{4, 8\}$; see Figure 3. One resource (flight) is associated with each spoke-hub pair in either direction. There is a high-fare and a low-fare product for every origin-destination pair (each ordered pair of nodes). This results in $2L$ resources and $2L(L+1)$ products, which require either one or two resources. The price for a high-fare product is five times the price of a low-fare product. The price for each low-fare product is drawn from a discrete uniform distribution over the interval $[14, 49]$, as in Adelman (2007).

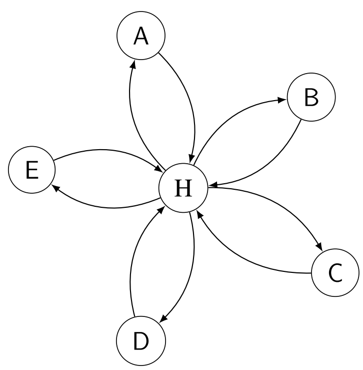

Figure 3: An example network with $L = 5$ spoke nodes and one hub **H**.

We divide the booking horizon into T periods, where $T \in \{5, 10, 20\}$. The expected total number of requests over the booking horizon is set equal to a constant F, selected as follows. For each origin-destination pair, we split the expected demand such that 25% corresponds to the high-fare product, and 75% to the low-fare product. Throughout, we assume that the demand D_{jt} for product j in period $t \in \{1, \ldots, T\}$ is distributed according to a binomial, where n_{jt} is proportional to the expected demand in period t and $\sum_j \sum_t n_{jt} = F$. We use a binomial distribution since this resembles the settings in the literature, where at most one customer requests a certain product in a single time period according to a Bernoulli distribution. In our setting, we have multiple customer arrivals, which corresponds to the sum of Bernoulli trials, i.e., a binomial distribution. We also assume that the actual arrival times at period t are independent and uniformly distributed. We consider two demand settings: (i) the demand distribution for each product is time-stationary, and (ii) the expected demand for high-fare products increases over time, whereas the expected demand for low-fare products decreases over time. More specifically, if μ_j denotes the expected demand for product j over the entire booking horizon and μ_{jt} in period t, then in (i) we take $\mu_{jt} = \mu_j/T$, while in (ii) we take $\mu_{jt} = \mu_j t/(T(T+1)/2)$ when j is a high-fare product and $\mu_{jt} = \mu_j(T-t+1)/(T(T+1)/2)$ when j is a low-fare product.

The initial capacity for each resource is set to a common constant $c_i = C$, where $C \in \{16, 80, 160\}$ for $L = 4$ and $C \in \{8, 40, 80\}$ for $L = 8$. We measure the busyness of each resource by

$$\theta = \frac{\sum_t \sum_j \sum_i a_{ij}\mu_{jt}}{\sum_i c_i},$$

and we set F such that $\theta \in [1, 1.5]$. In our examples, this gives $\theta = F/(C(L+1))$.

For the scenario-based stochastic programming methods proposed in Section 5, we set the number S of demand scenarios to a value selected in $\{1, 5, 10\}$. The choice of $S = 1$ means that each demand D_{jt} is replaced by its expectation. If $S > 1$, we sample S demand scenarios from the corresponding demand distribution. When using a closed-loop control, we resample the demand independently each time that we re-solve an LP at the beginning of a new time period t.

For the MOTD method, we used $\boldsymbol{\lambda}$ equal to the bid prices based on DLP. Using the bid prices of the DLP model is the simplest approach, and also the most logical in our opinion since it represents an approximation for the value of a resource. In practice, the actual overbooking costs should be used as values for $\boldsymbol{\lambda}$. By using the bid prices based on DLP as penalty costs, we introduce a bias towards overbookings in MOTD since the price of the product exceeds the corresponding penalty costs. For ROTD, we took $\alpha \in \{0.5, 0.7, 0.9\}$ and the overbooking bound $\mathbf{b}$ equal to 10% of the capacity $\mathbf{x}$ rounded up to the nearest integer. We also performed numerical experiments for other values of $\boldsymbol{\lambda}$ and $\mathbf{b}$, but they did not perform as well. Note that we used rather low values for α since the number of demand scenarios S is restricted to at most 10 scenarios due to the excessive computational effort (see Section 6.3). Especially when $S = 1$, ROTD is insensitive for any value of α and, therefore, this approach does not use the concept of VaR or CVaR when there is only one demand scenario taken in consideration. However, we include these results for completeness.

For each setting (each choice of L, C, T and demand setting) we randomly generate 10 problem instances where the price vector $\mathbf{r}$ of the products is different. For each problem instance, we first set the bid prices via the different optimization techniques at the beginning of the first time period $t = 1$. Next, we evaluate the performance over 100 simulation runs, where we use the same sequence of customer demands across different policies. We also analyze the performance when the optimization techniques are used to re-optimize the bid prices at the beginning of each time period t, for $t = 1, \ldots, T$, during the booking horizon (the closed-loop control).

Our measure of performance is the average relative gain in revenue obtained by using the considered optimization method compared with DLP. We denote by G the empirical value of this relative gain:

$$G = \frac{1}{10} \sum_{l=1}^{10} \sum_{k=1}^{100} \frac{R_{k,l} - R_{k,l}^{\mathrm{DLP}}}{R_{k,l}^{\mathrm{DLP}}},$$

where $R_{k,l}$ and $R_{k,l}^{\mathrm{DLP}}$ denote the total observed revenue in simulation k for problem instance l with the optimization technique under study and with DLP, respectively.

The computations have been performed on a 2.0Ghz AMD Opteron 246 processor running Linux and Java. We used the RMSIM library presented in Bijvank et al. (2011) to perform the simulations and CPLEX to solve the LP problems.

6.3 Numerical Results

Table 1 summarizes our results for $L = 4$ when the bid prices are computed only once, for $t = 1$ (the open-loop control). We observe that RLP performs well when the busyness of the resources is high and the capacity is large. In such settings, it is easier to allocate and reserve capacities, since uncertainty about the arrival process is lower. The Lagrange relaxation method of Kunnumkal and Topaloglu (2010), adapted to include arbitrary time periods between bid price updates, has the worst performance and does not improve over the static DLP approach in most instances, in contrast with our TDED model, where improvements up to 12% have occured, especially in the case of non-stationary demand. The bid prices based on our VOTD model with only one demand scenario (equal to the average demand) perform almost as well as when they are based on the TDED model. Including multiple demand scenarios is especially beneficial when demand is non-stationary. In general, we find that MOTD does not improve over TDED, and ROTD does not perform as well as VOTD. For the scenario-based stochastic programming methods, a large S (several scenarios) provides better (more stable) results, as expected. This behaviour shows up quite clearly for ROTD.

In summary, the numerical results illustrate that the bid prices based on our TDED, VOTD and ROTD models perform the best. The advantages of TDED and VOTD are attractive in the sense that no model parameters have to be set (such as α and the bound $\mathbf{b}$ in ROTD and the penalty costs in MOTD). However, ROTD and MOTD provide an advanced framework for dealing with overbookings. In such settings, the parameters in ROTD and MOTD reflect the decision maker's attitude towards overbookings. When we contrast the performance of TDED against VOTD, where we only consider one demand scenario equal to the expected demand (i.e., $S = 1$), the results show that TDED outperforms VOTD. However, when randomness is included within the simulated demand scenarios, VOTD can outperform TDED even when the number of scenarios is small. Including scenarios in VOTD or ROTD actually comes at an extra computational effort (see Table 2). In general, we conclude that the extra computational effort is significantly more and requires extra attention to improve the expected revenue in practical settings. Although, in order to reduce the CPU times, scenario reduction algorithms could be applied (see e.g. Heitsch and Römisch (2009)). This topic is out of the scope of the present paper. Similar conclusions can be drawn when we consider the 8-spoke network. Table 3 presents the relative revenue gain G and the CPU times for all models, when a single scenario based on the average demand is considered. It is interesting to note that the CPU times of the TDED model increase when demand is non-stationary, whereas they remain stable for the VOTD model, regardless of the type of demand setting. For VOTD at most one out of the ten test instances for a certain setting takes longer and therefore increases the average CPU time, which is not the case of TDED (where multiple instances take longer). Our intuitive explanation for this phenomenon has to deal with the fact that the variables $\overline{w}_{it}^{(s)}$ provide a certain structure to the feasible solutions; first these variables have a zero value when enough capacity is available, and for some time period t these variables become one. As a result, all allocations $y_{jt}^{(s)}$ become zero when the corresponding values for $\overline{w}_{it}^{(s)}$ equal one, which also specifies that the sum of the bid prices involved for product j should be larger than the revenue r_j.

We also considered a closed-loop control strategy where new bid prices were determined by re-solving an LP at the beginning of each period t, for $t = 1, \ldots, T$. In order to limit CPU time, we limited ourselves to a single demand scenario equal to the average demand (similar to Table 3). Table 4 presents the relative improvements of the revenue over the static DLP model without re-solving. From these results, we observe that re-solving the RLP (and DLP) problems always results in improvements, but does not perform as well as the other methods. The adaptation of the LRD model by Kunnumkal and Topaloglu (2010) only performs well when few customers arrive within a time period during which bid prices are fixed (i.e., when T is large and F is small). This is to be expected, since such settings are consistent with the assumptions underlying the study of Kunnumkal and Topaloglu (2010). Even though the LRD model yields much improved solutions when it is re-solved throughout the booking horizon, it can still perform worse than the static bid prices based on DLP without re-solving. The results obtained by the TDED and VOTD models are the best, similar to those displayed in Table 1 and Table 3. Due to the limited number of scenarios (one, actually), ROTD cannot accommodate overbookings, and therefore does not perform as well as the other techniques. Even with a single scenario, our MOTD model performs reasonably well when it is re-solved. Indeed, it performs on average better than RLP when both models are re-solved, contrary to the results obtained without re-solving (see Tables 1 and 3). Especially since the CPU times for MOTD are always small (within 4 seconds), this approach is very appealing for practical purposes. Therefore, we propose to use either MOTD (with or without multiple demand scenarios) or a best-of-two approach where first VOTD or TDED is used with one

Table 1: The relative revenue gain G (in %) of RLP, LRD, TDED, VOTD, ROTD and MOTD over DLP for different settings in the hub-and-spoke network with $L = 4$. The values of S are given in parentheses.

L	C	F	θ	T	RLP	LRD	TDED	VOTD (S) (1)	VOTD (5)	VOTD (10)	ROTD (S) (1)	ROTD (5) $\alpha = 0.5$	ROTD (5) $\alpha = 0.7$	ROTD (5) $\alpha = 0.9$	ROTD (10) $\alpha = 0.5$	ROTD (10) $\alpha = 0.7$	ROTD (10) $\alpha = 0.9$	MOTD (S) (1)	MOTD (5)	MOTD (10)
stationary demand																				
4	16	88	1.10	5	0.36	0.00	3.41	3.26	3.02	3.33	0.78	2.79	2.65	2.24	2.73	2.64	2.66	2.17	1.82	1.78
4	16	88	1.10	10	0.30	0.00	2.97	3.30	-0.01	3.30	0.52	1.72	1.09	1.26	2.82	3.15	2.50	2.06	-0.94	2.23
4	16	88	1.10	20	0.37	-0.09	3.08	3.39	-10.82	0.50	0.53	-5.61	-9.27	-9.89	1.98	1.81	1.72	1.76	-6.87	1.30
4	80	440	1.10	5	0.08	0.00	3.55	3.15	3.32	3.15	-0.09	2.87	2.62	2.35	2.98	2.35	2.70	2.54	1.40	1.63
4	80	440	1.10	10	0.18	0.12	3.56	3.30	3.21	3.46	-0.06	2.56	2.94	3.04	2.91	3.12	3.04	1.13	2.12	1.58
4	80	440	1.10	20	0.00	-0.12	3.90	3.64	3.30	3.73	-0.01	2.88	2.98	3.12	3.27	3.25	3.51	2.54	2.36	2.12
4	160	880	1.10	5	0.05	0.00	3.80	3.01	2.71	2.83	-0.30	2.70	2.16	2.62	2.59	2.85	3.06	2.70	1.64	1.19
4	160	880	1.10	10	0.05	0.00	3.79	3.22	3.45	3.68	-0.30	2.84	2.81	3.08	3.13	2.99	3.03	1.38	2.01	1.52
4	160	880	1.10	20	0.00	0.06	4.03	3.67	3.65	3.82	-0.27	2.40	3.03	3.12	3.06	3.28	3.46	2.37	2.77	1.73
4	16	100	1.25	5	-0.62	-4.75	2.99	2.65	2.40	2.71	0.49	1.59	1.94	1.49	2.26	2.32	1.72	2.68	0.68	-0.09
4	16	100	1.25	10	-0.80	-4.82	2.70	2.78	1.02	2.92	0.56	1.18	0.76	-0.44	2.42	1.40	2.15	1.27	-0.01	0.46
4	16	100	1.25	20	-0.76	-0.20	3.27	3.13	-7.94	2.07	0.55	-3.42	-6.27	-7.52	1.96	1.64	1.28	2.06	-6.70	1.31
4	80	500	1.25	5	1.44	-4.72	4.03	3.32	3.54	3.60	2.36	2.82	3.07	2.67	2.88	3.08	2.86	2.95	0.46	0.24
4	80	500	1.25	10	1.14	-4.55	3.89	3.68	3.28	4.05	2.11	2.79	2.59	2.69	3.00	3.13	3.01	2.65	1.57	0.61
4	80	500	1.25	20	1.37	-4.81	4.38	4.13	3.93	4.19	1.97	2.93	2.64	3.11	3.27	3.70	3.26	3.15	2.66	1.80
4	160	1000	1.25	5	3.01	-5.18	4.53	4.04	3.57	3.67	2.68	3.05	3.02	2.91	3.37	3.49	2.83	3.27	1.81	1.67
4	160	1000	1.25	10	3.04	-5.10	4.48	4.21	4.09	4.50	2.42	2.96	3.05	2.98	3.46	3.37	3.34	2.93	1.82	1.33
4	160	1000	1.25	20	3.13	-5.50	4.72	4.60	4.30	4.41	1.75	3.19	2.88	3.26	3.35	3.61	3.49	4.06	3.05	2.31
non-stationary demand																				
4	16	88	1.10	5	0.57	0.00	8.45	8.28	9.80	10.34	2.59	7.33	8.29	8.83	8.37	9.07	10.04	7.00	2.96	4.16
4	16	88	1.10	10	0.63	0.05	7.72	7.95	7.34	11.12	2.21	7.48	6.41	6.64	9.94	10.36	9.40	6.38	4.29	4.83
4	16	88	1.10	20	0.77	0.04	9.21	9.90	-5.18	10.45	2.50	0.22	-2.04	-2.93	7.50	8.40	8.60	6.50	-0.51	5.30
4	80	440	1.10	5	0.22	0.00	9.72	9.52	9.69	10.58	1.61	7.16	8.31	8.47	8.59	9.33	9.69	7.00	3.85	3.47
4	80	440	1.10	10	0.22	1.14	10.55	10.71	11.25	12.08	1.75	8.56	8.95	9.91	8.94	9.96	10.71	7.56	6.71	5.25
4	80	440	1.10	20	0.65	0.00	11.17	11.60	12.34	13.12	2.07	9.56	10.03	10.47	10.48	10.58	11.58	9.26	7.15	6.71
4	160	880	1.10	5	0.18	0.00	10.40	10.11	10.12	10.81	1.27	7.15	7.82	8.56	7.83	8.62	9.26	7.33	3.84	3.36
4	160	880	1.10	10	0.21	1.01	11.22	11.39	11.59	12.26	1.42	8.37	8.60	9.95	7.93	9.56	10.85	7.98	6.97	3.63
4	160	880	1.10	20	0.58	0.08	12.15	12.01	12.71	13.38	1.65	8.44	9.51	10.44	9.69	10.67	11.12	9.44	7.43	6.18
4	16	100	1.25	5	-2.13	-7.17	6.31	4.94	8.48	9.36	0.88	4.88	5.05	6.73	6.88	6.95	7.91	5.36	1.52	0.70
4	16	100	1.25	10	-2.04	-0.60	6.41	6.49	6.35	9.38	1.99	5.11	4.88	5.50	7.09	8.09	7.15	3.13	0.68	0.15
4	16	100	1.25	20	-1.72	0.91	7.63	7.78	-4.46	8.65	1.46	1.50	-0.16	-2.15	7.59	6.97	7.98	4.24	-1.01	3.21
4	80	500	1.25	5	2.30	-10.27	8.26	7.99	9.09	10.00	3.15	5.90	6.97	6.78	7.66	8.32	8.26	7.31	2.73	0.37
4	80	500	1.25	10	3.00	-13.10	9.42	9.34	10.26	11.42	3.99	5.86	7.08	7.94	8.22	7.89	9.62	3.97	3.54	2.03
4	80	500	1.25	20	2.71	-13.66	10.43	10.18	11.09	11.65	3.95	7.49	7.35	7.99	6.89	8.06	8.91	6.38	5.25	4.17
4	160	1000	1.25	5	6.00	-11.16	9.53	9.05	10.02	10.57	4.01	6.29	7.26	6.93	7.38	8.10	9.38	8.11	4.68	1.59
4	160	1000	1.25	10	6.47	-14.23	10.42	10.56	11.21	12.01	3.74	6.22	7.28	8.41	7.83	8.82	9.44	4.20	4.16	4.94
4	160	1000	1.25	20	6.82	-14.97	11.57	11.69	11.78	13.11	4.21	6.55	6.12	7.74	7.88	8.42	8.02	7.81	5.56	5.47

Table 2: The CPU time (sec) of DLP, RLP, LRD, TDED, VOTD, ROTD and MOTD for different settings in the hub-and-spoke network with $L = 4$.

L	C	F	θ	T	DLP	RLP	LRD	TDED	VOTD (S) (1)	VOTD (S) (5)	VOTD (S) (10)	ROTD (S) (1)	ROTD (S) (5) $\alpha=0.5$	ROTD (S) (5) $\alpha=0.7$	ROTD (S) (5) $\alpha=0.9$	ROTD (S) (10) $\alpha=0.5$	ROTD (S) (10) $\alpha=0.7$	ROTD (S) (10) $\alpha=0.9$	MOTD (S) (1)	MOTD (S) (5)	MOTD (S) (10)
stationary demand																					
4	16	88	1.10	5	0.02	4.35	0.02	0.07	0.12	1.03	2.63	0.06	0.51	0.30	0.28	0.90	1.72	0.65	0.07	0.16	0.16
4	16	88	1.10	10	0.02	4.67	0.04	0.14	0.33	1.08	5.54	0.19	1.94	0.69	0.46	1.73	1.56	1.15	0.15	0.38	0.50
4	16	88	1.10	20	0.02	5.21	0.15	0.38	1.26	3.35	8.10	0.26	13.79	1.83	1.66	4.02	3.03	2.11	0.41	1.19	1.90
4	80	440	1.10	5	0.02	5.03	0.02	0.08	0.18	1.27	9.20	0.04	0.57	0.76	0.81	0.87	1.23	1.83	0.07	0.10	0.17
4	80	440	1.10	10	0.02	6.00	0.07	0.15	0.39	3.50	44.92	0.12	2.84	2.18	5.05	6.11	3.82	13.99	0.20	0.29	0.52
4	80	440	1.10	20	0.03	6.04	0.15	0.32	1.02	14.90	158.22	0.33	6.23	3.06	2.45	32.30	19.48	20.08	0.46	1.06	1.65
4	160	880	1.10	5	0.02	6.70	0.03	0.08	0.12	0.93	3.88	0.06	0.45	0.33	0.76	0.58	0.75	1.28	0.08	0.12	0.19
4	160	880	1.10	10	0.02	6.59	0.04	0.10	0.33	4.95	56.28	0.10	2.50	1.40	9.79	3.26	5.33	5.86	0.19	0.25	0.42
4	160	880	1.10	20	0.01	7.53	0.18	0.38	1.04	29.35	351.97	0.28	6.08	9.26	5.14	27.58	135.19	25.26	0.48	0.99	1.75
4	16	100	1.25	5	0.02	3.43	0.02	0.13	0.09	0.76	13.87	0.05	2.18	1.01	0.43	5.95	10.09	1.92	0.05	0.13	0.17
4	16	100	1.25	10	0.01	5.01	0.06	0.22	0.38	2.31	24.29	0.20	2.17	1.62	0.71	11.87	8.74	3.77	0.23	0.39	0.51
4	16	100	1.25	20	0.02	5.78	0.19	1.24	1.24	3.77	16.04	0.49	9.03	2.13	1.35	21.50	7.50	3.08	0.48	1.39	1.95
4	80	500	1.25	5	0.01	4.41	0.02	0.11	0.10	2.01	10.38	0.05	3.32	4.91	2.19	6.62	8.44	10.37	0.05	0.14	0.21
4	80	500	1.25	10	0.02	5.70	0.06	0.19	0.31	8.66	275.92	0.13	203.86	17.05	30.29	382.63	506.92	283.07	0.16	0.36	0.53
4	80	500	1.25	20	0.01	6.67	0.17	0.75	0.71	22.39	854.07	0.32	121.92	10.64	9.68	877.10	849.91	598.02	0.30	1.08	1.63
4	160	1000	1.25	5	0.02	7.01	0.03	0.13	0.08	1.56	3.68	0.06	2.48	3.40	3.68	4.19	6.71	13.51	0.05	0.17	0.24
4	160	1000	1.25	10	0.03	7.23	0.07	0.16	0.33	8.28	115.66	0.15	327.75	232.42	241.24	528.95	481.46	385.46	0.19	0.40	0.62
4	160	1000	1.25	20	0.03	8.22	0.20	0.98	0.78	36.08	913.57	0.35	704.98	371.20	279.01	947.08	1132.53	1017.55	0.33	1.15	1.91
non-stationary demand																					
4	16	88	1.10	5	0.02	4.28	0.02	1.42	0.16	1.04	11.07	0.06	0.44	0.36	0.29	1.07	1.14	0.68	0.11	0.19	0.20
4	16	88	1.10	10	0.02	4.83	0.07	0.79	0.53	2.04	11.13	0.12	1.67	0.60	1.07	2.23	2.11	1.15	0.17	0.45	0.67
4	16	88	1.10	20	0.02	5.51	0.20	0.91	1.60	6.91	12.85	0.34	3.74	2.20	1.34	3.51	3.54	1.95	0.40	1.27	2.30
4	80	440	1.10	5	0.02	5.25	0.03	0.68	0.19	3.62	25.98	0.07	0.61	0.87	0.62	1.81	1.85	1.97	0.09	0.15	0.19
4	80	440	1.10	10	0.02	6.11	0.07	0.52	0.56	10.99	108.40	0.10	4.88	2.38	1.42	3.90	4.52	4.45	0.16	0.32	0.50
4	80	440	1.10	20	0.03	6.36	0.17	0.78	1.39	21.14	376.67	0.33	5.72	4.20	1.93	9.57	9.26	8.88	0.48	1.17	1.90
4	160	880	1.10	5	0.02	6.09	0.02	1.09	0.22	2.62	23.54	0.06	0.58	0.91	0.75	1.19	1.29	1.26	0.08	0.11	0.16
4	160	880	1.10	10	0.02	7.24	0.06	0.60	0.49	12.39	127.99	0.12	3.82	2.38	1.98	5.34	6.65	5.70	0.26	0.30	0.48
4	160	880	1.10	20	0.03	7.27	0.14	0.74	1.23	59.62	672.58	0.27	5.43	6.19	3.54	6.88	9.91	11.56	0.50	1.17	1.73
4	16	100	1.25	5	0.02	5.01	0.03	0.12	0.12	3.16	28.95	0.09	1.19	0.47	0.43	2.41	2.56	1.12	0.07	0.16	0.22
4	16	100	1.25	10	0.02	5.19	0.06	0.71	0.72	3.93	29.23	8.38	9.18	1.43	0.65	8.92	9.89	1.75	0.20	0.43	0.55
4	16	100	1.25	20	0.02	5.35	0.18	1.20	3.17	3.93	23.14	0.75	6.32	3.50	1.77	13.93	8.15	4.96	0.42	1.43	2.00
4	80	500	1.25	5	0.02	5.75	0.03	0.10	0.15	5.70	47.21	0.31	2.83	3.69	1.91	9.98	24.25	4.76	0.05	0.15	0.22
4	80	500	1.25	10	0.01	5.98	0.05	0.53	0.90	24.13	500.93	71.06	46.06	17.43	4.62	218.76	110.48	38.87	0.16	0.38	0.61
4	80	500	1.25	20	0.03	6.45	0.19	0.86	1.71	48.33	971.60	0.96	40.39	16.46	12.37	454.31	344.37	141.78	0.42	1.19	1.87
4	160	1000	1.25	5	0.02	7.07	0.04	0.09	0.11	3.95	33.80	0.32	4.28	5.62	3.49	15.53	35.64	5.46	0.06	0.15	0.24
4	160	1000	1.25	10	0.02	7.26	0.08	0.68	0.78	26.20	355.51	9.14	109.00	54.35	43.32	235.22	331.26	104.25	0.17	0.40	0.59
4	160	1000	1.25	20	0.02	8.33	0.18	0.83	1.97	370.26	1123.73	0.93	233.86	84.52	62.02	630.62	677.24	692.64	0.43	1.24	1.91

Table 3: The relative revenue gain (%) of RLP, LRD, TDED, VOTD, ROTD and MOTD over DLP and the CPU times (sec) for different settings in the hub-and-spoke network with $L = 8$.

					relative revenue gain (%)						CPU times (sec)						
L	C	F	θ	T	RLP	LRD	TDED	VOTD	ROTD	MOTD	DLP	RLP	LRD	TDED	VOTD	ROTD	MOTD
stationary demand																	
8	8	88	1.22	5	1.45	0.41	4.60	4.24	1.84	1.18	0.03	7.38	0.17	0.61	1.91	0.66	0.39
8	8	88	1.22	10	1.28	3.29	4.07	3.99	1.72	1.76	0.03	9.25	0.49	2.14	12.58	2.80	1.21
8	8	88	1.22	20	1.66	3.52	4.34	4.08	1.24	1.29	0.03	11.79	1.65	8.42	167.42	8.09	4.32
8	40	440	1.22	5	2.58	-1.37	5.72	5.19	3.17	3.40	0.02	10.10	0.24	0.54	1.64	0.71	0.76
8	40	440	1.22	10	2.69	1.02	5.71	5.40	2.76	3.32	0.03	11.38	0.50	1.74	4.69	1.64	1.17
8	40	440	1.22	20	2.64	1.21	5.48	5.36	3.02	2.52	0.03	14.06	2.09	5.56	50.60	6.72	3.30
8	80	880	1.22	5	3.88	-3.23	6.46	5.90	3.62	4.41	0.04	10.50	0.17	0.38	1.53	0.73	0.66
8	80	880	1.22	10	3.72	1.16	6.22	5.98	3.27	4.51	0.03	11.61	0.51	1.31	3.96	1.42	1.03
8	80	880	1.22	20	3.78	1.23	6.13	5.95	3.34	4.16	0.03	13.91	1.52	4.41	33.00	4.94	3.09
8	8	100	1.39	5	2.01	0.37	5.46	5.27	2.92	1.99	0.04	8.36	0.19	0.59	1.47	3.18	1.76
8	8	100	1.39	10	1.77	4.69	5.44	5.39	2.54	3.25	0.04	8.89	0.53	2.04	12.57	18.12	1.33
8	8	100	1.39	20	1.72	4.52	5.19	4.97	2.21	2.04	0.03	11.36	1.74	6.82	174.24	8.32	3.29
8	40	500	1.39	5	5.05	-3.88	7.20	6.69	5.11	4.42	0.05	9.83	0.16	0.44	1.14	2.49	0.55
8	40	500	1.39	10	5.20	0.52	7.16	6.94	4.54	4.02	0.04	10.56	0.50	1.71	4.93	39.01	1.26
8	40	500	1.39	20	5.29	0.86	7.13	6.94	4.28	3.48	0.04	12.60	1.65	6.54	42.48	5.08	2.96
8	80	1000	1.39	5	6.53	-4.35	7.98	7.29	5.91	5.28	0.05	11.74	0.18	0.35	1.18	2.41	0.55
8	80	1000	1.39	10	6.65	0.82	7.95	7.66	5.16	5.18	0.04	12.24	0.56	1.34	4.13	4.88	1.53
8	80	1000	1.39	20	6.41	1.33	7.89	7.78	4.91	5.17	0.05	14.69	1.59	5.59	22.99	4.58	3.56
non-stationary demand																	
8	8	88	1.22	5	2.89	0.63	9.32	8.69	3.70	3.28	0.04	8.62	0.22	144.90	1.90	1.25	0.48
8	8	88	1.22	10	2.82	6.82	9.64	9.40	3.18	3.55	0.03	8.90	0.49	13.74	17.79	21.31	1.13
8	8	88	1.22	20	3.26	6.80	9.94	9.62	2.60	3.14	0.03	11.42	1.52	764.05	156.07	764.65	3.12
8	40	440	1.22	5	5.42	-5.62	12.46	12.22	5.18	8.77	0.04	10.46	0.22	90.86	1.28	0.76	0.49
8	40	440	1.22	10	5.65	0.25	13.76	13.39	5.96	9.25	0.03	10.48	0.49	7.29	7.07	7.28	1.10
8	40	440	1.22	20	5.98	0.82	14.12	13.85	6.63	6.79	0.03	13.32	1.99	350.89	51.55	22.38	3.44
8	80	880	1.22	5	8.27	-6.35	14.59	13.81	5.99	9.38	0.04	10.34	0.19	62.63	1.26	0.81	0.41
8	80	880	1.22	10	8.53	0.32	15.42	15.23	7.38	9.64	0.03	11.98	0.53	6.88	5.07	13.51	1.03
8	80	880	1.22	20	9.02	1.82	16.00	15.69	7.81	10.79	0.03	13.18	1.95	41.78	35.30	13.54	3.72
8	8	100	1.39	5	3.40	0.93	11.30	10.74	4.15	4.28	0.04	8.34	0.18	0.69	1.32	15.92	0.38
8	8	100	1.39	10	3.52	9.82	11.97	11.90	5.07	5.58	0.03	8.47	0.55	763.62	6.81	120.12	1.37
8	8	100	1.39	20	4.05	9.04	11.94	12.10	4.96	4.31	0.03	9.53	1.42	885.48	142.55	53.24	3.36
8	40	500	1.39	5	10.17	-7.97	15.46	15.09	8.91	11.31	0.04	9.72	0.17	0.65	1.39	293.49	0.38
8	40	500	1.39	10	11.39	0.51	16.42	16.32	9.24	10.98	0.03	9.63	0.46	538.71	4.40	1087.01	1.22
8	40	500	1.39	20	11.78	0.94	17.55	17.31	9.56	9.65	0.04	12.37	1.57	620.16	31.87	13.89	3.33
8	80	1000	1.39	5	13.99	-9.25	17.13	16.11	10.66	12.68	0.04	11.14	0.20	0.57	1.05	183.31	0.56
8	80	1000	1.39	10	15.30	0.61	18.71	18.77	10.74	13.28	0.03	11.69	0.62	575.46	4.67	1085.63	1.27
8	80	1000	1.39	20	15.72	0.28	19.70	19.12	10.79	15.81	0.02	12.67	1.93	109.17	19.58	9.32	3.48

demand scenario and if the computation time takes longer than some threshold value the bid prices should be based on MOTD.

7 Conclusion

In the present work, we have considered numerical approaches for addressing resource allocation in network revenue management where product availability is adjusted over time with time-dependent bid prices. Based on the simple DLP method, new formulations have been proposed. These involve additional variables such that the bid prices are computed directly without resorting to duality (as in DLP). First, we have used deterministic but time-varying demand and next we have developed scenario-based stochastic programming formulations, each one dealing with overbookings in its own way. In our numerical experiments, the method with deterministic demand set to its expectation (TDED) outperforms the existing techniques from the literature such as DLP, RLP and LRD by 2-3% on average when bid prices are re-optimized over the booking horizon. When the bid prices are not re-optimized, this is even 4-6% on average. Especially when the demand is non-stationary, our method performs very well. Similar conclusions are found for the scenario-based stochastic programming methods, where performance can be improved even further when additional demand scenarios are considered, albeit this comes at an extra computational effort. The approach where virtual overbookings are included performs very well when there is only one demand scenario that is set to the expected demand. ROTD cannot address overbookings correctly when a single demand scenario is taken into consideration. ROTD and MOTD provide advanced frameworks for dealing with overbookings, since the booking control policy is only based on the time-dependent bid prices and not on real-time information

Table 4: The relative revenue gain (%) of re-solving DLP, RLP, LRD, TDED, VOTD, ROTD and MOTD over static DLP for different settings in the hub-and-spoke network with $L = 4, 8$.

					stationary demand							non-stationary demand						
L	C	F	θ	T	DLP	RLP	LRD	TDED	VOTD	ROTD	MOTD	DLP	RLP	LRD	TDED	VOTD	ROTD	MOTD
4	16	88	1.10	5	3.12	3.48	2.23	6.59	6.16	4.24	5.47	9.71	12.15	7.29	13.88	13.77	8.18	12.61
4	16	88	1.10	10	3.87	5.29	5.48	6.99	6.80	4.16	6.22	12.13	17.15	12.72	15.53	15.79	8.57	14.55
4	16	88	1.10	20	4.69	3.80	7.32	7.24	7.13	4.35	6.38	14.24	15.73	17.15	17.80	17.74	10.21	16.58
4	80	440	1.10	5	3.48	4.97	0.29	5.08	4.73	3.46	4.44	9.86	11.45	2.38	13.21	12.78	7.08	12.18
4	80	440	1.10	10	4.20	5.07	2.41	5.73	5.42	3.46	5.28	12.87	14.50	9.31	15.04	14.95	7.97	14.78
4	80	440	1.10	20	4.73	5.06	4.39	6.03	5.83	3.52	5.72	14.55	15.42	13.45	16.21	16.44	9.33	16.13
4	160	880	1.10	5	3.72	4.92	0.04	4.63	4.35	2.98	4.22	9.94	11.20	1.06	12.91	12.43	6.18	11.95
4	160	880	1.10	10	4.17	5.46	1.54	5.33	5.10	2.99	4.95	12.95	14.42	7.98	14.76	14.59	7.08	14.55
4	160	880	1.10	20	4.69	5.02	3.62	5.65	5.40	3.03	5.44	14.51	15.03	12.25	15.76	15.94	8.11	15.87
4	16	100	1.25	5	1.17	2.84	-0.50	6.50	5.94	3.58	5.00	6.54	10.06	2.37	12.16	11.97	4.51	10.26
4	16	100	1.25	10	2.36	4.19	3.91	6.97	6.83	3.47	5.68	9.95	16.00	10.26	14.38	14.40	6.70	12.77
4	16	100	1.25	20	3.65	6.88	6.63	7.53	7.42	3.80	6.36	12.01	16.17	14.78	15.94	16.12	7.40	14.33
4	80	500	1.25	5	2.67	5.14	-3.47	5.78	5.18	3.87	4.65	8.24	10.99	-4.41	12.83	12.26	5.87	11.46
4	80	500	1.25	10	3.34	5.87	1.03	6.37	6.21	3.49	5.60	11.02	13.80	5.15	14.32	14.59	6.31	13.69
4	80	500	1.25	20	4.44	5.67	4.26	7.00	6.95	3.56	6.46	13.02	15.11	11.28	15.66	15.88	6.89	15.28
4	160	1000	1.25	5	2.99	5.39	-4.60	5.57	5.07	3.65	4.57	8.79	11.23	-6.25	13.11	12.43	5.34	11.69
4	160	1000	1.25	10	3.62	6.25	-0.07	6.26	6.03	3.10	5.51	11.48	14.05	3.83	14.52	14.58	5.33	13.94
4	160	1000	1.25	20	4.49	6.21	3.45	6.71	6.60	2.96	6.17	13.28	15.03	10.48	15.57	15.80	5.88	15.37
8	8	88	1.22	5	2.81	2.91	5.55	8.40	7.94	4.89	6.86	10.94	13.13	11.04	15.49	15.01	8.59	13.10
8	8	88	1.22	10	4.36	4.68	9.21	8.95	8.74	5.18	7.88	13.83	17.01	19.42	17.16	17.93	9.86	15.81
8	8	88	1.22	20	5.77	5.94	10.55	8.02	8.96	5.38	10.01	15.95	19.28	22.12	24.18	19.20	10.62	19.76
8	40	440	1.22	5	5.90	7.61	0.67	8.64	8.12	5.45	8.04	14.06	16.37	4.14	17.87	17.67	10.31	17.36
8	40	440	1.22	10	6.90	7.38	5.65	9.39	9.07	5.64	8.83	17.56	19.61	14.48	20.45	20.54	11.57	19.86
8	40	440	1.22	20	7.58	8.26	8.35	9.69	9.57	5.69	9.18	19.96	21.02	19.95	22.05	22.17	12.88	21.41
8	80	880	1.22	5	6.77	8.06	-1.15	8.73	8.19	5.15	8.03	15.19	17.12	1.88	18.93	18.75	10.03	18.30
8	80	880	1.22	10	7.34	8.49	4.38	9.14	8.91	5.06	8.81	18.73	20.43	13.43	21.35	21.44	11.05	20.95
8	80	880	1.22	20	7.85	8.61	7.36	9.42	9.30	5.02	9.10	20.53	21.80	19.26	22.33	22.41	11.78	21.95
8	8	100	1.39	5	2.56	2.69	5.09	9.61	8.89	5.60	7.40	10.76	13.64	10.81	17.24	16.81	9.07	13.86
8	8	100	1.39	10	4.67	5.44	9.77	10.07	9.75	5.52	8.37	14.78	16.94	20.76	19.38	10.76	19.37	14.78
8	8	100	1.39	20	6.00	6.43	11.19	9.90	9.07	5.71	11.69	16.70	19.95	23.40	25.47	20.52	10.16	21.90
8	40	500	1.39	5	6.48	9.17	-0.54	10.53	9.72	6.99	9.32	16.33	19.40	3.86	21.21	20.92	13.10	20.14
8	40	500	1.39	10	7.61	9.05	5.93	11.08	10.74	6.85	10.13	20.15	22.72	16.46	23.57	23.51	13.97	22.79
8	40	500	1.39	20	8.60	9.69	9.31	11.41	11.25	6.78	10.62	22.66	24.10	22.24	25.24	25.19	14.74	24.39
8	80	1000	1.39	5	7.84	9.91	-2.17	10.68	9.99	6.77	9.63	17.71	20.35	1.75	22.26	21.84	12.62	21.35
8	80	1000	1.39	10	8.47	10.36	5.00	11.18	10.80	6.60	10.45	21.78	23.82	15.60	24.73	24.84	13.88	24.33
8	80	1000	1.39	20	9.14	10.55	8.45	11.33	11.16	6.41	10.79	23.87	25.69	21.95	25.98	25.94	14.37	25.51

on resource availability (overbookings are allowed, either constrained or penalized). This may prove to be an important feature in many practical situations. Another advantage of MOTD is that the computation times for this approach are small (within a few seconds). Therefore, this approach is very appealing for practical purposes.

Another approach to compute time-dependent bid prices would be to combine ROTD and MOTD where the CVaR is taken on the overbooking costs instead of the overbookings themselves. Also an interesting topic for further research is the selection of a limited number of demand scenarios. A final direction for future interest is to extend the methods proposed in this paper with customer choice behavior, whereas we assumed independent demand (i.e., a customer requests only one product independently of the availability of other products).

References

Adelman, D. 2007. Dynamic bid prices in revenue management. *Oper. Res.* **55**(4) 647–661.

Akan, M., B. Ata. 2008. On bid-price controls for network revenue management. Working paper, Kellogg School of Management, Northwestern University.

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

Berman, O., M. Hu, Z. Pang. 2010. On the bid price trends in revenue management. Working paper, Rotman School of Management, University of Toronto.

Bertsimas, D., S. de Boer. 2005. Simulation-based booking limits for airline revenue management. *Oper. Res.* **53**(1) 90–106.

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

Bijvank, M., P. L'Ecuyer, P. Marcotte. 2011. RMSIM: A Java library for simulating revenue management systems. *Proceedings of the 2011 Winter Simulation Conference*. IEEE Press, Pistacaway, New-Jersey, 2703–2714.

Chen, L., T. Homem-de Mello. 2010. Re-solving stochastic programming models for airline revenue management. *Annals of Operations Research* **177** 91–114.

Cooper, W.L. 2002. Asumptotic behavior of an allocation policy for revenue management. *Oper. Res.* **50**(4) 720–727.

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

de Boer, S., R. Freling, N. Piersma. 2002. Mathematical programming for network revenue management revisited. *Eur. J. Oper. Res.* **137** 72–92.

DeMiguel, V., N. Mishra. 2008. What multistage stochastic programming can do for network revenue management. Working paper, London Business School.

Farias, V.F., B. van Roy. 2007. An approximate dynamic programming approach to network revenue management. Working paper, Electrical Engineering, Stanford University.

Heitsch, H., W. Römisch. 2009. Scenario tree reduction for multistage stochastic programs. *Computational Management Science* **6**(2).

Higle, J.L. 2007. Bid-price control with origin-destination demand: A stochastic programming approach. *Journal of Revenue and Pricing Management* **5**(4) 291–304.

Jasin, S. 2012. A class of "error-shrinking" heuristics for network revenue management. Working paper, Stephen M. Ross School of Business, University of Michigan.

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

Klein, R. 2007. Network capacity control using self-adjusting bid-prices. *OR Spectrum* **29** 39–60.

Koenig, M., J. Meissner. 2010. Value-at-risk optimal policies for revenue management problems. Working paper, Department of Management Science, Lancaster University Management School.

Kunnumkal, S., H. Topaloglu. 2010. Computing time-dependent bid prices in network revenue management problems. *Transportation Sci.* **44**(1) 38–62.

Möller, A., W. Römisch, K. Weber. 2008. Airline network revenue management by multistage stochastic programming. *Comput Management Science* **5** 355–377.

Reiman, M. I., Q. Wang. 2008. An asymptotically optimal policy for a quantity-based network revenue management problem. *Math. Oper. Res.* **33**(2) 257–282.

Rockafellar, R.T., S. Uryasev. 2000. Optimization of conditional value-at-risk. *Journal of Risk* **2**(3) 21–41.

Rockafellar, R.T., S. Uryasev. 2002. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance* **26**(7) 1443–1471.

Secomandi, N. 2008. An analysis of the control-algorithm re-solving issue in inventory and revenue management. *Manufacturing Service Oper. Management* **10**(3) 468–483.

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

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

Talluri, K., G. van Ryzin. 1999. A randomized linear programming method for computing network bid prices. *Transportation Sci.* **33**(2) 207–216.

Talluri, K., G. van Ryzin. 2004. *The theory and practice of revenue management*. Kluwer Academic Publishers, Norwell, Massachusetts.

Tong, C., H. Topaloglu. 2011. On approximate linear programming approach for network revenue management problems. Working paper, School of Operations Research and Information Engineering, Cornell University.

Topaloglu, H. 2008. A stochastic approximation method to compute bid prices in network revenue management problems. *INFORMS J. Comput.* **20**(4) 596–610.

Topaloglu, H. 2009. On the asymptotic optimality of the randomized linear program for network revenue management. *Eur. J. Oper. Res.* **197** 884–896.

van Ryzin, G., G. Vulcano. 2008. Simulation-based optimization of virtual nesting controls for network revenue management. *Oper. Res.* **56**(4) 865–880.

Williamson, E.L. 1992. Airline network seat control. Ph.D. thesis, MIT, Cambridge, Massachusetts.