Bid Price Control for Demand Fulfillment in a Make-to-Stock Production System

Yao Yang Moritz Fleischmann

Working Paper

1. Introduction

After three decades' development, revenue management has become an active area of research. Its idea is not only applied in the traditional service industries such as airline, hotel and car rental, but also in manufacturing industry. In this paper, we consider the application of revenue management to a demand fulfillment problem of a make-to-stock (MTS) system, where the manufacturer is facing stochastic demand from heterogeneous customers with different unit revenues. We assume that inventory replenishments are scheduled exogenously and deterministic. The manufacturer has to decide for each order whether to satisfy it from stock, backorder it at a penalty cost, or reject it, in anticipation of more profitable future orders, with the objective to maximize the expected profit over a finite planning horizon, taking into account sales revenues, inventory holding costs, and backorder penalties.

The main problem task in our demand fulfillment problem is the same as in the traditional revenue management problems, namely to allocate limited resources to customers with different willingness to pay to maximize revenue or profit. Because our problem considers multiple resources (different replenishments) it is closely linked with the network revenue management (NRM) problem. However, different from the traditional network revenue management problem where each incoming order requests a specific set of resources, we have the flexibility to choose between different supply options. This flexibility links our problem to another emerging research topic in literature, the so-called revenue management problem with flexible products, which can be considered as an extension of the traditional network revenue management problem.

Network revenue management is a very important research stream in revenue management literature as it reflects plenty of problems in reality. Normally, it refers to the decision-making problem of selling products that are composed of a bundle of resources under various terms and conditions, with an aim to maximize revenue (Talluri and van Ryzin, 2004).

In the airline industry, where this class of problem originates, this is mirrored by a network of different flight legs, consisting of a mix of local- and connecting traffic. A product is then an "origin-destination itinerary fare class combination".

In the hotel case, each room-night is a separate resource. When customers stay multiple nights, they are consuming multiple resources and the multi-night stays are analogous to multi-leg itineraries in an airline case.

Unlike single-resource revenue management problem, in the network case, if one of the resources in the bundle faces limitation in its availability, sales of the whole bundle will be constrained. This implies that there are interdependencies between resources and therefore total revenue maximization requires a joint management of all capacity controls across the network (Talluri and van Ryzin, 2004). In literature, the interdependencies between resources are sometimes referred to as *network effects*.

In a make-to-stock (MTS) production system, the resources to sell are the finished goods inventory. In nowadays advanced planning system (APS), the available finished goods inventory is represented by the so-called available-to-promise (ATP) quantities. To satisfy a given order, using ATP quantities from different replenishment batches entails different costs (e.g., inventory holding or backlogging cost). Therefore, the order acceptance in an MTS system also resembles a network revenue management problem, with ATP quantities from different replenishments as different resources. However, different from a traditional network revenue management problem where different resources are complementary to each other due to the network effect, in the MTS case, different ATP supplies are substitutive. This is because, since in an MTS setting all finished goods are physically identical, theoretically any of the available supplies can be used to satisfy any incoming order, and the lack of any specific ATP does not constrain the sales of the others. This flexibility links our problem to the research stream – revenue management with flexible products.

The incorporation of flexible products into capacity control is relatively new in revenue management research. A flexible product is defined as a set of alternative products serving the same market (Gallego and Philips, 2004). Purchasers of flexible products are assigned to one of the alternatives at a later time, normally when most of the demand has been realized and uncertainty is lower. Therefore, in revenue management, flexible products are usually provided as supplementary to the more traditional specific products at a lower price to hedge against demand uncertainty, and they are viewed as inferior to specific products by most customers.

In our MTS setting, ATP quantities from different replenishments can also be treated as different products alternatives, or in other words, flexible products. However, there are several differences between our demand fulfillment problem and revenue management with flexible products. First, in an MTS setting, customers normally do not ask for products from a specific batch. Therefore, we always have the flexibility to choose between different supply alternatives, i.e., we do not have specific products. All products in our problem setting are flexible products. Second, we have to decide the choice of resources in real time and cannot postpone the order promise after collecting most of the demand. Therefore, the risk pooling effect of flexible products does not exist. Third, the decision to be made is more complex. In the traditional network revenue management problem where only specific products are offered, the decision to be made is a simple yes or no: whether or not to accept an incoming request. If flexible products are included, one has to go a step further to decide which alternative to assign to an accepted flexible request. In our case, we have to decide not only which alternative to assign, but also how many of each alternative to use as the order size is normally larger than one in an MTS setting. This process can be considered as repeating the alternative selecting decision multiple times. Forth, in our problem setting, time plays a particular role in defining the multiple resources. Due to inventory holding cost and backlogging cost, the margin of choosing a certain resource changes over time. This makes our system more dynamic than the traditional NRM case (either with or without flexible products).

From a modeling perspective, theoretically, all network revenue management problems can be modeled by a dynamic program (DP) to determine the optimal policy. The problem with the resulting DP is that, due to the high-dimensional state space, solving the DP analytically usually yields models of intractable complexity which is commonly referred to as “Bellman’s curse of dimensionality” (Adelman, 2007). Consequently, approximations have been developed which neglect certain factors or estimate certain inputs to generate tractable and implementable solutions that, despite occasional non-optimality, increase companies’ revenue (Talluri and van Ryzin, 1998).

Among all of the methods, bid-price control is proving to become the dominant method (Talluri and van Ryzin, 2004; Klein and Steinhardt, 2008). For network revenue management problems, bid-price control sets a threshold price (bid price) for each resource in the network and an order for a certain product is only accepted if its revenue exceeds the sum of the bid prices of all required resources. From a dynamic programming perspective, bid-price control actually does not always generate the optimal policy for network revenue management problems due to the nonlinearity of the value function. However, it gains its popularity because of its intuitive nature and implementation simplicity.

With certain assumptions, Quante et al. (2009) also formulate our demand fulfillment problem as a stochastic dynamic program. It turns out that the optimal policy of the DP is a generalization of the booking-limit policy which is easy to implement. However, because of the “curse of dimensionality”, it is computationally expensive and therefore not really applicable for real-size problem.

The purpose of this paper is then to develop the bid-price control methods to solve the demand fulfillment problem in our MTS system. Since bid-price controls are proved to be successful in the traditional revenue management settings, it is reasonable for us to expect a similar performance in the MTS environment. However, due to the differences we identify between our problem and the problems of the two research streams mentioned above, we are not just applying the existing methods in a different setting, but develop bid-price control methods to solve a new different problem.

In summary, we make the following contributions to the field:

- We identify the similarities and differences between the demand fulfillment problem in an MTS system and network revenue management problems.
- Using insights from the traditional revenue management settings, we develop three bid-price control models to solve a demand fulfillment problem in an MTS production system.
- We evaluate the performance of the three bid-price control models numerically and compare them to other existing benchmarking methods.

The paper is structured as follows. In § 2, we review the current literature and further motivate our research. In § 3, we explain the problem setting and the stochastic dynamic programming formulation of the problem. In § 4, we analyze three airline bid-price control methods and adapt them to our problem setting. § 5 provides a numerical study which compares the relative performance of the proposed three bid-price control methods and two benchmark policies. We conclude with §6 and discuss future research potentials.

2. Literature Review

In literature, there are different research streams for solving network revenue management problem. In this section we only review the bid-price control methods.

Most of the work on bid-price control in network revenue management problems is within the airline industry and considers only specific products. As an emerging topic, a few papers discuss the situation with flexible products.

2.1 Bid-Price Control Model for Network Revenue Management with Specific Products

In general, the bid-price control models can be classified as generating a static or dynamic estimate of the marginal value of the remaining capacities. While a static model yields bid prices only on basis of the remaining time and capacity at the time of computation, the ultimate goal of dynamic models is to generate bid prices for every possible time-capacity combination until departure (Talluri and van Ryzin, 2004). Despite the different properties of the generated bid prices, the key ideas behind all of the models are the same: To approximate the dynamic programming formulation of the original problem by certain efficient mathematical programming formulation, e.g., linear programming, and calculate the bid prices by solving the dual problem (Bertsimas and Popescu, 2003).

Static models

Among the models proposed in literature to compute bid prices, static models are distinguished by the essential characteristic that the resulting bid prices do not change as a function of time or capacity, but stay constant until recomputed.

Williamson (1992) was one of the first to propose the so-called deterministic linear program (DLP) to compute bid prices as the optimal dual prices. Assuming demand equals to its mean, the author uses the partitioned allocation of capacity for different products as the decision variable with the objective to maximize the total revenue. Talluri and van Ryzin (1998) carefully analyze the resulting policy and point out that the DLP is actually a linear functional approximation to the dynamic programming value function of the network revenue management problem. The main advantage of the DLP model is that it is intuitive and efficient to solve. The weakness is that it treats demand as being deterministic and considers only expected demand while neglecting all further distributional information (Talluri and van Ryzin, 2004; Kunnumkal and Topaloglu, 2010). Despite this shortcoming, several numerical studies have shown that with frequent recalculation, the DLP bid-price control model generate promising performance and outperforms the probabilistic nonlinear programming model (Wiliamson, 1992; Belobaba and Lee, 2000; Belobaba, 2001).

With slight additional complexity, Talluri and van Ryzin (1999) refine the DLP and encompass more distributional information in their randomized linear program (RLP) by substituting the expected demand with independent samples of the random demand. Talluri and van Ryzin claim that the RLP allows a closer to optimal revenue although their computational results do not confirm an absolute dominance over the DLP in a random network setting. Topaloglu (2009) reinvestigates the relative performance of the DLP and RLP model under different scenarios with different problem parameters and number of samples. The results show that on a majority of the test problems, the RLP model is a robust solution method and performs better than the DLP model.

As another attempt to capture the randomness in demand, the probabilistic nonlinear programming (PNLP) method is developed. Its main difference compared to the DLP model is that the PNLP calculates the total revenue based on expected sales instead of the partitioned allocation of capacity, i.e., it considers the possibility that real demand might be lower than the allocated quantity. However, simulations have found it to be usually outperformed by the DLP (Talluri and van Ryzin, 2004).

Bertsimas and Popescu (2003) propose an alternative application of a linear approximation to estimate the marginal value of capacities. Instead of computing leg-based bid prices via dual solutions, their certainty equivalent control (CEC) estimates the opportunity cost (OC) for each itinerary by computing the marginal value of capacity. As with typical bid-price controls, a request is accepted if and only if the proposed fare exceeds the estimated OC. The authors report a revenue increase of 5 to 10 percent over the DLP-based bid-price controls. The main disadvantage of the CEC method is that one has to solve a separate LP for each product, which is computationally much more expensive than the DLP-based bid-price control.

Dynamic models

As mentioned above, the static models do not incorporate the dynamics of the underlying system and generate reasonable bid prices only under frequent reoptimization. In practice, however, frequent reoptimization might not be feasible due to the limitation of computation capacity. Thus, a dynamic model, which generates bid prices that vary with time and capacity and therefore can be solved less frequently, is appealing.

To develop such dynamic model, Adelman (2007) proposes to make an affine functional approximation to the value function of the DP and plug them into the linear programming formulation of the DP. Solving its dual with column generation, the author obtains a time trajectory of bid prices all at once. In the numerical study, Adelman (2007) shows that the dynamic model outperforms the static bid-price controls by up to 21.4%.

With the same objective to capture the temporal dynamics of demand, Kunnumkal and Topaloglu (2010) relax the capacity constraints of the DP using Lanrangian relaxation. Consequently, their method decomposes the optimality equation by periods remaining until departure and yields bid prices that vary with time. The two dynamic models (Adelman, 2007; Kunnumkal and Topaloglu, 2010) generate very similar time-trajectories and performance in the proposed settings.

Topaloglu (2009) moves a step further and approaches the NRM problem with the goal of computing bid prices that not only encompass the temporal dynamics within the system, but are also contingent on the remaining capacities on the different flight legs. Similar to Kunnumkal and Topaloglu (2010), the author uses Langrangian relaxation to decompose the NRM problem into a sequence of single-leg revenue management problems. Concentrating on one flight leg at a time, the author generates both capacity-and time-dependent bid prices. Computational experiments indicate that the model outperforms the benchmark strategies such as DLP, RLP and the model proposed by Adelman (2007) within the suggested experimental setup, but with more computational expense.

2.2 Network Revenue Management with Flexible Products

As the first publication that introduces the concept of flexible products for revenue management, Gallego and Phillips (2004) consider a simple two-period, two-flight problem for an airline offering a flexible product at a discount in addition to specific products. They provide EMSR- based algorithms for calculating booking limits on both specific and flexible products. The numerical study shows that under reasonable assumptions, offering flexible products generates considerable benefits.

Gallego et al. (2004) extend the work of Gallego and Phillips (2004) to a more general network setting with an arbitrary number of products. The authors consider a continuous time model and approximate the DP of the resulting network revenue management problem by a deterministic linear program which can be considered as a generalization of the LP approximation of the usual network revenue management problem without flexible products studied by Williamson (1992) and Talluri and van Ryzin (1998). Using numerical experiments, they verify how the benefits of offering flexible products vary as a function of various parameters like time horizon, discount etc.

With very similar problem setting as Gallego et al. (2004), Petrick et al. (2012) discretize the planning horizon into individual time periods such that there is at most one order arrival in each period. Different from Gallego et al. (2004) who assume that the assignment to different alternatives for flexible products can only occur at the end of the planning horizon, Petrick et al. (2012) allow an arbitrary notification date within the planning horizon, during which all accepted flexible requests have to be assigned to an available alternative and after which no more flexible products may be sold. The authors then provide the dynamic programming formulation of the problem and extend three popular static approximation models, namely the DLP, RLP and CEC method, to the flexible products case. The authors report an increase in revenue of up to 4% due to incorporating flexible products and the DLP-based bid-price control model exploits the additional flexibility best.

For network revenue management problem with or without flexible products, the order acceptance rule is the same: An incoming order is only accepted if there is enough capacity available and its revenue exceeds the sum of the bid prices of all required resources. But if flexible products are incorporated, this is no longer the end of the story as one still needs to decide which alternative to assign to each accepted flexible request. If this decision is made at the end of the planning horizon, it is possible to achieve an optimal assignment as one has observed all the demands, like Gallego et al. (2004) who model the assignment problem as a LP. If an arbitrary notification date is allowed, the problem is more complex as the current assignment can limit the flexibility within the remaining planning horizon. In Petrick et al. (2012), a flexible product is assigned to the alternative with the highest difference between revenue and the corresponding bid prices. Petrick et al. (2010) investigate different assignment mechanisms that differ in the extent to which they exploit the flexibility.

2.3 Network Revenue Management in Manufacturing

Literature on bid-price control for network revenue management problem in manufacturing is very rare. Due to the direct analogy between the perishable production capacity in a make-to-order (MTO) system and the perishable flight seats in airline industry, it is possible to use most of the aforementioned models for the order acceptance problem in an MTO environment.

Spengler et al. (2005) implement a static bid-price control to manage the order promising in a make-to-order system in iron and steel industry. Since the orders obtained in their problem setting are unique and cannot be classified into classes, the standard deterministic linear programming approximation, which is restricted to multiple fare classes, is not applicable here. So the authors employ a multi-dimensional knapsack problem formulation. According to the computational analysis using real world production data, the proposed bid-price controls perform significantly better than a first-come-first-serve (FCFS) strategy.

We are not aware of any work that applies bid-price control in a make-to-stock manufacturing system.

For our demand fulfillment problem described in Section 1, beside Quante et al. (2009) who model it as a stochastic dynamic program, Meyr (2009) proposes a two-step procedure to solve it: in the first allocation planning step, a deterministic linear programming model which is similar to the one from Williamson (1992) is developed with the objective to maximize the overall profit. Its optimal solution is used as partitioned quantity reserved for each customer class and each arrival period. In the second order promising step, allocated quantities are consumed by incoming orders in real time based on certain consumption rules. Following the same framework, Quante (2008) adapts the randomized linear programming (RLP) idea from Talluri and van Ryzin (1999) and solves Meyr (2009)'s DLP model repetitively with realizations of the random demand with known distribution. The optimal allocation quantity is estimated by a weighted-average of the results over all repetitions.

As among all the static models, the DLP (Williamson, 1992; Talluri and van Ryzin, 1998) and the RLP model (Talluri and van Ryzin, 1999) are proved to be efficient and perform well, in this paper, we use the two models from Meyr (2009) and Quante (2008) as the primal problem to calculate the corresponding static bid prices. To capture the temporal dynamics of demand, we adapt Adelman (2007)'s affine functional approximation method to calculate the dynamic bid prices.

3. Model Formulation

We consider a MTS manufacturing system with exogenously determined replenishments and stochastic demand from heterogeneous customers. In order to maximize the expected profit, the manufacturer has to decide for each arriving order whether to satisfy it from stock, backorder it at a penalty cost, or reject it, in anticipation of more profitable future orders. The manufacturer needs to take into account not only sales revenues, but also inventory holding costs and backorder penalties.

With the following assumptions, Quante et al. (2009) model this demand fulfillment problem as a stochastic dynamic program: (1) A finite planning horizon of T is considered and is subdivided into discrete time periods $t = 1, \dots, T$, such that there is at most one order arrival in each period; (2) Inventory replenishment schedule is known and x_i denotes the ATP quantities arriving at the beginning of period i, $i = 1, \dots, T$; (3) Customers are differentiated into C different classes, $c = 1, \dots, C$, with corresponding unit revenues of r_c. Demand of a given class follows a compound Poisson process and is independent of the demand from other classes, as well as the available supply; (4) Order due dates equal to the arrival date. Delaying an order causes backorder cost of b per unit per period and the unit holding cost is h per period; (5) Partial delivery is allowed and u_i is used to denote the amount of ATP quantities arriving in period i used to satisfy a given order.

The notations are summarized in Table 1.

Table 1 Notation

Indices:	
$t = 1, \dots, T$	Periods of the planning horizon
$i = 1, \dots, T$	Periods of inventory replenishment
State variables:	
$\vec{x} = (x_1, \dots, x_T)$	Vector of available supply quantities
Decision variables:	
$\vec{u} = (u_1, \dots, u_T)$	Vector of supply quantities used to fulfill a given order
Random variables:	
c	Customer class
d	Order quantity
$F(c, d)$	Joint cdf of customer class c and order quantity d
Data:	
r_c	Unit revenue from customer class c
b	Unit backorder cost per period
h	Unit holding cost per period

Source: Quante et al. (2009)

As mentioned in the introduction, what makes this problem complex is that unlike traditional airline revenue management, one has to decide not only whether to satisfy a given order or not, but also which supply to use and how many of each supply to use.

In modeling, this is reflected as follows: In the airline environment, for a specific order, its required resources are known which is normally defined by an incidence matrix, e.g., matrix $A \equiv (a_{j,k})$ where $a_{j,k} = 1$ if resource j is used by product k and $a_{j,k} = 0$ otherwise (Talluri and van Ryzin, 2004; Adelman, 2007; Topaloglu, 2009). Then in period t, the decision variable u_t is a binary variable where $u_t = 1$ if the request is accepted and $u_t = 0$ otherwise. In our MTS system, however, the order size is a random variable which is normally larger than one and we do not have the incidence matrix A. So for an incoming order, we have to decide which resource to use and how many of each resource to use, as all finished products are physically identical. Therefore, in period t, our decision variable is an integer vector $\vec{u} = (u_1, \dots, u_T)$ where u_i denotes the amount of ATP quantities arriving in period i used to satisfy the given order ($u_i = 0$ means that the ATP from period i is not used for this order) and $\sum_{i=1}^{T} u_i$ represents the total amount used to satisfy the order.

Using $V_t(\vec{x})$ to denote the maximum expected profit-to-go from period t to the end of the planning horizon, Quante et al. (2009) develop the following Bellman equation

$$V_t(\vec{x}) = E_{d,c}\left[\max_{\substack{\vec{u} \\ 0\le u_i \le x_i, \sum_{i=1}^{T} u_i \le d}} \left\{\sum_{i=1}^{T} (u_i P_t(i,c) - h x_i \delta_{it}) + V_{t+1}(\vec{x} - \vec{u})\right\}\right] \qquad (1)$$

where $P_t(i,c)$ is defined as the incremental profit per unit of ATP i used to satisfy one unit of an order of class c in period t and δ_{it} is defined as 1 if $i \le t$ and 0 otherwise.

After analyzing the structural properties, Quante et al. (2009) prove that the optimal policy of the proposed stochastic dynamic program resembles a booking-limits policy which sets nested protection levels for each class and supply arrival. Supplies are consumed in a First-In-First-Out (FIFO) order, i.e., for each incoming order, either the earliest available supply is used to satisfy it or the order is rejected.

In the numerical study, Quante et al. (2009) show that their model outperforms current common fulfillment policies, such as first-come-first-served (FCFS) and the deterministic optimization model from Meyr (2009). However, as mentioned in the introduction, because of its high-dimensional state space, this model has very limited scalability.

As mentioned in the previous chapter, the basic idea of bid-price control methods for NRM problem is to approximate the corresponding DP by simpler models, such as mathematical programming. For our problem, the DP formulation is provided by Quante et al. (2009) as above. In the next section we propose three bid-price control methods which generate efficient yet well-performing approximations to (1).

4. Bid-Price Control Models for Demand Fulfillment

As mentioned in the introduction, the order acceptance process of our demand fulfillment problem has several differences compared to the traditional network revenue management problems. For the traditional case, if only specific products are considered, an incoming order is accepted if and only if its revenue exceeds the sum of the bid prices of the required resources. For situation where flexible products are offered, one needs to go a step further to assign an alternative to each accepted flexible request. For our problem, one has to go even further as one still needs to decide how many of each resource to use. Therefore, each of the following bid-price control models we propose contains two steps, namely a bid price calculation step and an additional order promising step for deciding the final consumption scheme.

4.1 Bid-price control based on DLP

Following the same problem settings, Meyr (2009) models our demand fulfillment problem as a deterministic linear program. If we use atp_i to denote the initial available inventory of replenishment i, D_{ct} to denote the random demand from class c that arrives in period t (with known distribution information), and as decision variable, y_{ict} to denote the partitioned allocation of each atp_i to class c that arrives in period t, the DLP model can be formulated as follows.

$$max \sum_{i=1}^{T} \sum_{c=1}^{C} \sum_{t=1}^{T} p_{ict} \cdot y_{ict} \qquad (2)$$

subject to

$$\sum_{i=1}^{T} y_{ict} \leq E(D_{ct}) \qquad \forall c, t \qquad (3)$$

$$\sum_{c=1}^{C} \sum_{t=1}^{T} y_{ict} \leq atp_i \qquad \forall i \qquad (4)$$

$$y_{ict} \geq 0, \text{integer} \qquad \forall i, c, t \qquad (5)$$

where p_{ict} represents the profit of using one unit of supply i to satisfy the order from customer class c with arrival date t, and can be calculated as follows.

$$p_{ict} = r_c - b(i - t)(1 - \delta_{it}) - h(t - i)\delta_{it}$$

Note that the above formulation charges inventory holding cost only when supply is allocated. Inventory holding cost for unallocated supply is not considered. Actually, it is easy for us to also include the inventory holding cost for unallocated supply in the model, but in order to stay in line with the original model from Meyr (2009), we leave it out here. Whether or not to include the inventory holding cost for unallocated supply does not have any impact on the numerical results here, because the inventory holding cost is low enough that it is always beneficial to allocate the supply to customers, whenever possible.

The optimal value of the objective function can be considered as an approximation to the value function of the original DP, and Meyr (2009) uses the primal solution directly as the partitioned quantity reserved for each customer class and arrival date, based on which some rule-based order processing methods are used to complete the demand fulfillment problem.

Following the idea of Williamson (1992) and Talluri and van Ryzin (1998), we do not use the primal solution of the above DLP but calculate the optimal set of dual variables associated with constraint (4) as the bid prices for each corresponding ATP supply.

To process the incoming order, for each supply i, we first calculate the difference between the net profit of using one unit of this supply to satisfy the incoming order and the bid price of this supply.

$$p_{ict} - BP_i \tag{6}$$

Then we choose the supply with the highest positive difference to satisfy the order. If the chosen supply has no enough quantity to fulfill the order, we move to the supply with the second highest positive difference, and so on, until it is not beneficial to use any supply to satisfy the order, i.e., equation (6) generates only negative results, or there is no more supply available. We then stop the order promising procedure for this order and move to the next one.

To choose supply, actually we should compare the difference between the profit of using certain supply and its corresponding bid price. Here, the sunk inventory holding cost is included in the calculation of the net profit p_{ict}. As p_{ict} is used as the coefficient in the objective function, the resulting bid price BP_i also considers the sunk inventory holding cost. To be consistent with the bid price, we have to use p_{ict} here to calculate the difference as it also includes the sunk inventory holding cost.

4.2 Bid-price control based on RLP

Similar to the DLP model in airline setting, Meyr (2009)'s model is efficient to solve, but has been criticized as it neglects the demand uncertainty and only takes the expected demand into consideration. To overcome this limitation, Quante (2008) borrows the idea of randomized linear programming (Talluri and van Ryzin, 1999) and modifies Meyr (2009)'s model by replacing the expected demand in constraint (3) with random demands drawn from the known demand distribution. The resulting LP is then solved repetitively, each with an independent sample of the random demand.

In his PhD thesis, Quante (2008) uses the weighted average primal solution as the partitioned allocation quantity. Different from that, we discard the primal solutions and calculate the RLP-based bid prices based on the associated dual prices. Assume that the model is solved N times, it then provides N dual prices for each resource. Following the idea of Talluri and van Ryzin (1999), we calculate the final bid price for supply i by taking average of the N dual prices of supply i.

$$BP_i = \frac{\sum_{n=1}^{N} BP_i^n}{N} \qquad (7)$$

where BP_i denote the final bid price of supply i and BP_i^n is the shadow price of supply i in sample $n(n = 1, \dots, N)$.

The order promising procedure is then the same as the DLP-based bid-price control in 4.1.

4.3 Dynamic bid-price control

The above two models generate only static bid prices which do not capture the temporal dynamics of the system. To obtain time-dependent dynamic bid prices, Adelman (2007) derives a model which computes a time trajectory of bid prices all at once.

The main steps of this model are as follows: (1) Make an affine functional approximation to the value function of the DP; (2) Plug the affine functional approximations into the linear program formulation of the DP; (3) Solve the dual problem using column generation and obtain the corresponding bid prices.

Following the steps, we derive our dynamic bid-price control model in what follows.

We start with our original dynamic program (1).

Similar to Adelman (2007), we use the available supply quantities of replenishment i in period t as the basis functions, and approximate the value of the state vector $\vec{x}$ by

$$V_t(\vec{x}) \approx \theta_t + \sum_i V_{t,i} \cdot x_i \qquad \forall t, \vec{x} \qquad (8)$$

where the parameter $V_{t,i}$ is the estimation of the marginal value of a unit of supply i in period t, or in other word, the bid price of ATP supply i in period $t-1$, and θ_t is a constant offset. We assume $V_{T+1,i} = 0$ and $\theta_{T+1} = 0$.

The state vector $\vec{x}$ satisfies

$\vec{x} \in \mathcal{X} \equiv \{\vec{x} \in \mathbb{Z}_+^T : x_i \in \{0,1, \dots, atp_i\}\ \forall i\}$.

In period 1, we have $\vec{x} = \overrightarrow{atp}$, so for the further analysis, we define

$$\mathcal{X}_t = \begin{cases} \{\overrightarrow{atp}\} & \text{if } t = 1 \\ \mathcal{X} & \text{if } t = 2, \dots, T \end{cases}$$

Assume the maximal possible order size is M, we can then use a $(T \times C \times M)$-vector $\vec{u} \equiv \{u_{i,c,d}\}$ to denote the supply quantity used to satisfy an order from a certain class with a certain order size. When the system is in state $\vec{x}$, this vector has to satisfy

$$\vec{u} \in \mathcal{U}_{\vec{x}} \equiv \{u_{i,c,d} \in \mathbb{Z}_+ : u_{i,c,d} \le x_i, \textstyle\sum_i u_{i,c,d} \le d, \forall i, c, d\} \qquad \forall \vec{x}$$

Compared to Adelman (2007) where a one-dimensional binary vector is used to denote the acceptance decision, here we need a three-dimensional integer vector $\vec{u}$. This is due to the fact we discussed in the previous section that in the traditional airline setting with only specific products, the only decision needs to be made is whether or not to accept an incoming order, but in our MTS case, we have to decide which supply to use and how many of each supply to use.

Then, the linear programming formulation of the DP from Section 3 can be written as

$(\boldsymbol{D_0})$ $\quad \min_{V(.)} V_1(\overrightarrow{atp})$

$$V_t(\vec{x}) \geq \sum_c \sum_d F(c,d) \cdot \left[\sum_i \left(u_{i,c,d} P_t(i,c) - hx_i\delta_{it}\right) + V_{t+1}\left(\vec{x} - \vec{u}_{c,d}\right)\right] + \left(1 - \sum_c \sum_d F(c,d)\right) \cdot V_{t+1}(\vec{x}) \qquad \forall t,\ \vec{x} \in \mathcal{X}_t,\ \vec{u} \in \mathcal{U}_{\vec{x}} \tag{9}$$

Note that as our initial DP is different from Adelamn (2007), the resulting linear programming formulation is also different here. Substituting the affine functional approximation into the linear program, it becomes

$(\boldsymbol{D_1})$

$$\min_{\theta,V} \theta_1 + \sum\nolimits_i V_{1,i} \cdot atp_i \tag{10}$$

$$\theta_t - \theta_{t+1} + \sum_i \left[V_{t,i} \cdot x_i - V_{t+1,i}\left(x_i - \sum_c \sum_d F(c,d) \cdot u_{i,c,d}\right)\right] \geq \sum_c \sum_d F(c,d) \cdot \left[\sum_i \left(u_{i,c,d} P_t(i,c) - hx_i\delta_{it}\right)\right] \qquad \forall t,\ \vec{x} \in \mathcal{X}_t,\ \vec{u} \in \mathcal{U}_{\vec{x}} \tag{11}$$

Note that by using equation (8) to approximate the value function, we reduce the number of decision variables from

$1 + (T-1) \cdot \prod_{i=1}^{T} (atp_i + 1)$

which is exponential in T, to $T(T+1)$. However, $\boldsymbol{D_1}$ still has an exponential number of constraints. Therefore, we use column generation to solve its dual problem $\boldsymbol{P_1}$.

$(\boldsymbol{P_1})$

$$z_{p_1} = \max_Y \sum_{t,\vec{x}\in\mathcal{X}_t,\vec{u}\in\mathcal{U}_{\vec{x}}} \left(\sum_c \sum_d F(c,d) \cdot \left[\sum_i \left(u_{i,c,d} P_t(i,c) - hx_i\delta_{it}\right)\right]\right) \cdot Y_{t,\vec{x},\vec{u}} \tag{12}$$

$$\sum_{\vec{x}\in\mathcal{X}_t,\vec{u}\in\mathcal{U}_{\vec{x}}} x_i\, Y_{t,\vec{x},\vec{u}} \qquad \forall i,t \tag{13}$$

$$= \begin{cases} atp_i & \text{if } t = 1, \\ \sum_{\vec{x} \in \mathcal{X}_{t-1}, \vec{u} \in \mathcal{U}_{\vec{x}}} \left(x_i - \sum_c \sum_d F(c,d) \cdot u_{i,c,d} \right) \cdot Y_{t,\vec{x},\vec{u}} & \forall t = 2, \dots, T \end{cases}$$

$$\sum_{\vec{x} \in \mathcal{X}_t, \vec{u} \in \mathcal{U}_{\vec{x}}} Y_{t,\vec{x},\vec{u}} = \begin{cases} 1 & \text{if } t = 1 \\ \sum_{\vec{x} \in \mathcal{X}_{t-1}, \vec{u} \in \mathcal{U}_{\vec{x}}} Y_{t-1,\vec{x},\vec{u}} & \forall t = 2, \dots, T \end{cases} \qquad \forall t \tag{14}$$

$Y \geq 0.$

$V_{t,i}$ are then the dual prices on constraint (13), and we can interpret the decision variable $Y_{t,\vec{x},\vec{u}}$ as state-action probabilities since constraint (14) can be rewritten as

$$\sum_{\vec{x} \in \mathcal{X}_t, \vec{u} \in \mathcal{U}_{\vec{x}}} Y_{t,\vec{x},\vec{u}} = 1 \qquad \forall t.$$

To use column generation, we first need an initial feasible solution to $\boldsymbol{P_1}$ to start the recursion. Then, we solve $\boldsymbol{P_1}$ with the initial feasible solution to obtain the corresponding dual prices. Using the obtained dual prices as input, we solve the sub-problem to decide whether to add any additional columns to the existing solution set. Finally, we add the chosen columns to the existing solution set and repeat the procedure until the stopping criterion is met.

Similar to Adelman (2007), the "offering nothing" strategy provides a feasible solution to our $\boldsymbol{P_1}$, i.e.,

$$\hat{Y}_{t,\vec{x},\vec{u}} = \begin{cases} 1 & \text{if } x_i = atp_i, u_{i,c,d} = 0, \forall i,c,d \\ 0 & \text{otherwise.} \end{cases} \qquad \forall t, \vec{x} \in \mathcal{X}_t, \vec{u} \in \mathcal{U}_{\vec{x}} \tag{15}$$

Assume the resulting dual prices are denoted by V, θ, the sub-problem can be written as follows

$$\max_{t,\vec{x} \in \mathcal{X}_t, \vec{u} \in \mathcal{U}_{\vec{x}}} \pi_{t,\vec{x},\vec{u}} = \max_{t,\vec{x} \in \mathcal{X}_t, \vec{u} \in \mathcal{U}_{\vec{x}}} \sum_c \sum_d F(c,d) \cdot \left[\sum_i \left(u_{i,c,d} P_t(i,c) - h x_i \delta_{it} \right) \right] - \sum_i \left[V_{t,i} \cdot x_i - V_{t+1,i} \left(x_i - \sum_c \sum_d F(c,d) \cdot u_{i,c,d} \right) \right] - \theta_t + \theta_{t+1}$$

which maximizes the reduced profit from (11). When $t = 1$, we have $x_i = atp_i, \forall i$, and for any fixed $t > 1$, the sub-problem can be rewritten as the following integer program, specifying the conditions on the solution set explicitly as constraints.

$$\max_{\vec{x},\vec{u}} \sum_c \sum_d F(c,d) \cdot \left[\sum_i \left(u_{i,c,d} \left(P_t(i,c) - V_{t+1,i} \right) - h x_i \delta_{it} \right) \right] - \sum_i \left(V_{t,i} - V_{t+1,i} \right) \cdot x_i - \theta_t + \theta_{t+1} \tag{16}$$

$$u_{i,c,d} \leq x_i \qquad \forall i,c,d \tag{17}$$

$$\sum_i u_{i,c,d} \leq d \qquad \forall c, d \qquad (18)$$

$$x_i \in \{0, \dots, atp_i\} \qquad \forall i \qquad (19)$$

$$u_{i,c,d} \geq 0, \text{integer} \qquad \forall i, c, d \qquad (20)$$

If the objective value of (16) is positive, we add the corresponding column to the existing set of columns for $\boldsymbol{P_1}$, i.e., for each iteration, we do not add only one column (the one with the maximal reduced profit) as the standard column generation algorithm does, but we add a batch of columns, one for each time period, as long as the associated reduced profit is positive.

As stopping criterion, we specify a percentage φ, such that as soon as the sum of the optimal objective values of the sub-problems ($\sum_t \pi_t^*$) is smaller than φ-percent of the optimal objective value of $\boldsymbol{P_1}$ with the current set of columns, we stop the column generation iteration.

Using $\mathfrak{I}$ to denote the current set of columns and $Z_{\mathfrak{I}}$ to denote the corresponding optimal objective value of P_1, the column generation algorithm is summarized in Table 2.

Table 2 Column generation algorithm

Algorithm Column generation

Set $\mathfrak{I} = \{(t, \overrightarrow{atp}, \vec{0}) \forall t\}$, solve the restricted problem ($\mathrm{P}_1(\mathfrak{I})$), and set $\pi_t^* = \infty$ for all t.

while $\sum_t \pi_t^* \geq \varphi Z_{\mathfrak{I}}$ do
  for all $t \in (1, \dots, T)$
    compute $\pi_t^* = \max_{\vec{x},\vec{u}} \pi_{t,\vec{x},\vec{u}}$
    select an $(\vec{x}_t, \vec{u}_t) \in \arg\max_{\vec{x},\vec{u}} \pi_{t,\vec{x},\vec{u}}$
    update $\mathfrak{I} \leftarrow \mathfrak{I} \cup \{(t, \vec{x}_t, \vec{u}_t)\}$.
  solve $\mathrm{P}_1(\mathfrak{I})$)

The order promising procedure is almost the same as the one we propose for the DLP-based bid-price control model. The only difference is that for choosing supply, we use the following equation (21) instead of equation (6) to calculate the difference between the profit of using certain supply and its corresponding bid price: In period t, we compare the incremental profit $P_t(i, c)$ of the incoming order to the *current* bid price of the corresponding supply and calculate the difference

$$P_t(i, c) - V_{t+1,i} \qquad (21)$$

Here the incremental profit $P_t(i, c)$ is used because the corresponding bid price $V_{t+1,i}$ is calculated based on the profit-to-go, i.e., the sunk inventory holding cost is not included.

5. Numerical Study

In order to evaluate the performance of their stochastic dynamic programming model (the DP discussed in Section 3), Quante et al. (2009) set up a simulation-based numerical study framework. In what follows, we define a similar numerical experiment with a test bed containing a wide range of problem instances and use simulation to evaluate the performance of the proposed bid-price control models. In subsection 5.1 we define the test beds and in subsection 5.2 we analyze the results of the numerical study.

5.1 Test bed

To make our numerical results comparable with Quante et al. (2009), we set up a similar test bed which is based on a full factorial design with five design factors and six fixed parameters. We fix our planning horizon to 14 periods with two inventory replenishments in period 1 and period 8. Replenishments quantity is fixed to 50 units each time, i.e., $\mathrm{ATP}_1 = \mathrm{ATP}_8 = 50$. We consider 3 customer classes with different revenues. Inventory holding cost is fixed to \$1 per unit per period. We assume that the mean demand order size of each incoming order is constant and equal to 12 units. We summarize our choices for the design factors and fixed parameters in Table 3. Different from our settings here, Quante et al. (2009) consider only the first three design factors and assume equal order arrival probabilities and a fixed backlogging cost of \$10 per unit per period for all customer classes.

Table 3 Design factors and fixed parameters for the numerical study

Name	**Value**
Fixed parameters	
Planning horizon (T)	14
Arrival periods of replenishments	Period 1, Period 8
Replenishments quantity (S)	50
Number of customer classes (C)	3
Inventory holding cost (h)	1
Mean demand per order (μ)	12
Design factors	
Coefficient of variation of order size (CV)	$\left\{\frac{1}{3}, \frac{5}{6}, \frac{4}{3}, \frac{11}{6}\right\}$
Customer heterogeneity ($\mathbf{r}$)	$\{(100,90,80), (100,80,60), (100,70,40)\}$
Supply shortage rate (sr)	$\{40\%, 24\%, 1\%\}$

Customer arrival ratio (**w**)	$\{(1:2:3),(1:1:1),(3:2:1)\}$
Backlogging cost proportion (b)	$\{0.05,0.1,0.2\}$

The total number of all possible combinations for these design factors is $3^4 \times 4 = 324$, i.e. we have 324 scenarios. For each scenario, we generate 30 different demand profiles and run the corresponding simulations for every policy. This gives us in total $324 \times 30 = 9720$ instances for each policy in our numerical study. This scenario size ensures that when we test the significance of performance difference and the impact of the design factors, both of the type I and type II error are limited to 5%.

We now explain the design factors in detail. The first factor in the factorial design is the coefficient of variation of order size (CV). We fix the mean of the order size to $\mu = 12$, but the actual order size can vary from order to order and the variation is represented by the coefficient of variation of the order size $CV = \sigma/\mu$, where σ is the standard deviation of the order size. We choose the same range of CV as Quante et al. (2009) to ensure a reasonable range of variability.

The second factor of the factorial design is customer heterogeneity, which is represented by the revenue vector $r = (r_1, r_2, r_3)$ of the customer classess. The revenue vector $(100,90,80)$ represents a low customer heterogeneity while $(100,70,40)$ represents a high customer heterogeneity. These choices are also identical to Quante et al. (2009).

The third factor of the factorial design is the supply shortage rate (sr), which reflects the degree of supply scarcity. It is defined as follows:

$$sr = 1 - \frac{\sum_{i=1}^{T} ATP_i}{(1 - p_0) \times \mu \times T}$$

Since in our case the supply quantity and the mean demand of each order are both fixed, the supply shortage rate (sr) depends solely on the probability of no order arrival p_0. A large p_0 corresponds to a low shortage rate while a small $p_0$leads to a high shortage rate. In our factorial design, we vary sr between 1% and 40% by varying p_0 from 0.4 to 0. We choose these levels because since we only consider situations where supply is scarce, 1% shortage rate is almost the lowest shortage rate we can use and 40% corresponds to a no arrival probability of 0 and is therefore the highest shortage rate we can use. Quante et al. (2009) use the same levels for the shortage situation, but also consider two more levels for oversupply, i.e., sr being negative.

The fourth factor of the factorial design is customer arrival ratio (w). This factor reflects the fraction of demand from each customer class. For instance, when no arrival probability $p_0 = 0$, a customer arrival ratio $w = (1:2:3)$ corresponds with an arrival probability of $1/6$ for Class 1, $1/3$ for Class 2 and $1/2$ for Class 3.

The fifth factor of the factorial design is the backlogging cost proportion (b). Quante et al. (2009) assume a fixed backlogging cost for all customer classess. We generalize this assumption to allow different backlogging cost for different customer classes, as customers from different classes pay different prices. In the numerical study, we assume that the backlogging cost for different customer class is proportional to the corresponding revenue. When this proportion is small, e.g., $\mathrm{b} = 0.05$, backlogging penalty is low and when this proportion is large, e.g., $\mathrm{b} = 0.2$, backlogging cost takes 20% of the revenue, which makes the penalty high. Considering the holding cost $\mathrm{h} = 1$, the chosen levels of backlogging cost ratio ensure that the resulting service level is within a reasonable range, e.g., if we fix the other parameters to their middle values (i.e., $\mathrm{CV} = \frac{13}{12}, \mathrm{r} = (100{,}80{,}60), \mathrm{sr} = 24\%, \mathrm{w} = (1:1:1)$), our replenishment schedule achieves an average cycle service level between 56% and 82% for all classes if we vary b from 0.05 to 0.2.

5.2 Performance Comparison of Different Bid-Price Control Models

Literature shows that for traditional NRM problem, if the static bid-price control models are re-optimized frequently, they perform quite well (Talluri and van Ryzin, 2004). According to Adelman (2007), resolving the dynamic bid-price control model also leads to a better result. This motivates us to consider the proposed bid-price control models both with and without resolving. The considered policies are summarized as follows.

- **DLP-BPC**: Solve the DLP model in §4.1 once. Given a fixed set of DLP-based static bid prices, use the order promising procedure in §4.1.
- **DLP-BPC Resolved:** Resolve the DLP model in §4.1 every 4 periods over the time horizon T. Between solution epochs, use the order promising procedure in §4.1.
- **RLP-BPC**: Solve the RLP model in §4.2 once with $N = 30$. Given a fixed set of RLP-based static bid prices, use the order promising procedure in §4.1.
- **RLP-BPC Resolved:** Resolve the RLP model in §4.2 with $N = 30$ every 4 periods over the time horizon T. Between solution epochs, use the order promising procedure in §4.1.
- **DBPC**: Solve the dynamic model in §4.3 once. Given a set of fixed dynamic bid prices, use the order promising procedure in §4.3.
- **DBPC Resolved**: Resolve the dynamic model in §4.3 every 4 periods over the time horizon T. Between solution epochs, use the order promising procedure in §4.3.
- **Stochastic dynamic programming model (SDP):** This strategy applies the optimal policy of the DP from Quante et al. (2009) that we are approximating. It provides the optimal ex-ante policy and therefore serves as a benchmark to calculate the optimality gap in our numerical comparison.

For the dynamic bid price control models, we choose the optimality tolerance of $\varphi = 1\%$, i.e., we stop the column generation iteration as soon as the sum of the optimal objective value of the sub-problem is smaller than 1% of the optimal objective value of $\boldsymbol{P_1}$. This optimality tolerance is smaller than Adelman (2007)'s 5% and thus provides a more accurate estimation.

Using the test bed, we obtain the simulated profits of all the 9720 instances for each of the fulfillment strategies. Using a standard PC with a 3.2GHz Intel Core CPU and 32.00GB memory, the average run-time for one simulation instance is summarized in Table 4. The run-time data shows that all the bid-price control models we propose are much more efficient than the SDP model. The dynamic model takes longer than the static models, but is still tractable.

Table 4 Run Time Data

	SDP	DLP-BPC	RLP-BPC	DBPC	DLP-BPC Resolved	RLP-BPC Resolved	DBPC Resolved
Run time (seconds)	1774.56	2.54	3.57	12.35	3.16	3.99	17.82

By comparing the simulated profits of other strategies to the simulated profits of the **SDP** model, we obtain the optimality gaps. We then calculate the average optimality gap for all the above mentioned models over (i) all 9720 test instances, and (ii) all subsets where one of the design factors is fixed to one of its admissible values. The results are shown in Table 5. Beside the average optimality gap (shown in bold), we also show the average backlog percentage (first value in parenthesis), the average lost sales percentage (second value in parenthesis) and the ratio between the average service levels of Class 1 and Class 3 (third value in parenthesis) of each strategy. As complementary data, the second and third row of Table 5 differentiate the average backlogging percentage and average lost sale percentage by customer, for each fulfillment model.

Table 5 Simulation Results

Test bed subset	N	Average optimality gap (%)(backlog pct., lost sale pct., differentiation ratio)			
		SDP	DLP-BPC	RLP-BPC	DBPC
All instances	9720	**0.00**(4.34, 24.39, 1.45)	**7.96**(5.55, 30.55, 1.95)	**6.72**(5.66, 29.59, 1.87)	**3.17**(4.45, 26.90, 1.70)
Avg. backlogging (%) (Cl.1, Cl.2, Cl.3)		*(6.07, 4.19, 1.52)*	*(6.92, 4.92, 3.11)*	*(6.39, 5.43, 3.25)*	*(5.33, 4.16, 1.93)*
Avg. lost sale (%) (Cl.1, Cl.2, Cl.3)		*(0.12, 0.19, 0.39)*	*(0.12, 0.22, 0.55)*	*(0.12, 0.22, 0.53)*	*(0.12, 0.19, 0.49)*
CV = 1/3	2430	**0.00**(3.18, 24.58, 1.82)	**4.59**(4.99, 27.42, 1.93)	**5.91**(4.06, 28.70, 2.10)	**4.68**(2.71, 28.09, 2.23)
CV = 5/6	2430	**0.00**(4.22, 24.66, 1.57)	**7.51**(5.94, 29.92, 1.97)	**8.03**(5.24, 30.72, 2.13)	**3.25**(3.99, 27.40, 1.95)
CV = 4/3	2430	**0.00**(4.36, 24.20, 1.33)	**9.68**(5.35, 31.97, 2.00)	**8.28**(5.68, 30.88, 1.83)	**3.03**(4.61, 26.96, 1.57)
CV = 11/6	2430	**0.00**(5.59, 24.12, 1.19)	**10.76**(5.90, 32.90, 1.91)	**4.50**(7.66, 28.07, 1.51)	**1.40**(6.50, 25.15, 1.30)
r = (100,90,80)	3240	**0.00**(4.43, 23.53, 1.28)	**10.46**(4.11, 31.55, 1.98)	**6.99**(4.91, 28.32, 1.63)	**3.57**(4.13, 25.93, 1.52)
r = (100,80,60)	3240	**0.00**(4.37, 24.54, 1.44)	**7.08**(5.91, 30.48, 1.97)	**6.88**(5.77, 29.89, 1.91)	**2.90**(4.60, 26.84, 1.69)
r = (100,70,40)	3240	**0.00**(4.21, 25.10, 1.66)	**5.85**(6.62, 29.62, 1.91)	**6.21**(6.31, 30.58, 2.12)	**2.98**(4.63, 27.93, 1.95)
sr = 1%	3240	**0.00**(4.73, 11.84, 1.09)	**2.03**(8.63, 10.83, 1.00)	**1.06**(7.62, 11.06, 1.01)	**0.80**(6.25, 11.32, 1.03)
sr = 24%	3240	**0.00**(5.13, 23.61, 1.41)	**8.58**(4.65, 32.75, 2.61)	**4.68**(6.28, 28.42, 2.01)	**2.84**(4.68, 26.77, 1.85)
sr = 40%	3240	**0.00**(3.15, 37.72, 2.31)	**11.98**(3.36, 48.07, 7.03)	**12.98**(3.08, 49.30, 9.24)	**5.32**(2.43, 42.61, 4.25)
w = (1:2:3)	3240	**0.00**(4.36, 24.53, 1.38)	**8.70**(5.78, 30.82, 1.64)	**7.49**(6.00, 29.91, 1.61)	**3.93**(4.29, 27.36, 1.53)
w = (1:1:1)	3240	**0.00**(3.94, 24.32, 1.47)	**7.84**(5.44, 30.27, 2.02)	**5.42**(5.24, 28.97, 2.00)	**3.20**(4.10, 27.06, 1.78)
w = (3:2:1)	3240	**0.00**(4.70, 24.32, 1.50)	**7.45**(5.42, 30.57, 2.32)	**7.27**(5.74, 29.90, 2.06)	**2.51**(4.96, 26.28, 1.84)
b = 0.05	3240	**0.00**(5.84, 23.98, 1.47)	**7.89**(7.84, 30.57, 1.99)	**6.87**(7.67, 29.47, 1.89)	**3.38**(5.68, 26.74, 1.73)
b = 0.1	3240	**0.00**(4.47, 24.31, 1.45)	**7.63**(5.18, 30.33, 1.95)	**6.73**(5.81, 29.43, 1.88)	**3.02**(4.74, 26.59, 1.70)
b = 0.2	3240	**0.00**(2.70, 24.87, 1.42)	**8.37**(3.62, 30.76, 1.91)	**6.54**(3.50, 29.88, 1.83)	**3.13**(2.94, 27.37, 1.68)

Table 5 Simulation Results (continued)

Test bed subset	**N**	**Average optimality gap (%)**(backlog pct., lost sale pct., differentiation ratio) **DLP-BPC Resolved**	 **RLP-BPC Resolved**	 **DBPC Resolved**
All instances	9720	**2.80**(4.75, 25.99, 1.61)	**2.80**(4.63, 26.29, 1.60)	**2.47**(4.97, 26.11, 1.65)
Avg. backlogging (%) (Cl.1, Cl.2, Cl.3)		*(6.56, 4.17, 1.95)*	*(5.99, 4.14, 1.95)*	*(5.95, 4.33, 2.70)*
Avg. lost sale (%) (Cl.1, Cl.2, Cl.3)		*(0.13, 0.21, 0.46)*	*(0.12, 0.22, 0.45)*	*(0.12, 0.19, 0.46)*
CV = 1/3	2430	**1.84**(4.23, 25.36, 1.80)	**2.20**(3.39, 25.99, 1.92)	**2.57**(2.04, 26.81, 2.22)
CV = 5/6	2430	**2.77**(5.00, 26.20, 1.70)	**2.88**(4.49, 26.70, 1.77)	**2.07**(3.42, 26.71, 1.91)
CV = 4/3	2430	**3.20**(4.43, 26.04, 1.53)	**3.28**(4.51, 26.42, 1.47)	**2.39**(4.94, 25.96, 1.47)
CV = 11/6	2430	**3.55**(5.35, 26.35, 1.42)	**2.92**(6.12, 26.07, 1.34)	**2.89**(9.48, 24.96, 1.25)
r = (100,90,80)	3240	**3.32**(3.42, 26.00, 1.58)	**3.08**(3.78, 25.81, 1.49)	**2.74**(4.98, 25.08, 1.46)
r = (100,80,60)	3240	**2.56**(5.10, 26.23, 1.63)	**2.69**(4.74, 26.52, 1.62)	**2.38**(4.98, 26.26, 1.65)
r = (100,70,40)	3240	**2.41**(5.74, 25.73, 1.62)	**2.59**(5.37, 26.55, 1.71)	**2.23**(4.96, 26.99, 1.88)
sr = 1%	3240	**1.97**(6.53, 11.69, 1.07)	**1.18**(5.58, 12.04, 1.09)	**1.02**(5.85, 12.03, 1.09)
sr = 24%	3240	**2.60**(4.66, 25.73, 1.64)	**2.92**(5.23, 25.94, 1.63)	**2.55**(5.36, 26.02, 1.78)
sr = 40%	3240	**3.61**(3.07, 40.54, 3.32)	**3.95**(3.07, 40.90, 3.10)	**3.51**(3.71, 40.27, 3.10)
w = (1:2:3)	3240	**2.89**(5.12, 25.71, 1.39)	**2.85**(5.09, 26.10, 1.44)	**2.61**(4.64, 26.38, 1.53)
w = (1:1:1)	3240	**2.81**(4.65, 25.71, 1.58)	**2.80**(4.18, 26.30, 1.66)	**2.61**(4.64, 26.18, 1.71)
w = (3:2:1)	3240	**2.70**(4.48, 26.54, 1.96)	**2.76**(4.60, 26.49, 1.76)	**2.22**(5.63, 25.77, 1.72)
b = 0.05	3240	**2.59**(6.22, 25.74, 1.63)	**2.71**(6.05, 25.92, 1.63)	**2.64**(6.59, 25.75, 1.66)
b = 0.1	3240	**2.75**(4.68, 26.00, 1.63)	**2.93**(4.75, 26.26, 1.61)	**2.41**(4.99, 26.06, 1.66)
b = 0.2	3240	**3.05**(3.36, 26.22, 1.56)	**2.76**(3.08, 26.70, 1.57)	**2.35**(3.34, 26.52, 1.61)

From the first row in Table 5, we see that without resolving, the performances of the two static bid-price control models are close to each other (with an average optimality of 7.96% for DLP-BPC and 6.72% for RLP-BPC) but substantially worse than the dynamic model (with an average optimality gap of 3.17%). For all the optimality gaps in Table 5, their 95% confidence intervals are within ± 0.35.

As expected, resolving the bid-price control models improve the performance. The DLP-based bid-price model benefits most from re-optimization with an average optimality gap decrease from 7.96% to 2.80% while the dynamic bid-price control model benefits least with an average optimality gap decrease from 3.17% to 2.47%. This is intuitive as the DLP-based model takes neither demand uncertainty nor system dynamics into consider, it has the highest potential for improvement. On the other hand, the dynamic bid-price model incorporates both factors at the first place, therefore resolving only leads to marginal improvement.

Actually, with the relatively high resolving frequency of every 4 periods, the performances of all three bid-price control models are quite similar, and are also very close to the dynamic model without resolving. Considering the computational time, the static models with resolving are even more efficient than the dynamic model without resolving. Therefore, one may conclude that for practical purpose, it might be better to adopt the static models and resolve them frequently than to use the dynamic model, as the static models generate similar results and are more efficient to solve. However, we have to notice that, in practice, very frequent re-optimization is usually not feasible. For instance, in the airline industry, re-calculation is normally executed overnight and during the day, there is no chance for re-optimization. The situation in MTS production system is similar. In this case, the dynamic model which incorporates system dynamics and generates a bid-price trajectory is much more appealing than the static models which have constant bid prices. This is also the motivation of developing dynamic bid-price control models (Adelman, 2007; Topaloglu, 2009; Kunnumkal and Topaloglu, 2010). Our simulation results also show that without frequent resolving, the DBPC model performs much better than the DLP-BPC and RLP-BPC model.

As the SDP model provides the optimal solution to our problem, we compare the decisions (i.e., the backlogging, lost sale and service level behavior reflected in the bracketed value of Table 5) made by the bid-price control models to it to understand their performance differences. From the first three rows of Table 5 we can see that, with frequent resolving, the three bid-price control models tend to behave quite similar: They do not only generate very close average optimality gap, but also have very similar backlogging and lost sale behavior. Therefore, we treat the resolved versions as one model and choose the DLP-BPC Resolved model as the representative for the following performance analysis.

Regarding lost sales, the SDP model has an average lost sale of 24.39%. Considering different customer classes, it has the highest lost sale rate for Class 3 and the lowest rate for Class 1, which shows clear class differentiation. If we further consider its backlogging behavior we can see that it backlogs much more for Class 1 and 2 than for Class 3. This behavior is reasonable because given the high revenue of Class 1, it is usually more profitable to backlog an order from Class 1 than to lose it, which leads to a high backlogging rate for Class 1. For Class 3, it is the other way round: it is usually better to keep the supply for future more profitable orders than to backlog it for Class 3.

Compared to the SDP model, the DLP-BPC model has a much higher average lost sale (30.55%). For Class 1, its lost sale rate is the same as SDP, but it loses much more customers from the lower classes, e.g., for Class 3, it loses more than half of the customers. Due to its very high lost sale rate of Class 3, the DLP-BPC model has the highest ratio between the average service levels of Class 1 and Class 3. This shows that the DLP-BPC model tends to over-protect Class 1 customers. Regarding backlogging, the DLP-BPC model backlogs more for each class. This excessive backlogging behavior suggests that the DLP-BPC model might under-estimate the value of the second supply during the demand fulfillment process. We will discuss this issue further in the sensitivity analysis.

The RLP-BPC model has a very similar behavior pattern as the DLP-BPC model, but performs slightly better. The average lost sale is 29.55% and for Class 1 and Class 2, it has the same lost sale rate as the DLP-BPC model, but it loses a little less Class 3 customers. Regarding backlogging, it backlogs a little more than the DLP-BPC model. Compared to the SDP model we can see that the RLP-BPC model has also the "over-protection" problem, but less severe than the DLP-BPC model.

The DBPC model performs closest to SDP among all three bid-price control models without resolving. With an average lost sale rate of 26.90%, it achieves the same service level for Class 1 and Class 2 as the SDP. For Class 3, its lost sale rate is higher than SDP but lower than both DLP-BPC and RLP-BPC. Regarding backlogging behavior, its backlogging rate for each class is also lower than both static models, i.e., the DPBC model achieves a higher service level with even less backlogging, which suggests that by incorporating temporal dynamics, the DBPC provides a better estimation of bid prices than the static models.

The DLP-BPC Resolved model performs quite similar as the DBPC model. It achieves an even lower lost sale rate of 25.99%. Compared to the static bid-price models without resolving, the DLP-BPC Resolved model lost more Class 1 and Class 2 customers but less Class 3 customers, which leads to a lower differentiation ratio. This means that this resolved version releases the over-protection problem to a certain extent, which contributes to its better performance.

The following Figure 1 shows the bid-price trajectories of the three models for different shortage rates when the other parameters are fixed to their medium values (i.e., $\mathrm{CV} = \frac{4}{3}, \mathrm{r} = (100{,}80{,}60), \mathrm{w} = (1:1:1), \mathrm{b} = 0.1$).

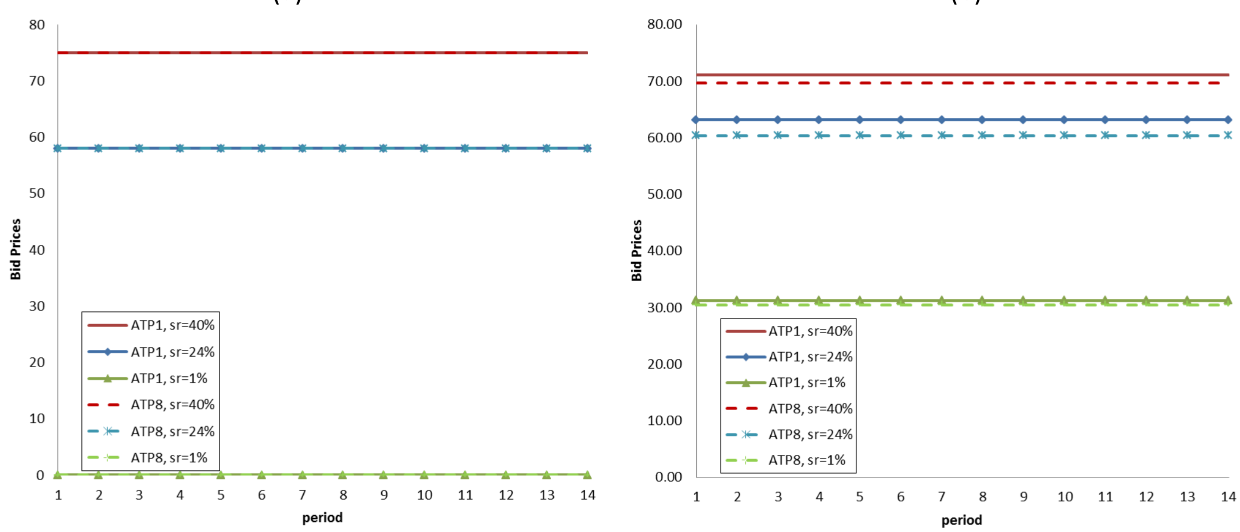
The dynamic bid-price trajectory shows a decreasing pattern in time and its shape is the same as the optimal booking-limit trajectory in Quante et al. (2009). The two curves, representing bid price of ATP1 and ATP8, converge in period 7, because from period 8 on, the two supplies are the same, i.e., they are both on hand inventory and generate the same profit for incoming orders. Towards the end of the planning horizon, the bid price drops drastically. This is intuitive, as we assume that after the planning horizon, unsold inventory has no value at all.

For the two static models, by definition, the bid prices are constant and do not change as over time. From Figure 1 we can see that when supply shortage rate is high (sr = 40%), both bid prices generated by the static models are higher than 60, which means Class 3 customers are always rejected. Compared to that, the dynamic model performs more reasonably. Towards the end of the planning horizon, the bid price drops below 60, i.e., Class 3 customers are acceptable in the last few periods. This makes sense because at the end of the planning horizon, the chance to sell becomes so little that one should not miss any incoming orders if one still have inventory on hand. From the above analysis we can see that in order to improve the performance, updating is necessary for the static models.

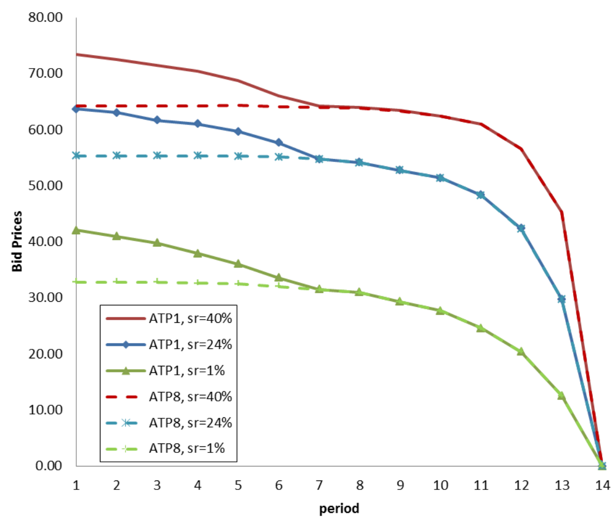
Compared to the RLP-BPC and the DBPC model, the DLP-BPC model tends to overestimate the bid prices when shortage rate is high and underestimate them when shortage rate is low. For example, when supply shortage rate is low (sr = 1%), the bid prices generated by the DLP-BPC model are 0, which makes the DLP-BPC model reduce to a FCFS policy. This might explain its poor performance in Table 5. We also note that in Figure 1(a), the bid price of ATP1 coincides with the bid price of ATP8. This is because in our example here, backlogging is relatively expensive ($b = 0.1$). The DLP-BPC model tends to avoid any backlogging, which makes the problem in the second supply interval (Period 8 - 14) a copy of the problem in the first supply interval (Period 1 to 7). Therefore, the bid prices of the two supplies become the same.

Figure 1 Bid-price trajectory of : (a) DLP-BPC, (b) RLP-BPC, (c) DBPC

(a) (b)



(c)



In summary, we have the following findings from the performance comparison:

- Our best-performing method, the dynamic bid-price control model, achieves a close approximation to the optimal SDP model (with an optimality gap of only 3.17% for the no-resolving version and 2.47% for the resolved version) with much lower computational effort.
- Without resolving, DBPC provides a better estimation of bid prices and performs substantially better than the static models
- DLP-BPC and RLP-BPC have excessive backlogging behavior which suggests that they under-estimate the value of second supply
- All bid-price control models tend to over-protect the more profitable customers
- Resolving improves performance, DLP-BPC benefits most
- With resolving, all three models have very close performance

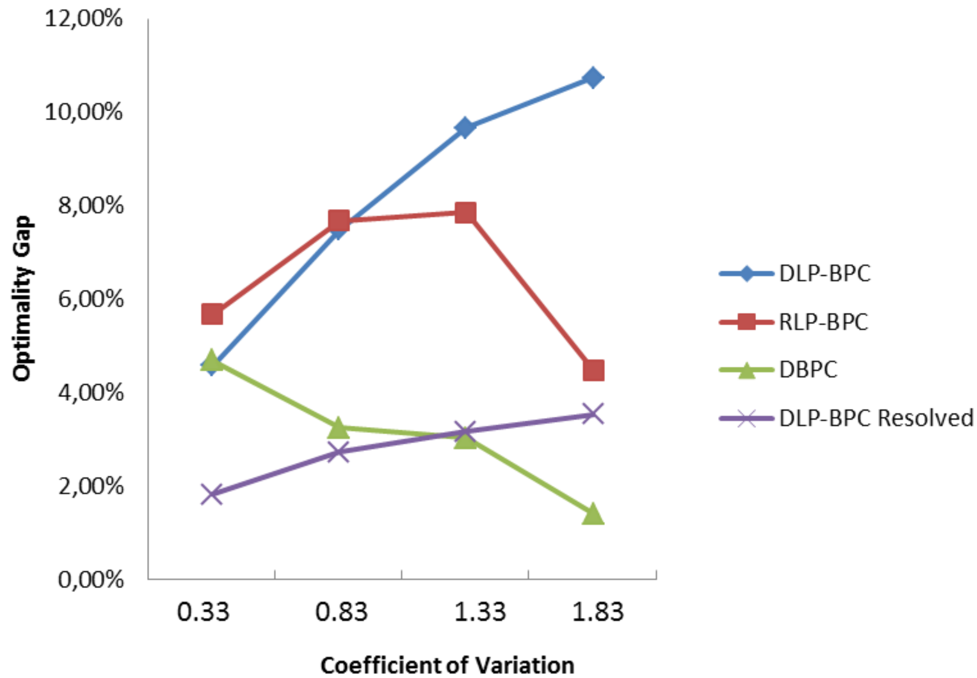
5.3 Sensitivity Analysis

The following part of Table 5 provides information on the impact of different design factors on the performance of each fulfillment model. The customer arrival ratio (w) and the backlogging cost proportion (b) turn out to have little impact on the performance of the models, thus we leave them out in the sensitivity analysis. The coefficient of variation of the order size (CV), customer heterogeneity (r), and supply shortage rate (sr) have a greater impact on the resulting optimality gap of each model. We discuss their impact in what follows.

<u>Coefficient of variation of order size (CV)</u>

From Table 5 and Figure 2, we can see a clear dependency between the optimality gaps and the CV values.

Figure 2 Average optimality gap for different CV values



We observe the following: (1) As the CV value increases, the DLP-BPC show a clear increasing trend in its average optimality gap. (2) For RLP-BPC, it has the same trend as CV increases from 0.33 to 1.33, but the optimality gap drops surprisingly as CV increases to 1.83. (3) The DBPC model shows a decreasing pattern in its average optimality gap as demand uncertainty increases. (4) When demand distribution is very low ($CV = 0.33$), the performance of all three bid-price control models (without resolving) are close to each other. As demand variability increases, the dynamic model performs substantially better than the two static models. (5) The DLP-BPC Resolved model shows same performance pattern as DLP-BPC, but performs much better.

The fast increasing optimality gap of the DLP-BPC model can potentially be attributed to the fact that this model considers only the expected demand. As the CV value increases, the generated bid prices do not change because the expected demand keeps constant. Therefore, this model ignores demand uncertainty totally, which makes its lost sale percentage increase and its performance drop drastically as demand variability increases. With resolving, the DLP-BPC resolved model performs much better because the actual demand realization is incorporated.

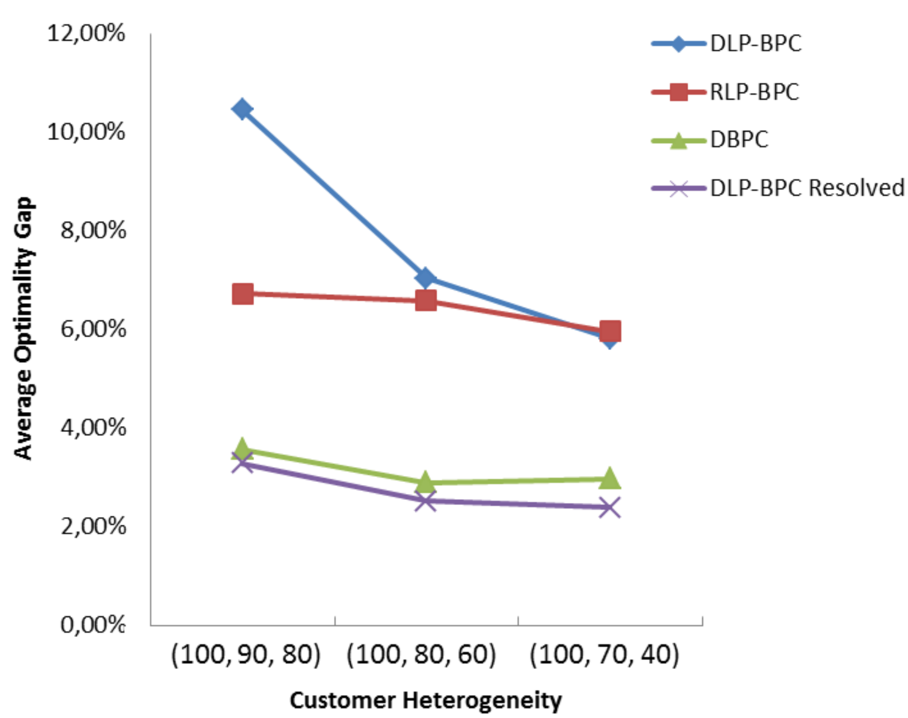
For the RLP-BPC model, when demand uncertainty is low, its performance is close to the DLP-BPC model. This is intuitive as when CV is low, the randomly generated demand is close to the mean, which makes the resulting average bid price close to the DLP-based version. As CV increases, the RLP-BPC model performs better than the DLP-BPC model and when CV increases to 1.83, its optimality gap even decreases. This might be because that when demand uncertainty is high, the randomly generated demand is no longer close to the mean, but represents better the real demand distribution. Therefore, the RLP-BPC model generates a better estimation of the bid prices, i.e., the randomization becomes more effective when demand uncertainty is really high.

As CV increases, the DBPC model increases backlogging and reduces class differentiation. By doing so, it reduces the average lost sale rate as demand uncertainty increases. Therefore, its lost sale rate is getting closer and closer to the SDP model, which might explain the decreasing performance discrepancy between the two models.

Customer Heterogeneity (r)

From Table 5 and Figure 3, we observe that customer heterogeneity shows great impact only on the DLP-BPC model. For the other models, there is no clear dependency between the resulting average optimality gap and customer heterogeneity. For DLP-PBC, its optimality gap decreases as the scale of customer heterogeneity increases.

Figure 3 Average optimality gap for different customer heterogeneity



SDP's main reaction to the increase of customer heterogeneity is to increase the class differentiation, which is reflected in the increasing value of the ratio between the average service levels of Class 1 and Class 3 (third value in parenthesis). This reaction is reasonable because it is more beneficial to better serve the more profitable customers when heterogeneity is high. This increased class differentiation leads to an increase of the lost sale rate.

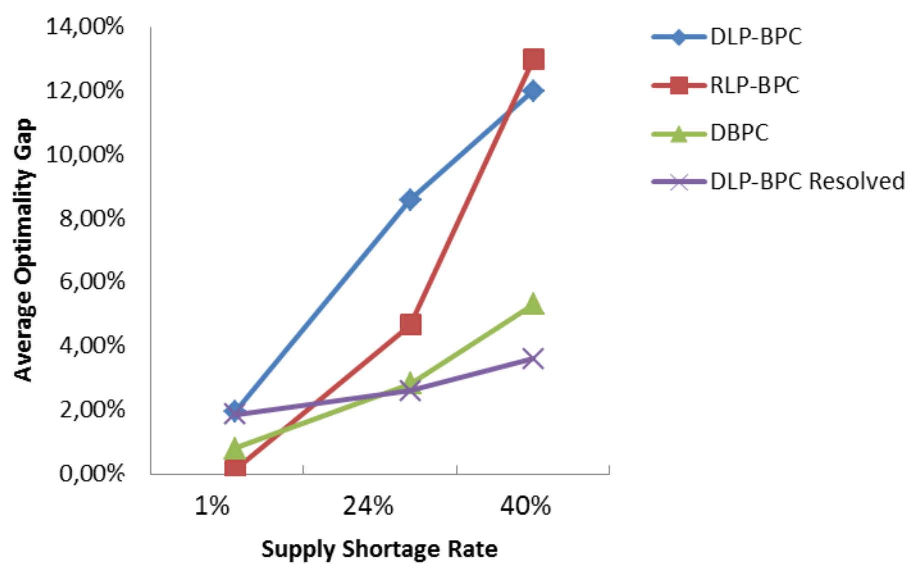
However, the DLP-BPC model keeps its differentiation ratio constant, which means it does not react to different heterogeneity levels at all and keeps over-protecting the more profitable customers. From

SDP's reaction we know that this "over-protecting" behavior only makes sense when customer heterogeneity is high. Therefore, the optimality gap of the DLP-BPC model decreases as customer heterogeneity increases.

Supply shortage rate (sr)

Finally, we consider the impact of the degree of supply scarcity. From Table 5 and Figure 4 we observe: (1) The supply scarcity has a huge impact on the performance of the bid-price control models, especially on the two static models. (2) The performance of all three bid-price control models shows a decreasing pattern as the shortage increases.

Figure 4 Average optimality gap for different supply scarcity



The three proposed bid-price control models without resolving and the DLP-BPC Resolved model increase class differentiation as supply becomes scarcer. This is intuitive to understand: As the models intend to keep the same service level for the higher classes, less supply are left for the lower classes when shortage increases.

However, compared to SDP which provides the "right" response to parameter changes, the bid-price control models seem to overreact to the shortage increase – when shortage rate is low ($sr = 1\%$), their ratios between the average service level of Class 1 and 3 is actually smaller than SDP, i.e., they do not differentiate enough, but the increase of their ratios is much higher than the SDP. When the shortage rate is intermediate or high ($sr = 24\%, 40\%$), for the bid-price control models, the higher the average service level ratio, the higher the corresponding lost sale rate and optimality gap, which shows that the bid-price models indeed overreact to the shortage increases and therefore their performance is hurt.

Regarding backlogging, all four bid-price models decrease their backlogging behavior as shortage increases. This is in line with their differentiation behavior: As class differentiation increases, the more profitable customers are better served. Therefore, the necessity for backlogging decreases. For the DLP-

BPC model, we find out that its excessive backlogging mainly happens with low shortage rate ($sr = 1\%$). From Figure 1 we have already seen that with low shortage rate ($sr = 1\%$), the DLP-BPC model indeed under-estimate the bid price of ATP8 as it is much lower than the estimation of the other two models. But actually from Figure 4 we see that when shortage rate is low ($sr = 1\%$), the performance of DLP-BPC is close to the other bid-price control methods. This shows us that the excessive backlogging behavior is not the main reason for DLP-BPC's poor performance. The over-protection behavior, which leads to high lost sale rate, is the main problem.

6. Conclusion

In this paper, we consider the demand fulfillment problem in make-to-stock manufacturing where customers are differentiated into different segments based on their profitability.

After discussing the similarities and differences between our demand fulfillment problem and the traditional network revenue management problems, we develop three bid-price control models to solve the problem, based on the idea of approximating the DP by simpler mathematical programming.

The numerical study shows that, without frequent resolving, the dynamic bid-price control model provides a better estimation of bid prices and performs substantially better than the static models. Its performance is also very close to the optimal SDP model which is the optimal ex-ante policy but computational very expensive.

With resolving, all bid-price control models have similar performance. However, we have to know that in reality, frequent resolving is usually not realistic. Therefore, the DBPC model which generates close-to-optimal results with tractable computational time seems to strike a reasonable balance between performance and computational expense.