http://pubsonline.informs.org/journal/isre/

INFORMATION SYSTEMS RESEARCH
Vol. 30, No. 2, June 2019, pp. 595–615
ISSN 1047-7047 (print), ISSN 1526-5536 (online)

Efficient Computational Strategies for Dynamic Inventory Liquidation

Mochen Yang,[a] Gediminas Adomavicius,[b] Alok Gupta[b]

[a] Kelley School of Business, Indiana University, Bloomington, Indiana 47405; [b] Carlson School of Management, University of Minnesota, Minneapolis, Minnesota 55455

Contact: yangmo@iu.edu, http://orcid.org/0000-0001-5101-9041 (MY); gedas@umn.edu, http://orcid.org/0000-0001-5251-5098 (GA); gupta037@umn.edu, http://orcid.org/0000-0002-2097-1643 (AG)

Received: October 20, 2016
Revised: October 5, 2017; June 13, 2018
Accepted: September 1, 2018
Published Online in Articles in Advance: May 30, 2019

https://doi.org/10.1287/isre.2018.0819

Abstract. We develop efficient computational strategies for the inventory liquidation problem, which is characterized by a retailer disposing of a fixed amount of inventory over a period of time. Liquidating end-of-cycle products optimally represents a challenging problem owing to its inherent stochasticity. The growing scale of liquidation problems further increases the need for solutions that are revenue- and time-efficient. We propose to address the inventory liquidation problem by deriving deterministic representations of stochastic demand, which provides significant theoretical and practical benefits as well as an intuitive understanding of the problem and the proposed solution. First, this paper develops a dynamic programming approach and a greedy heuristic approach to find the optimal liquidation strategy under deterministic demand representation. Importantly, we show that our heuristic approach is optimal under realistic conditions and is computationally less complex than dynamic programming. Second, we explore the relationships between liquidation revenue and several key elements of the liquidation problem via both computational experiments and theoretical analyses. We derive multiple managerial implications and demonstrate how the proposed heuristic approach can serve as an efficient decision support tool for inventory managers. Third, under stochastic demand, we conduct a comprehensive set of simulation experiments to benchmark the performance of our proposed heuristic approach with alternatives, including other simple approaches (e.g., the fixed-price strategy) as well as advanced stochastic approaches (e.g., stochastic dynamic programming). In particular, we consider a strategy that uses the proposed greedy heuristic to determine prices iteratively throughout the liquidation period. Computational experiments demonstrate that such iterative strategy stably produces higher total revenue than other alternatives and produces near-optimal total revenue in expectation while maintaining significant computational efficiency, compared with advanced techniques that solve the liquidation problem directly under stochastic demand. Our work advances the computational design for inventory liquidation and provides practical insights.

History: Kartik Hosanagar, Senior Editor; Gautam Pant, Associate Editor.
Supplemental Material: The online supplement is available at https://doi.org/10.1287/isre.2018.0819.

Keywords: inventory liquidation • scalable computational approaches • dynamic programming • greedy heuristic • simulation • design science

1. Introduction

Management of excess inventory is a common concern of retailers. The need to liquidate inventory may occur for a number of reasons. For instance, products may be seasonal in nature, newer versions of products may be forthcoming or have been recently released, or a retailer may no longer wish to carry a product line. The rapid release cycle of electronic devices represents one typical scenario in which inventory liquidation frequently happens. Apple, for example, releases a new generation of iPhone almost every year, at which point customer valuations for the previous generation products usually decline rapidly over time. Therefore, retailers have a strong motivation to liquidate inventory in a time-sensitive and revenue-efficient manner.

The area of inventory liquidation of consumer products has been revolutionized by retailers adopting direct-to-consumer approaches via the e-commerce stores and online auction sites (Bapna et al. 2009) enabled by advanced information technologies. For example, retailers such as Best Buy have liquidation stores on eBay, and sites such as uBid.com often liquidate previous generation electronics products. One of the key advantages of online platforms is the ability to consolidate inventories across stores and sell them to a wider consumer base that may not be accessible through traditional channels. The challenge of liquidating a large amount of consolidated inventory is in design of liquidation strategies (e.g., over what period of time the inventory should be liquidated and how prices should be adjusted throughout that period) given the stochasticity of demand. Computational approaches constitute a promising avenue toward either automating or providing support for managerial decision making in inventory liquidation settings. Furthermore, some of the emerging online B2C platforms (e.g., liquidation.com) need to deal with large-scale liquidation problems in real time. For them, utilizing highly scalable liquidation strategies is crucial for surviving in the growing market.

In this paper, we design highly computationally efficient strategies to liquidate a given inventory of identical items over a predetermined time horizon via periodic pricing. A retailer can set an item price for each time period on the basis of his or her knowledge about customer demand, to maximize total revenue from liquidation. We adopt the design science approach of Hevner et al. (2004) and establish the foundations of an *inventory liquidation support system*, with the goal of providing insights about the performance and implications of different computational inventory liquidation strategies and enabling retailers to dispose inventory in a revenue-efficient manner under a variety of settings. The inventory liquidation problem that we seek to address has strong relevance for business operations. Our proposed computational strategies are technology-based design artifacts, which provide viable solutions to the liquidation problem—especially in large-scale settings, where the business requires significant time efficiency (i.e., scalability) in managing inventory and doing so in an automated or semiautomated manner. In particular, our proposed approach represents a highly computationally efficient search process for finding the best inventory liquidation decisions in the vast space of potential liquidation actions. We provide a rigorous, in-depth performance evaluation of our approach with respect to a number of alternatives and benchmarks. As a result, we contribute to the knowledge and practice of inventory management and communicate our research findings both to the technology-oriented audience (by presenting theoretical, algorithmic, and simulation results) and to the management-oriented audience (by discussing managerial insights obtained from simulation experiments). Thus, our approach is closely aligned with a number of canonical design-science guidelines for information systems research, such as the *design as an artifact*, *problem relevance*, *design evaluation*, *research contributions*, *research rigor*, *design as a search process*, and *communication of research* guidelines (Hevner et al. 2004, p. 83).

Specifically, we first establish a theoretical model that describes the key elements and processes of an inventory liquidation problem. We consider stochastic demand and assume that the seller has high-level distributional knowledge about the demand, which is a standard assumption in the inventory management literature (e.g., Gallego and Van Ryzin 1994, Bitran and Mondschein 1997). However, differently from prior work, we propose to address the problem by first deriving deterministic representations of the stochastic customer demand (i.e., by calculating expected arrivals and expected valuations; details in Section 3.1). Taking advantage of these deterministic representations, we first discuss two simple and commonly used liquidation strategies: (1) setting a fixed per-item price for the entire inventory over the duration of the liquidation period; and (2) setting different prices from day to day (typically decreasing), which is designed to achieve a certain fixed daily sales quantity over the entire liquidation period. Given the retailer's goal to maximize revenue from liquidation, we point out that both strategies are, not surprisingly, suboptimal. Thus, it is imperative to utilize more sophisticated, dynamic pricing mechanisms to yield the highest revenue. Second, we propose (i) a dynamic programming approach that is generally optimal (i.e., achieves the maximum revenue for the retailer) under deterministic representation of stochastic demand; and (ii) a specialized greedy heuristic approach, which is much faster but can still solve the inventory liquidation problem optimally under deterministic representation of stochastic demand for a variety of commonly observed realistic customer valuation distributions. Third, we provide both theoretical and computational results regarding how the changes in the inventory liquidation problem configurations, such as initial level of inventory, length of liquidation period, and customer demand characteristics, affect the resulting optimal liquidation strategy and revenue. Our results have direct managerial implications that can inform the inventory management decisions. Finally, we conduct comprehensive simulations to evaluate the performance of our proposed heuristic in realistic settings (i.e., under a number of diverse instances of stochastic demand) by comparing it with a variety of benchmarks. We demonstrate that, under stochastic demand, liquidation strategies generated by the heuristic consistently produce near-optimal revenues with highly superior computational efficiency.

In summary, using a design science approach, we contribute to the information systems literature by proposing a computational approach with a greedy heuristic that can solve large-scale inventory liquidation problems with remarkable computational efficiency and revenue performance under a wide variety of realistic conditions. We also contribute to the dynamic pricing literature by highlighting an advantageous way to solve complex, stochastic inventory problem—the derived deterministic representations turn out to be appropriate and useful approximations of stochastic demand. Besides theoretical contributions, our work also offers rich managerial insights on the relationships between key liquidation decisions and the resulting optimal liquidation revenues and strategies.

2. Literature Review

The design science approach in information systems research seeks solutions to business problems by producing design artifacts (Hevner et al. 2004). Although this paper is rooted in such a research paradigm, it is also closely related to, and informed by, the broad literature on dynamic pricing and inventory management.

Dynamic pricing with inventory considerations is an important topic both for management science researchers and for practitioners. Previous literature on dynamic pricing and inventory management has focused on two broad categories of pricing mechanisms: *posted-price* mechanisms and *price-discovery* mechanisms (Elmaghraby and Keskinocak 2003). Under posted-price mechanisms, the retailer makes pricing decisions. Customers view the price as given and make purchasing decisions based on their willingness to pay. Under price-discovery mechanisms, however, prices are determined via auctions (McAfee and McMillan 1987). In this paper, we consider posted-price mechanisms for inventory liquidation, although auction-based price-discovery mechanisms may also be applied to inventory management (e.g., Wood et al. 2005), and our approaches can readily be adapted for multiunit auctions (Bapna et al. 2003, 2005). For example, by using the derived prices as marginal valuations of customers on a given day, an auctioneer can decide how many units to sell on that day and parameterize the multiunit auctions according to the approach presented in Bapna et al. (2003). A seller can increase their revenue by using discriminatory price versions of the auction (e.g., Yankee auctions) to extract more consumer surplus.

Depending on market characteristics, dynamic pricing problems under posted-price mechanisms often involve three key dimensions: replenishment versus no replenishment of inventory, dependent versus independent demand over time, and myopic versus strategic customers (Elmaghraby and Keskinocak 2003). The first dimension represents whether inventory replenishment is allowed during the time horizon under consideration. The second dimension concerns with whether demand is time-dependent. Although demand for durable goods is typically time-dependent because of rare repeat purchases, demand for nondurable goods or necessity products is mostly time-independent (Elmaghraby and Keskinocak 2003). The third dimension describes customer behavior. Myopic customers will always buy the product when the posted price is not above their willingness to pay, whereas strategic customers may choose to postpone purchases and wait for the price to further decrease.

A large body of literature has focused on different variations of dynamic pricing problems for inventory liquidation, which is typically characterized with no replenishment of inventory, because the retailer's goal is to sell out a given inventory. For example, Gallego and Van Ryzin (1994) modeled the liquidation of a fixed amount of inventory over a finite time horizon as a continuous pricing problem, where demand followed a time-invariant Poisson process and was dependent on prices. They established relationships between prices and two key parameters: inventory size and liquidation period length. They also showed that a simple heuristic, such as a fixed-price strategy, can be asymptotically optimal. Bitran and Mondschein (1997) examined a model for pricing seasonal goods within a fixed sales window. They relaxed the assumption of time-invariant demand in Gallego and Van Ryzin (1994) and assumed that customer arrival followed a time-variant Poisson process that was independent of prices. The distribution of customer valuations was assumed to be known by the seller and was also allowed to vary over time. They considered both continuous pricing (i.e., setting prices continuously throughout the sales window) and periodic pricing (e.g., revising prices every week). They solved the periodic pricing problems using a stochastic dynamic programming approach. Zhao and Zheng (2000) built upon the work of Gallego and Van Ryzin (1994). Similar to Bitran and Mondschein (1997), they also assumed time-variant demand (i.e., both customer arrival and valuations distribution were allowed to vary over time) and developed sufficient conditions under which the findings of Gallego and Van Ryzin (1994) would be valid for such nonhomogenous demand patterns. Other variations of the dynamic pricing problems extended the aforementioned work to incorporate more complicated factors. For example, Smith and Achabal (1998) considered end-of-season liquidation pricing and further took into account the fact that customer demand was dependent not only on time but also on the level of remaining inventory. Bitran et al. (1998) proposed a pricing policy for coordinating clearance sales in retail chains consisting of multiple stores. Craig and Raman (2015) included inventory transfers (i.e., move inventory from one store to another) and the timing of store closings as additional decision variables into the inventory liquidation problem. Finally, Su (2007) and Aviv and Pazgal (2008) considered strategic, forward-looking customers.

More recently, researchers have investigated the problem of inventory liquidation with demand learning, in which the seller has limited information on demand and learns about demand over time according to sales data. For example, several papers have adopted a Bayesian framework to model demand learning, where the seller begins with a certain prior belief on arrival process (e.g., a Poisson process with uncertain rate) and revises the belief as actual sales are observed (Aviv and Pazgal 2002, Araman and Caldentey 2009, Farias and Van Roy 2010). Because the dynamic pricing problem under demand uncertainty generally cannot be solved analytically, these papers have proposed several heuristic pricing policies.

The inventory liquidation problem we address in this paper deals with no replenishment of inventory and myopic customers. First, because we specifically consider the problem of *liquidating* a fixed amount of remaining inventory toward the end of a selling season, replenishment of inventory is not relevant. Second, customers that arrive during a specific pricing period are behaving myopically. In particular, they will purchase the product as long as its price for that period is equal to or below their willingness to pay, and they will not strategically wait until later periods. The myopic behavior of customers represents a realistic aspect of the inventory liquidation problem because, in many settings, consumers have no knowledge of remaining (and changing every day) inventory levels and, therefore, waiting is risky because items might be sold out before the customers come back to make purchases. Regarding time dependency of demand, our proposed solutions can deal with both time-variant and time-invariant demand. Finally, we focus on *periodic* pricing strategies, where the seller sets one price for a given time period and can adjust the prices across periods. This sets us apart from research on continuous pricing strategies (e.g., Araman and Caldentey 2009, Farias and Van Roy 2010). Compared with continuous pricing, periodic pricing is considered to be more practical in reality and incurs less coordination cost (Bitran and Mondschein 1997). Importantly, our proposed approach is flexible in that it can enable periodic pricing for various time granularities, ranging from very short time periods to relatively long periods, owing to its high computational efficiency.

Compared with previous research, our work is different in a number of noteworthy ways. First, instead of trying to solve the liquidation problem directly under stochastic demand (e.g., Bitran and Mondschein 1997, Zhao and Zheng 2000), we propose to first derive a deterministic representation of stochastic demand and then compute the liquidation strategies. Specifically, given knowledge on customer arrival processes and valuation distributions, we derive a set of deterministic representations, whereby (a) customer arrival in each period equals expected arrival, and (b) customer valuations are expected order statistics of the underlying valuation distribution (as will be discussed in more detail in Section 3.1). Such deterministic representation significantly reduces analytical complexity of the problem. More importantly, as will be shown in the paper, it retains a significant level of fidelity in the results; that is, under stochastic demand, the resulting liquidation strategies produce revenues that are very close to the ones obtained by the much more advanced (and much less scalable) algorithms.

Second, even though a dynamic programming formulation has been used in the literature to address similar problems (Bitran and Mondschein 1997, Bitran and Caldentey 2003), our paper moves from a stochastic dynamic programming approach to a much more scalable deterministic version of dynamic programming, by recasting the inventory liquidation problem using deterministic representations of stochastic demand. In addition, we further exploit the problem structure and propose a greedy heuristic approach that is guaranteed to find the optimal liquidation strategy for the deterministic demand representation under reasonable conditions. In the realistic scenarios of stochastic demand, the liquidation strategy produced by our heuristic approach has desirable computational efficiency and revenue performance. Compared with advanced methods, including stochastic dynamic programming (Bitran and Mondschein 1997) and approximate dynamic programming (Farias and Van Roy 2003), our approach is much more scalable and produces very close revenue in expectation. Meanwhile, compared with other simple heuristic pricing strategies (e.g., setting a fixed price or fixed daily sales), our approach consistently generates higher liquidation revenue on average. Moreover, we design a liquidation strategy that *iteratively* uses the greedy heuristic to revise liquidation prices after each period. Compared with using heuristic once to set all liquidation prices in advance, such dynamic strategy performs even better under stochastic demand. Besides its advantages in computational scalability, the proposed heuristic is applicable under various problem settings. When additional factors, such as inventory holding cost or time-dependent arrival (e.g., differing consumer arrival rates for weekdays versus weekends) are taken into consideration, the heuristic approach can be easily adapted. Both the scalability and the flexibility make the greedy heuristic a viable and useful tool to support the decision making of inventory managers.

Overall, following the design science research paradigm, we conduct comprehensive evaluations of our proposed heuristic approach with multiple benchmarks (detailed in Section 6) and demonstrate how it can be used as a decision support tool for inventory management purposes.

3. Theoretical Model of the Inventory Liquidation Problem

In this section, we formally introduce our approach to addressing the inventory liquidation problem. Throughout this paper, we make the following assumptions regarding what is known about the stochastic demand in our liquidation problem setup.

Problem Setup:

1. Customer arrival follows a known stochastic process, denoted by L_t for time period t. Specifically, the number of customers who will visit the store during time period t follows distribution L_t. Customer arrival process is independent of the liquidation strategy and inventory level.
2. Customer valuations follow a known probability distribution, with Cumulative Distribution Function CDF F_t for time period t.
3. Customer valuations decline over time, according to known decay rates $\alpha_t \in (0, 1)$. That is, the CDFs of valuations for time $t+1$ and time t satisfy: $\forall v$, $F_{t+1}(\alpha_t v) = F_t(v)$.

These assumptions represent a realistic liquidation scenario for two reasons. First, the seller's knowledge of arrival process, valuation distribution, and decay rates can be obtained by analyzing historical transaction data.[1] Second, these assumptions are common in the literature. The first assumption states that customers arrive to the store stochastically, according to a known process. We follow the literature in assuming that arrival is independent of inventory status and liquidation strategy, because arrival process is typically determined by customers' regular shopping behaviors (Gallego and Van Ryzin 1994, Bitran and Mondschein 1997, Zhao and Zheng 2000, Aviv and Pazgal 2008, Farias and Van Roy 2010). We allow the arrival process to be time-variant, consistent with prior work such as Bitran and Mondschein (1997) and Zhao and Zheng (2000). In later sections on simulation experiments, for simplicity, we operationalize the arrival process with the most common choice in research literature (e.g., Aviv and Pazgal 2008, Farias and Van Roy 2010), a time-independent Poisson process of rate λ, that is, $L_t = Poisson(\lambda)$. The second assumption states that customer valuations follow a distribution known by the seller. This, again, is a standard assumption in the literature (e.g., Bitran and Mondschein 1997, Aviv and Pazgal 2008, Farias and Van Roy 2010). The third assumption states that, over time, customers that arrive in later days generally have lower valuations than those who arrive in earlier days, because inventory liquidation typically deals with perishable products or products under the pressure of impending obsolescence. A similar assumption is made in Aviv and Pazgal (2008). The declining valuation also marks the key difference between inventory liquidation problem and other dynamic pricing problems (e.g., for airline seats, where the demand characteristics often create upward price pressure over time; McGill and Van Ryzin 1999). In later simulation sections, we typically consider a constant decay rate (i.e., $\alpha_t \equiv \alpha$) for numerical and expositional simplicity, though our solution approach is presented in the general case where decay rates may vary across different days.

3.1. Deterministic Representation of Stochastic Demand

Suppose a seller wants to liquidate N identical items over a period of D days.[2] The seller sets a price for the items on each day, and arriving customers whose valuations are higher than or equal to the price will purchase the items at the stated price. Because the number of items sold on each day does not depend on the order in which customers arrive, for notational convenience we can sort the customers on a specific day according to their valuations in descending order and subsequently label the valuation of the bth customer on the dth day as V_{bd}. This specific labeling of customers implies that $V_{b+1\,d} \le V_{bd}$.

In this paper, we propose a novel approach to overcome the significant analytical and computational complexities that arise when modeling customer demand stochastically, by deriving a deterministic representation of the stochastic demand. The deterministic representation of stochastic demand is derived in three steps. First, we transform the stochastic arrival process into a constant number of average arrivals; that is, we consider B customers come to the store every day ($\forall d \in \{1, \ldots, D\}, B = \mathbb{E}(L_d)$).[3] Second, we transform stochastic valuations of the B arriving customers into the B expected order statistics of the valuation distribution. Although individual customer valuations are random draws from the known distribution, the *expected values* of those valuations are equal to the expected order statistics of that distribution. That is, the expected value of the ith highest valuation among B customers drawn from the same distribution is the ith expected order statistic (in descending order) of that distribution, or more formally, $V_{id} = \mathbb{E}(X_{id})$, where X_{id} is the ith highest order statistic. Expected order statistics provide a precise characterization of the average, representative customer valuations from a known distribution. This also establishes a convenient relationship between price and sales such that, if the price on a specific day is set to the ith highest expected order statistic, then there would naturally be i sales under deterministic demand representation. Third, because of the first two steps, customer valuations across days are characterized fully by the known decay rates α_d, that is, $V_{b\,d+1} = \alpha_d V_{bd}$, where $\alpha_d \in (0, 1)$. These three transformational steps allow us to recast the stochastic inventory liquidation problem in a deterministic, tractable form and facilitate the development and presentation of our computational solutions. The designed liquidation strategies can then be applied and evaluated under realistic, stochastic demand.

With the goal of maximizing total sales revenue, the seller sets the sales price of the item for each day. Thus, the resulting liquidation strategy can be represented as a vector of daily prices $(p_1, p_2, \ldots, p_D)$. Given any daily price, the number of items sold on that day is equal to the number of customers whose valuations are equal to or greater than the price. Therefore, under deterministic representation of customer valuations, an equivalent notation of liquidation strategy is a vector of daily sales quantities $(S_1, S_2, \ldots, S_D)$, where S_d is the number of items sold on the dth day, $S_d = \sum_{b=1}^{B} \boldsymbol{I}(V_{bd} \ge p_d)$ for any d, and $\sum_{d=1}^{D} S_d \le N$. The daily revenue $R(S_d, d)$ obtained by selling S_d items on the dth day is $R(S_d, d) = S_d p_d$. The seller's total revenue is, therefore, $\sum_{d=1}^{D} R(S_d, d)$.

Please note that we assume zero salvage value for any unsold items by the end of the liquidation period, because the case with positive salvage value can be straightforwardly transformed to the case with zero salvage value by shifting the customer valuation distribution leftward with an amount equal to the unit salvage value. Such transformation technique has been used in prior work (e.g., Gallego and Van Ryzin 1994, p. 1004). Therefore, although the presence of positive salvage value, as compared with the case of no salvage value, may result in different optimal total revenue and liquidation strategy, it has only minimal impact on how we solve the liquidation problem.

3.2. Fixed-Price and Fixed-Quantity Strategies for Inventory Liquidation

Two very simple, commonly used liquidation strategies are the fixed-price strategy and fixed-quantity strategy. In a fixed-price strategy, the seller sets a single price, p^*, for the item throughout the entire liquidation period (i.e., $\forall d$, $p_d = p^*$). Note that, if the seller sets a price that equals the Nth valuation of all arriving customers throughout the liquidation period of D days, the strategy achieves allocative efficiency, because customers with top N valuations end up purchasing the items. Fixed-price strategy has been widely adopted in the past, mainly because it incurs very low menu cost (there is no price change) and because companies did not have enough accurate information about customer valuations (Elmaghraby and Keskinocak 2003). However, the fixed-price strategy is seldom optimal. Suppose the corresponding sales quantities vector for allocatively efficient fixed-price strategy is $(S_1, S_2, \ldots, S_D)$; then, by setting discriminant prices every day (i.e., $p_d = V_{S_d d}$), the seller can already acquire higher revenue than by using fixed-price strategy while still selling to the same exact customers with top N valuations. Importantly, even setting the aforementioned discriminant prices every day can be suboptimal, because the optimal strategy may not be selling to customers with the top N valuations at all. We illustrate this point using a numeric example. As shown in Table 1, suppose the seller is trying to liquidate four items over a period of four days, and where customer valuations decay by 5% from day to day. There are four customers coming to the store on each day (i.e., 16 customers in total over four days), whose valuations are summarized in the table. Because the four highest valuations are 100, 96, 95, and 92, a fixed-price strategy would set the liquidation price to be 92. The resulting revenue is, therefore, 368. Alternatively, if the seller sets discriminant prices on each day while still selling to the top four customers, the daily prices would be 92 and 95 for the first two days. The resulting revenue is 371. However, the optimal strategy in this case is actually to sell two items on the first day (at price of 96), one item on the second day (at price 95), and one on the third day (at price 90.25). The optimal strategy results in the total revenue of 377.25.

Under a fixed-quantity strategy, the seller liquidates the same number of items every day. A seller could use this strategy when there is a fixed capacity to store or transport the items. Note that fixed-quantity strategy is equivalent to a periodic pricing

Table 1. Numeric Example Showing the Suboptimality of Fixed-Price and Fixed-Quantity Strategies for Inventory Liquidation

V_{bd}	Day $d = 1$	Day $d = 2$	Day $d = 3$	Day $d = 4$
V_{1d}	100	95	90.25	85.74
V_{2d}	96	91.2	86.64	82.31
V_{3d}	92	87.4	83.03	78.88
V_{4d}	88	83.6	79.42	75.45

Note. V_{bd} represents the valuation of bth customer on dth day.

strategy with fixed discount, as described by Bitran and Mondschein (1997), if the initial price is associated with sales of the fixed quantity on the first day and the fixed discount is equal to a constant valuation decay rate. The fixed-quantity strategy is also suboptimal in general because, in liquidation scenarios, customer valuations are decreasing over time, and it is advantageous to sell more items earlier in the liquidation period rather than later. Again, this can be easily observed from the same example in Table 1. If the seller instead adopts a fixed-quantity strategy to sell four items over four days, then one item will be sold to the buyer with highest valuation on each of the four days, resulting in a total revenue of 370.99, which is less than the optimal total revenue of 377.25.

3.3. A Dynamic Programming Approach

Because of the specifics of the problem structure, an optimal solution to the revenue-maximizing liquidation problem can be readily found computationally, using a dynamic programming approach. Let $MAXREV(n,d)$ represent an intermediate problem (i.e., calculating the maximum revenue that a retailer can obtain from selling n items starting on dth day of the liquidation period). The recurrence equation of this problem is

$$MAXREV(n,d) = \max_{0 \le j \le \min(B,n)} \{ j \times V_{jd} + MAXREV(n-j, d+1)\},$$

where the calculation of $MAXREV(n,d)$ is decomposed into the possible revenue of selling exactly j items on day d (represented by $j \times V_{jd}$, where V_{jd} is the jth highest valuation on day d, equal to the price at which the seller would choose to sell the j items) and the rest of the items later, starting on the next day $d+1$ (represented by $MAXREV(n-j, d+1)$). Subsequently, $MAXREV(n,d)$ is determined by finding the maximum intermediate revenue, subject to the constraint that number of items sold on day d cannot be larger than the number of daily arriving customers B or the number of remaining items n to be sold (represented by $0 \le j \le \min(B,n)$). To constrain the liquidation within the desired period of D days, items that are not sold by the last day will have no value to the retailer; that is, $\forall n, MAXREV(n, D+1) = 0$. The maximum revenue of the original inventory liquidation problem can, therefore, be found by computing $MAXREV(N,1)$, and the optimal liquidation strategy can be efficiently found by solving all the intermediate dynamic programming subproblems backward. The worst-case computational complexity of this approach is $O(DNB)$, because there are at most DN intermediate subproblems in total, each of which takes $O(B)$ to solve, assuming B to be smaller than N in large-scale liquidation scenarios.

4. A Greedy Heuristic Approach

In this section, we propose a greedy heuristic approach to solve the inventory liquidation problem under deterministic demand representation, which is computationally faster than the dynamic programming approach. We also prove that, under a broad set of realistic conditions (i.e., deterministic representations of commonly used valuation distributions), the heuristic always finds the optimal solution.

4.1. Description of the Greedy Heuristic

In the previous section, we described the seller's decision problem as setting price (or, equivalently, quantity) of items to be liquidated every day, to maximize total revenue given the known information about customer valuations. An alternative way to think about the liquidation problem is to view it as a stepwise assignment problem. The seller can incrementally (i.e., in a greedy fashion) *assign* each item to be liquidated on a specific day, so that the total revenue is maximized. More specifically, for the first item to be liquidated, the seller could potentially assign it to be sold on any one of the D possible days, setting the sales price for that day to be the highest valuation on that day. To maximize revenue, the seller will assign it to be sold on the first day, because V_{11} is the highest valuation overall.[4] For the second item, the seller can again assign it to be sold on the first day (by setting the sales price for this day to be the second-highest valuation on that day) or on any other day. Again, because of the valuation decay, to maximize revenue the seller will choose between selling two items on the first day and selling one item on each of the first two days by comparing the corresponding revenues, $2V_{21}$ versus $(V_{11} + V_{12})$. The same process can be used until all items are assigned. It is a greedy heuristic approach because the seller only considers how to assign the next item to a specific day (i.e., not changing any of the previous assignments) and greedily tries to maximize total revenue at each assignment step. In fact, greedily maximizing total revenue at each assignment step is equivalent to maximizing the *marginal* revenue brought by the item under consideration. For example, when assigning the second item, comparing the total revenue $2V_{21}$ versus $(V_{11} + V_{12})$ is equivalent to comparing the marginal revenue $2V_{21} - V_{11}$ versus $(V_{11} + V_{12}) - V_{11}$, where V_{11} is the total revenue after assigning the first item. Therefore, it is useful to introduce the concept of daily marginal revenue (henceforth referred to as DMR). DMR is defined as the change in daily revenue by assigning one more item to be sold on a specific day (without changing any previous assignments). The high-level pseudocode of this heuristic approach is presented in Algorithm 1.

The straightforward implementation of Algorithm 1 has computational complexity of $O(DN)$, because there are N items in total, each of which can potentially be assigned to D days, and the DMR for each possible assignment only needs to be precomputed once, with $O(DB)$ complexity.[5] In practice, this implementation can be further improved. For example, the process of finding BestMarginalRevenue to determine the assignment for each item can be improved by using a priority queue data structure. The elements of the queue are indexes representing days, and the priority associated with each element is the DMR that would be obtained if the item under consideration is assigned to the corresponding day. Using typical implementation of priority queue (e.g., heap implementation), the computational complexity of the greedy heuristic can be reduced to $O(Nlog(D))$. Compared with the dynamic programming approach, which has computational complexity of $O(DNB)$, the greedy heuristic can facilitate significant computational speed-up. Such speed-up can be crucial, especially in real-time liquidation scenarios (e.g., when the minimum time unit of pricing is as short as a minute, instead of a day).

4.2. Sufficient Condition for the Optimality of Greedy Heuristic

In addition to the computational scalability benefit, the greedy heuristic approach is guaranteed to find the optimal solution to the deterministic representation of the liquidation problem under certain conditions. To demonstrate this, denote $r(S, d)$ as the DMR of assigning one more item to the dth day, given that S items are already assigned to be sold on that day, that is, $r(S,d) \stackrel{\text{def}}{=} R(S+1, d) - R(S, d)$.

Proposition 1. *For all $d \in \{1, 2, \ldots, D\}$, if $r(S,d)$ is a nonincreasing function with respect to S (henceforth referred to as the property of nonincreasing daily marginal revenue, or NDMR property), then the greedy heuristic approach is guaranteed to find the optimal liquidation strategy.*

Proof of Proposition 1 is included in Appendix 1 in the online supplement. Note that NDMR is a sufficient but not necessary condition for the optimality of the heuristic approach. The following numeric example in Table 2(a) shows that, even if NDMR is violated, the heuristic may still be able to find the optimal solution in some cases. Consider a simple liquidation problem, in which a seller is trying to liquidate three items over three days. Three customers come to the store on each day, and their valuations decay by 10% from day to day. This setup is summarized in Table 2(a). In this case, selling one item on day 1 yields revenue of 100, selling two items yields revenue of 120 (i.e., 2×60), and selling three items yields revenue of 150 (i.e., 3×50). Therefore, NDMR is violated for day 1 ($150 - 120 > 120 - 100$). Similarly, it is easy to see that, in this example, NDMR is violated for days 2 and 3 as well. Nonetheless, the heuristic in this case would still find the optimal strategy, which is

Algorithm 1 (Pseudocode Sketch for Greedy Heuristic)

```
CurrentTotalRevenue = 0                     // initializing: nothing sold so far
Sales[1..D] = 0                             // initializing: array with the number of items sold on each day
for items from 1 to N:                      // assigning each individual item
    AssignmentFound = FALSE                 // initializing: an indicator of whether assignment is successful
    BestMarginalRevenue = 0                 // initializing: best marginal revenue from selling this item
    for AssignDay from 1 to D:                      // try every possible day
        if Sales[AssignDay] < B:            // it is feasible to sell this item on AssignDay
            assign one more item to AssignDay       // assign this item temporarily to specific day
            calculate DMR                   // check marginal revenue of selling this item on AssignDay
            if (DMR > BestMarginalRevenue):         // if marginal revenue is best so far
                BestDay = AssignDay                 // record best day to assign
                BestMarginalRevenue = DMR           // record best marginal revenue
                AssignmentFound = TRUE              // set assignment indicator to be successful
    if AssignmentFound == TRUE:
        Sales[BestDay] += 1                         // assign new item permanently to best day
        CurrentTotalRevenue += BestMarginalRevenue  // add best marginal revenue to total revenue
    else:
        break              // no assignment increases revenue, terminate
return Sales[1..D], CurrentTotalRevenue     // return sales on each day (i.e., sales strategy) and total revenue
```

Table 2. Numeric Example (a) Showing NDMR Is Not a Necessary Condition for Heuristic Optimality and (b) Where Heuristic Does Not Find Optimal Liquidation Strategy

	Day $d = 1$	Day $d = 2$	Day $d = 3$
Panel (a)			
V_{1d}	100	90	81
V_{2d}	60	54	48.6
V_{3d}	50	45	40.5
Panel (b)			
V_{1d}	80	8	0.8
V_{2d}	40	4	0.4
V_{3d}	30	3	0.3

Note. V_{bd} represents the valuation of bth customer on dth day.

to sell one item to the first customer on each of the three days. It is worth noting that the heuristic does not *always* find optimal liquidation strategy. The numeric example in Table 2(b) provides an illustration. Three customers come to the store on each day, and their valuations decay by 90% from day to day. In this case, the heuristic would find a strategy to sell one item on each of the three days, at prices 80, 8, and 0.8, respectively, resulting in revenue of 88.8. However, the optimal strategy is actually to sell all three items on the first day, at price 30, which yields optimal revenue of 90.

Next, we give the necessary and sufficient condition for customer valuations to satisfy the NDMR property. Let $\Delta(b, d)$ denote the difference between the valuations of the bth and $(b+1)$th customers on day d, that is, $\Delta(b, d) = V_{bd} - V_{b+1\,d}$.

Proposition 2. *The NDMR property is satisfied if and only if* $\frac{\Delta(b,d)}{\Delta(b+1,d)} \le \frac{b+2}{b}$ *for all* $b \in \{1, \ldots, B-2\}$ *and for all* $d \in \{1, \ldots, D\}$.

Proof of Proposition 2 is included in Online Appendix 1. An important benefit of this proposition is that we can directly evaluate whether the NDMR property is satisfied for the deterministic representation of any given customer valuation distribution, either analytically or numerically. In the following Corollary 1, we show that the NDMR property is satisfied if V_{bd} follows a uniform distribution, an exponential distribution, or a Weibull distribution with shape parameter larger than 1. These distributions are commonly used to model customer valuations in prior work. For example, uniform distribution was used in Aviv and Pazgal (2002) and Araman and Caldentey (2009); exponential distribution was used in Gallego and Van Ryzin (1994), Smith and Achabal (1998), Araman and Caldentey (2009), and Farias and Van Roy (2010); Weibull distribution was used in Bitran and Mondschein (1997) and Bitran and Caldentey (2003).

Corollary 1. *The NDMR property is satisfied if the underlying valuation distribution is uniform, exponential, or Weibull (with shape parameter larger than* 1*).*

Proof of Corollary 1 is included in Online Appendix 1 of the online supplement. More generally, Pearson (1902) shows that, for a random variable X that follows a known distribution with cumulative distribution function $F(x)$, the expected gap between two consecutive order statistics can be calculated as

$$E(X_{(k)} - X_{(k+1)}) = (n!)/((n-k)!k!) \int_{-\infty}^{+\infty} F(x)^{n-k}[1 - F(x)]^k dx.$$

Therefore, for any distribution, one can determine whether it satisfies the condition in Proposition 1, as long as the above integral can be evaluated.

4.3. Incorporating Inventory Holding Cost

Another advantage of the greedy heuristic approach is that it is still effective in finding the optimal solution when taking into account inventory holding cost. Suppose holding one unit of item for one day incurs a cost of h; then each item sold on the dth day has been held in the inventory for $(d-1)$ days, incurring the cost of $h(d-1)$. Therefore, as compared with the situation of no holding cost, the daily revenue with holding cost can be expressed as $R(S, d, h) = R(S, d) - h(d-1)S$ and the daily marginal revenue as $r(S, d, h) = r(S, d) - h(d-1)$. Because of this relationship, if the original customer valuation distribution (without holding cost) satisfies NDMR property (such as the case of uniform distribution of valuations), the revenue structure after we incorporate holding cost will also automatically satisfy NDMR, which guarantees the optimality of the heuristic approach.

4.4. Accounting for Time-Variant Arrival Process

Many real-world inventory liquidation scenarios are characterized with time-variant arrival processes. For example, in some liquidation settings, there may be more customer arrivals on weekends than on weekdays. Although we present our theoretical results with time-invariant arrival for notational simplicity, our heuristic approach can be readily extended to the case of time-variant arrival. As an illustration, suppose daily arrival on a weekday follows $L_{weekday} = Poisson(\lambda_1)$, and daily arrival on a weekend follows $L_{weekend} = Poisson(\lambda_2)$, and that $\lambda_1 \neq \lambda_2$. Following our solution strategy of deriving a deterministic representation of stochastic demand, we can transform the stochastic, time-variant arrival process into its simpler, deterministic representation, such that B_1 customers arrive on each weekday and B_2 customers arrive on each weekend, where $B_1 = \mathbb{E}(L_{weekday}) = \lambda_1$ and $B_2 = \mathbb{E}(L_{weekend}) = \lambda_2$. Accordingly, the deterministic representations of customer valuations on a weekday (weekend) are the B_1 (B_2) expected order statistics of the underlying customer valuation distributions. As long as the valuation distribution satisfies the NDMR property, our proposed heuristic will be able to find the optimal liquidation strategy, regardless of differing numbers of daily arrivals. In other words, our heuristic approach is flexible and can deal with both time-invariant and time-variant arrival processes.

5. Properties of Optimal Liquidation Strategy Under Deterministic Demand Representation

In this section, we take a deeper look into the optimal liquidation strategy under the deterministic representation of stochastic demand. Besides the underlying demand features, the inventory liquidation problem is typically characterized by several distinct factors, including the following: size of inventory (N), length of liquidation period (D), number of daily arriving customers (B), valuation decay rate (α), and the potential presence of inventory holding cost (h). In our theoretical models, we take these factors as given and fixed. In reality, however, an inventory liquidation manager often may need to make decisions that are related to these factors. For example, given a fixed amount of inventory to be liquidated, a manager may need to determine the best length of liquidation period and whether to conduct marketing promotions or advertising, which could indirectly affect the number of daily arriving customers and the valuation decay rates.[6] To make informed decisions, it is therefore beneficial to understand the relationships between these key factors and the resulting optimal revenue as well as the liquidation strategy. In this section, we first conduct a series of simulation experiments to demonstrate how insights on such relationships can be obtained conveniently and efficiently by solving the liquidation problem using the proposed heuristic. We also discuss the managerial implications for retailers that can be derived from each set of simulations. Then we point out that the same insights about optimal liquidation strategy and revenue under deterministic representation of demand, which we obtain using simulations, can in fact be proven theoretically.

5.1. Simulation Experiments

We run multiple sets of simulations to demonstrate how different configurations of the liquidation factors lead to different optimal liquidation strategies and total revenues under deterministic demand representation. For illustration purposes, we present simulation results based on the following basic configuration of inventory liquidation problem. Suppose the seller is liquidating 100 items ($N = 100$) over a period of 10 days ($D = 10$). There are 100 customers coming to the store every day ($B = 100$), whose valuations for the item are expected order statistics of a uniform distribution. On the first day, valuations are B expected order statistics of the standard uniform $U(0, 1)$. Additionally, we choose a constant valuation decay rate (α) of 0.99 (i.e., $\forall b, d,\ \ V_{b\,d+1} = 0.99V_{bd}$). In other words, for the items that need to be liquidated, valuations keep decreasing by 1% as compared with the previous day. Applying the heuristic approach on this basic configuration, which is guaranteed to find the optimal liquidation strategy because uniformly distributed valuations satisfy the NDMR property, the optimal revenue from liquidation is 86.27.

We conduct five sets of simulations, each of which varies a particular liquidation factor, while holding the other configurations the same as in the aforementioned basic setup. In each set of simulations, we use the heuristic approach to find the optimal liquidation strategies (represented as the number of items to be sold on each day) as well as the optimal revenues *under deterministic representation of the demand*. These liquidation strategies and revenues are depicted in Figures 1–5. Below,

Figure 1. Optimal Strategies for Three Levels of Inventory Under Deterministic Representation of Stochastic Demand (a) and Optimal Revenues for All Levels of Inventory Under Deterministic Representation of Stochastic Demand (b)

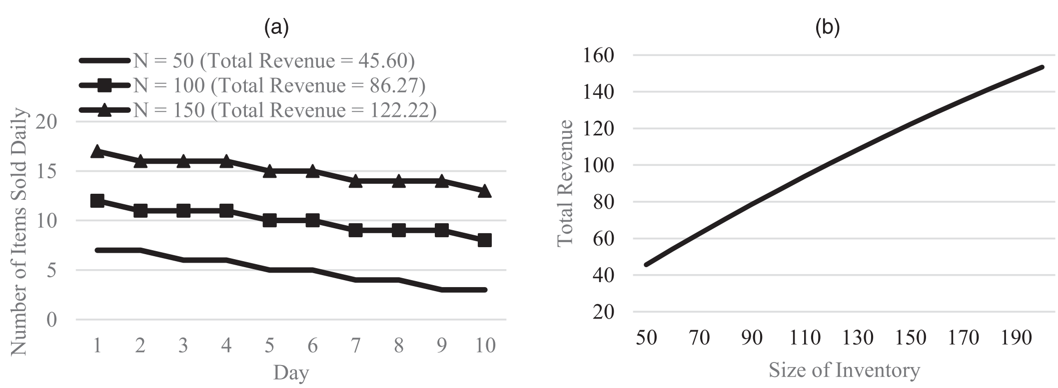
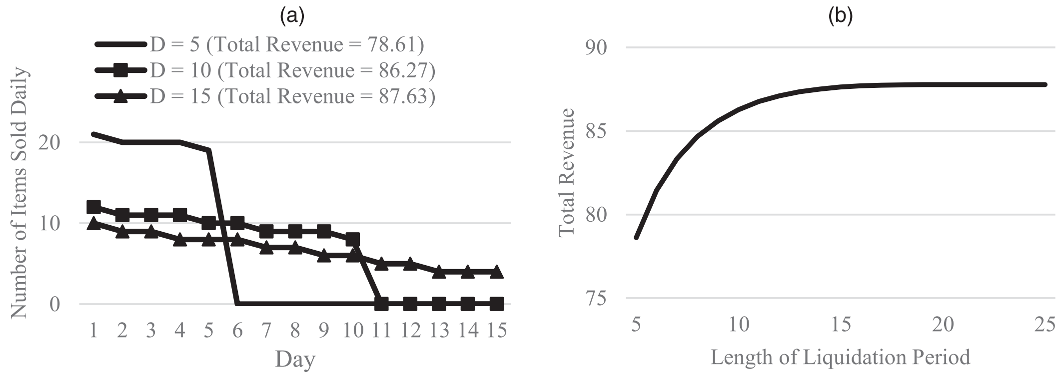


Figure 2. Optimal Strategies for Three Liquidation Periods Under Deterministic Representation of Stochastic Demand (a) and Optimal Revenues for All Lengths of Liquidation Periods Under Deterministic Representation of Stochastic Demand (b)



we describe each set of simulations and the corresponding results in detail.

Figure 1, (a) and (b), shows how optimal liquidation strategy and revenue change with respect to inventory level (N). We vary the size of inventory from 50 units to 200 units with an increment of 10 units and report liquidation strategies for three particular inventory levels: 50, 100, and 150. According to Figure 1(a), as the size of inventory becomes larger, the daily sales in optimal strategy shift upward almost proportionally. However, the optimal revenue has a concave relationship with inventory size, as demonstrated in Figure 1(b). In other words, given a fixed time window for liquidation, having more items to be sold means that the seller has to price the item cheaper on average.

Figure 2, (a) and (b), shows how optimal liquidation strategy and revenue change with respect to the length of liquidation period (D). We vary the liquidation period from 5 days to 25 days with an increment of 1 day and report liquidation strategies for three particular liquidation periods: 5, 10, and 15. Intuitively, the manager tends to sell more items on each day as the liquidation period becomes shorter, consistent with Figure 2(a). However, a shorter liquidation period is also associated with a lower optimal revenue, as compared with a longer liquidation period. On the other hand, having a longer liquidation period does not always produce higher total revenue. As shown in Figure 2(b), in our experiment settings, liquidation periods that are longer than 20 days are all associated with the same amount of total revenue, because customers arriving in the last few days have such decayed valuations that make them not as profitable as customers arriving in earlier days. This set of simulation indicates that, given a fixed size of inventory, there exists an optimal liquidation window (in this case, the optimal window is 20 days). Although faster liquidation may harm the total revenue, slower liquidation will not further increase the revenue.

Figure 3, (a) and (b), shows how optimal liquidation strategy and revenue change with respect to number of daily arriving customers (B). We vary the number of daily arriving customers from 50 to 150 with an increment of 10 and report liquidation strategies for three particular numbers of daily customers: 50, 100, and 150. As shown in the figures, the seller tends to sell more items earlier rather than later when there are more daily arriving customers, and the total revenue is also higher. This is because higher store traffic is

Figure 3. Optimal Strategies for Three Levels of Daily Arrivals Under Deterministic Representation of Stochastic Demand (a) and Optimal Revenues for All Levels of Daily Arrivals Under Deterministic Representation of Stochastic Demand (b)

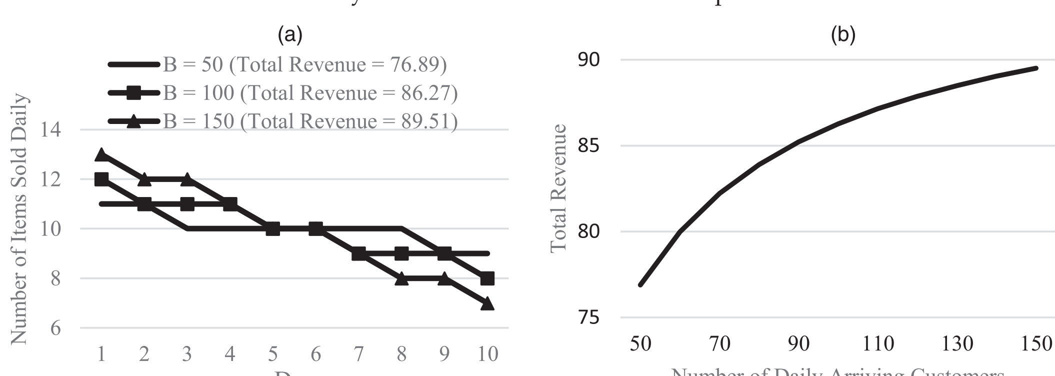
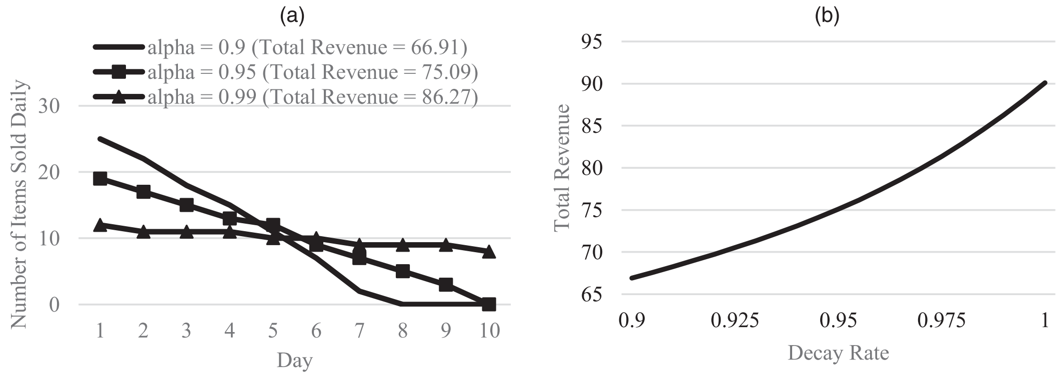


Figure 4. Optimal Strategies for Three Levels of Decay Rates Under Deterministic Representation of Stochastic Demand (a) and Optimal Revenues for All Levels of Decay Rates Under Deterministic Representation of Stochastic Demand (b)

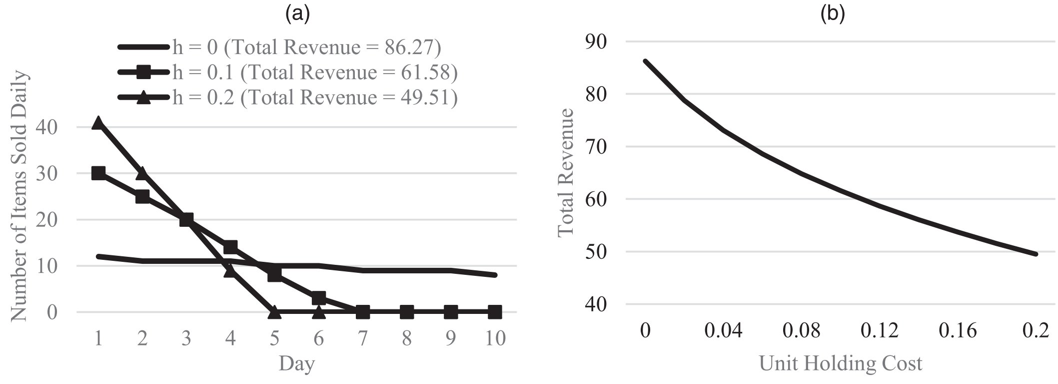


associated with more customers with higher valuations. By selling to these high-valuation customers, the seller is able to get more revenue than in the case with lower store traffic. Notably, as Figure 3(b) indicates, the increase in optimal total revenue tends to taper off as the number of daily arriving customers keeps increasing. This nonlinear relationship has a clear managerial implication to liquidation managers. Although it is possible to attract more customers into the store by conducting marketing promotions, the seller needs to balance the growth in liquidation revenue (due to increased traffic) with the potential costs of marketing promotions.

Figure 4, (a) and (b), shows how optimal liquidation strategy and revenue change with respect to valuation decay rate (α). We vary the decay rate from 0.90 to 1.00 with an increment of 0.005 and report liquidation strategies for three particular decay rates: 0.90, 0.95, and 0.99. We can see from Figure 4(a) that, as valuation decay becomes faster (i.e., with a smaller decay rate α), the manager tends to sell more items in earlier days in this example setting. Accordingly, optimal revenue is lower for greater valuation decay than for smaller decay. Interestingly, Figure 4(b) shows a convex relationship between optimal total revenue and decay rate. As decay rate becomes larger (i.e., less rapid valuation decay), total revenue grows at an increasing speed. For liquidation managers, if they can (indirectly) affect valuation decay rates effectively through marketing activities, then there will be considerable revenue margin available for mitigating the valuation decay.

Figure 5, (a) and (b), shows how optimal liquidation strategy and revenue change with respect to unit holding cost (h) associated with holding one item in the inventory for one day. We vary the unit holding cost from 0 to 0.2 with an increment of 0.02 and report liquidations strategies for three particular levels of holding cost: 0, 0.1, and 0.2. Introducing the holding cost changes the optimal liquidation strategy because the seller will be inclined to sell more items in earlier days to avoid holding them in inventory for a longer time, as shown in Figure 5(a). According to Figure 5(b), as the unit holding cost becomes higher, the optimal total revenue follows a convex decline. This can be understood intuitively, because a higher holding cost pushes the seller to liquidate more items in earlier days and consequently leads to a significant revenue

Figure 5. Optimal Strategies for Three Levels of Unit Holding Cost Under Deterministic Representation of Stochastic Demand (a) and Optimal Revenues for All Levels of Unit Holding Cost Under Deterministic Representation of Stochastic Demand (b)



drop not only due to holding cost, but also because of the NDMR property of customer valuations.

In summary, the set of simulation experiments presented in this section delineate the relationships between key factors of the inventory liquidation problem and the resultant optimal revenue and optimal liquidation strategies under deterministic demand representation. The proposed greedy heuristic, because of its computational efficiency, makes for a flexible and informative decision support tool that a liquidation manager can use to understand the implications of a wide variety of decisions in their specific settings.

5.2. Theoretical Properties of Optimal Liquidation Strategy and Revenue

We are able to theoretically prove the relationships between the liquidation factors and optimal liquidation revenue under deterministic demand representation, which we obtained using simulations in the previous section. The following Proposition 3 summarizes the monotonicity- and concavity-related properties of total revenue with respect to different liquidation factors.

Proposition 3. *Under deterministic representation of stochastic demand, if NDMR property is satisfied, then the total revenue obtained by optimal liquidation strategy has the following properties:*

1. *total revenue increases concavely as inventory size (N) increases;*
2. *total revenue increases concavely as the length of liquidation period (D) increases, and stops increasing beyond a certain length;*
3. *total revenue increases as number of daily arrivals (B) increases;*
4. *total revenue increases convexly as daily decay rate (α) increases;*
5. *total revenue decreases convexly as unit holding cost (h) increases.*

Proof of Proposition 3 is included in Online Appendix 1. The first two properties are consistent with Gallego and Van Ryzin (1994), which found similar patterns in a different liquidation problem context. The concavity relationship between optimal revenue and the number of daily arrivals (B) is affected by how values of expected order statistics change with respect to B and, therefore, is likely to be dependent on valuation distributions. Although we have empirically observed that total revenue increases *concavely* as number of daily arrivals increases for uniform, exponential, and Weibull valuation distributions, the general theoretical result regarding concavity is not available. Finally, note that all of these theoretical results have been adequately captured in our previous simulations. In other words, liquidation managers can obtain a diverse set of insights regarding their specific liquidation problems *empirically*, using the proposed heuristic approach as an efficient decision support tool.

6. Comprehensive Performance Evaluation Under Stochastic Demand

In this section we evaluate the performance of the proposed heuristic approach under the realistic conditions of stochastic customer valuations and arrival. In other words, throughout this section, customer valuations and daily arrivals are random draws from known underlying distributions—a standard assumption in inventory liquidation literature (e.g., Bitran and Mondschein 1997, Elmaghraby and Keskinocak 2003). We compare the revenue performance of the heuristic approach against two sets of benchmarks, including (1) several simple and computationally efficient liquidation strategies, such as the fixed-price strategy and the fixed-quantity strategy as well as their variations, all of which are based on the solution strategy using deterministic demand representation; and (2) advanced algorithms that attempt to solve for optimal liquidation strategies directly under stochastic demand, including stochastic dynamic programming (Bitran and Mondschein 1997) and approximate dynamic programming (Farias and Van Roy 2003).

6.1. Performance Relative to Other Scalable Liquidation Strategies

As discussed in Section 3.2, two common examples of simple and computationally efficient liquidation strategies are the fixed-price strategy and fixed-quantity strategy. Under deterministic representation of stochastic demand, we have shown that fixed-price/-quantity strategies are not guaranteed to be optimal. In this section, we compare the performance of our proposed greedy heuristic against these two strategies and their variations, directly under stochastic demand.

In reality, when faced with stochastic demand, the seller has potentially two ways of designing liquidation strategies. As one option, the seller can choose to ignore the randomness in actual demand and still design the liquidation strategy for the entire liquidation period ahead of time. Once the strategy is determined, the retailer keeps the daily item prices as planned throughout the entire liquidation period and sells the items to actual arriving customers whose valuations are equal to or above the daily prices. For the fixed-price strategy (FP), the seller can set the fixed price to be the Nth expected order statistic across the entire liquidation horizon. For the fixed-quantity strategy (FQ), the seller can set the price on day d to be the Qth expected order statistics on that day, where $Q = \lfloor N/D \rfloor$ is the fixed quantity to be sold on each day. Similarly, the seller can apply the proposed greedy heuristic on the basis of the expected daily customer arrivals and the expected order statistics of the valuation distribution and obtain the liquidation price for each day. We refer to this strategy as the *plan ahead strategy* (PAS).

Alternatively, the seller can *dynamically* adjust the prices from day to day and thereby account for some degree of demand stochasticity. More specifically, we discuss three such strategies. First, we consider a dynamic-price strategy (DP), a variation of the fixed-price strategy. The seller sets price on day d to be the M_dth expected order statistic across the *remaining* liquidation horizon, where M_d is the amount of remaining inventory at the beginning of day d. Second, we consider a dynamic-quantity strategy (DQ), a variation of the fixed-quantity strategy. The seller sets price on day d to be the Q_dth expected order statistic on day d, where $Q_d = \lfloor M_d/(D-d+1)\rfloor$ is the average daily quantity that needs to be sold (i.e., the total remaining quantity divided by the number of remaining days).[7] Third, we consider a similar "dynamic" variation of the proposed greedy heuristic. At the beginning of each day d, the seller can use the same greedy heuristic to solve a new liquidation problem where M_d is the number of remaining items and $D-d+1$ is the number of remaining days. The solution to this problem gives the item price for the current day. This strategy is referred to as the *plan dynamically strategy* (PDS).

We run two sets of comprehensive simulation experiments to benchmark the performances of PAS against FP and FQ, and PDS against DP and DQ. Whereas PAS, FP, and FQ represent strategies without adjustable daily prices, PDS, DP, and DQ represent strategies with adjustable daily prices. In the first set of simulations, the daily number of arrivals is randomly drawn from *Poisson*(300). Customer valuations on the first day are randomly drawn a uniform distribution $U[0,1]$, an exponential distribution *Exp*(1), or a Weibull distribution with scale parameter $\lambda = 1$ and shape parameter $\rho = 4$ (similar to the configuration in Bitran and Mondschein 1997).[8] Accordingly, customer valuations on later days are independently drawn from the same distributions, then multiplied by the decay rates (similar to the practice in Aviv and Pazgal 2008). We also vary several key configurations, across different simulations. Specifically, for each valuation distribution, we simulate a basic configuration with ($N = 5000, D = 100$, $\alpha = 0.99$) and then vary inventory size, liquidation period length, and decay rate, respectively, in three separate simulations: (1) holding $D = 100$, $\alpha = 0.99$, and varying $N \in \{2{,}500, 3{,}000, 3{,}500, 4{,}000, 4{,}500, 5{,}000\}$; (2) holding $N = 5{,}000$, $\alpha = 0.99$, and varying $D \in \{50, 60, 70, 80, 90, 100\}$; and (3) holding $N = 5{,}000$, $D = 100$, and varying $\alpha \in \{0.99, 0.98, 0.96, 0.94, 0.92, 0.9\}$. Note that, because of the computational efficiency of these liquidation strategies, we are able to simulate relatively large liquidation problems. Across all problem configurations presented here, solving for each liquidation strategy takes less than 1 second.

For each problem configuration, we randomly generated 1,000 demand instances (i.e., 1,000 sets of daily arrivals and customer valuations). Under each demand instance, we applied each of the aforementioned strategies to calculate the corresponding revenue. All simulations are implemented in Python and performed on a 3.6-GHz Intel Core i7 computer with 8GB RAM. Because of space restrictions, we report a representative subset of results in Table 3. We include the complete set of simulation results in the online supplement (Appendix 4) in both table and figure formats.

Table 3. Performance Benchmarking with Other Scalable Liquidation Strategies (Poisson Arrival)

				Without adjustable daily prices			With adjustable daily prices		
	N	D	α	PAS	FP	FQ	PDS	DP	DQ
Valuations ~ $U[0,1]$	5,000	100	0.99	**2,800.25** (26.57)	2,637.69 (14.42)	2,630.51 (30.89)	**2,812.14** (11.96)	2,647.35 (8.33)	2,639.87 (15.55)
	2,500	100	0.99	**1,657.11** (25.12)	1,603.32 (20.03)	1,438.79 (21.91)	**1,675.51** (7.75)	1,617.01 (5.63)	1,448.65 (10.19)
	5,000	50	0.99	**2,631.71** (23.22)	2,567.49 (16.86)	2,625.96 (24.68)	**2,640.24** (17.40)	2,577.40 (12.07)	2,633.30 (19.02)
	5,000	100	0.9	**733.42** (13.64)	356.75 (3.13)	416.99 (13.85)	**733.86** (13.31)	357.95 (5.68)	417.03 (13.37)
Valuations ~ *Exp*(1)	5,000	100	0.99	**5,743.37** (64.94)	5,611.81 (41.24)	5,666.91 (66.20)	**5,770.37** (48.11)	5,643.51 (39.41)	5,684.55 (51.23)
	2,500	100	0.99	**4,022.99** (73.04)	3,986.54 (49.84)	3,884.11 (76.46)	**4,073.75** (35.65)	4,017.35 (28.41)	3,923.21 (47.01)
	5,000	50	0.99	**4,323.40** (45.28)	4,276.76 (35.28)	4,322.50 (45.47)	**4,331.72** (53.16)	4,292.21 (46.48)	4,329.84 (54.48)
	5,000	100	0.9	**1,099.19** (23.64)	566.19 (5.13)	892.81 (28.84)	**1,099.43** (23.50)	568.95 (9.54)	893.08 (28.12)
Valuations ~ *Weibull*(4)	5,000	100	0.99	**4,122.02** (46.13)	4,024.73 (33.46)	3,642.98 (49.74)	**4,149.38** (15.11)	4,045.31 (10.37)	3,662.86 (22.59)
	2,500	100	0.99	**2,338.21** (39.89)	2,325.25 (25.64)	1,952.21 (36.74)	**2,371.58** (8.71)	2,343.87 (6.99)	1,979.06 (14.63)
	5,000	50	0.99	**4,079.55** (41.62)	4,003.16 (32.94)	4,018.34 (45.78)	**4,105.31** (16.44)	4,024.50 (10.80)	4,039.63 (20.82)
	5,000	100	0.9	**1,507.63** (22.44)	776.49 (8.11)	578.16 (18.01)	**1,510.16** (20.39)	778.34 (6.98)	578.30 (17.09)

Notes. Standard deviation of liquidation revenue is included in parentheses. The highest revenue in each set of simulations is shown in bold. See complete results in the online supplement (Appendix 4).

We make several important observations based on results in Table 3. First, among strategies with and without adjustable daily prices, respectively, heuristic-based strategies (i.e., PAS and PDS) consistently outperform the alternative price-based or quantity-based strategies. According to the complete results in Online Appendix 4, across all configurations, PAS is associated with 0.6%–106% higher average revenue than FP and 0.02%–161% higher average revenue than FQ; PDS is associated with 0.9%–105% higher average revenue than DP and 0.04%–161% higher average revenue than DQ. Notably, PAS/PDS seem to be especially advantageous when the valuations decay faster (i.e., when α is smaller). This is a valuable advantage, because many real-world liquidation settings are characterized by moderately fast valuation decay.[9] Therefore, PAS and PDS are likely to outperform price-based and quantity-based strategies by a significant margin in practical scenarios. Second, strategies that adjust daily prices generally perform better than strategies that set all prices ahead of time. For example, PDS is associated with higher revenue than PAS in every simulation configuration. This is intuitive because strategies with adjustable prices can take into account the realized sales in previous days when setting next-day prices. PDS generally also has smaller standard deviation of revenue than PAS, indicating better stability. In addition, we also observe via these simulations that all strategies are able to sell more than 99% of inventory on average and that strategies with adjustable daily prices are able to sell even more items on average than strategies without adjustable daily prices.

In the second set of simulations, the daily number of arrivals is randomly drawn from a uniform distribution, $U[150, 450]$. Compared with the Poisson arrival used in the first set of simulations, this arrival setup has the same mean (i.e., average arrival is still 300) but substantially higher variance and therefore simulates the scenario of highly volatile demand. Other configurations are the same as before. Similarly, we report a representative subset of results in Table 4 and include the complete set of results in the online supplement (Appendix 5) in both table and figure formats.

According to Table 4, we can see that, despite significant demand stochasticity, we still observe largely the same relationships among different liquidation strategies. In particular, heuristic strategies consistently outperform corresponding price-/quantity-based strategies, and strategies that adjust daily prices generally perform better than strategies that set all prices ahead of time. As expected, the standard deviations of revenue are generally higher than in the case of Poisson arrival, indicating more variation due to high demand volatility.

These simulations highlight the unique advantage of the proposed heuristic approach over other computationally efficient approaches. PAS and PDS are able to consistently produce higher average revenue than FP/FQ and DP/DQ strategies. Allowing daily price adjustments further enhances the performance of heuristic. Specifically, PDS generates significantly higher average revenue than PAS, while also having smaller revenue standard deviation (i.e., providing more stable output). Overall, PDS represents a practical, effective, and highly efficient way of applying the proposed heuristic approach under stochastic demand.

Table 4. Performance Benchmarking with Other Scalable Liquidation Strategies (Uniform Arrival)

				Without adjustable daily prices			With adjustable daily prices		
	N	*D*	α	PAS	FP	FQ	PDS	DP	DQ
Valuations ~ $U[0,1]$	5,000	100	0.99	**2,783.14** (76.55)	2,608.70 (67.75)	2,623.90 (66.63)	**2,810.01** (35.84)	2,644.23 (27.40)	2,639.27 (36.91)
	2,500	100	0.99	**1,654.30** (47.63)	1,594.08 (38.80)	1,436.87 (43.92)	**1,678.12** (17.59)	1,618.42 (13.59)	1,450.27 (19.27)
	5,000	50	0.99	**2,597.89** (76.53)	2,527.93 (68.75)	2,593.65 (76.66)	**2,623.14** (63.20)	2,566.28 (48.00)	2,614.88 (65.88)
	5,000	100	0.9	**735.74** (51.73)	338.14 (19.23)	419.37 (32.20)	**737.06** (51.09)	343.13 (14.38)	419.52 (29.89)
Valuations ~ *Exp*(1)	5,000	100	0.99	**5,691.37** (147.00)	5,556.96 (123.79)	5,623.47 (141.42)	**5,743.35** (113.13)	5,621.30 (97.55)	5,659.44 (117.48)
	2,500	100	0.99	**3,996.89** (133.49)	3,952.18 (102.07)	3,876.12 (124.08)	**4,066.79** (72.74)	4,012.61 (61.01)	3,920.14 (81.00)
	5,000	50	0.99	**4,299.23** (127.63)	4,233.73 (101.39)	4,297.08 (128.47)	**4,337.76** (175.60)	4,299.86 (156.16)	4,337.46 (176.03)
	5,000	100	0.9	**1,095.19** (70.99)	545.27 (34.29)	890.19 (59.91)	**1,095.66** (70.89)	551.32 (18.97)	890.86 (57.80)
Valuations ~ *Weibull*(4)	5,000	100	0.99	**4,091.99** (113.05)	3,986.06 (93.62)	3,623.63 (96.34)	**4,151.82** (42.05)	4,048.61 (34.55)	3,659.31 (44.00)
	2,500	100	0.99	**2,331.60** (75.91)	2,307.68 (66.61)	1,960.28 (61.76)	**2,374.69** (21.71)	2,345.76 (18.04)	1,984.26 (26.32)
	5,000	50	0.99	**4,032.48** (118.40)	3,942.44 (109.88)	3,975.18 (120.09)	**4,093.77** (50.09)	4,017.10 (39.98)	4,020.78 (57.07)
	5,000	100	0.9	**1,489.86** (96.94)	706.75 (34.97)	576.67 (45.28)	**1,498.47** (91.68)	716.31 (34.95)	576.70 (41.75)

Notes. Standard deviation of liquidation revenue is included in parentheses. The highest revenue in each set of simulations is shown in bold. See complete results in the online supplement (Appendix 5).

This also indicates that using deterministic representations of stochastic demand is an advantageous way of designing efficient liquidation strategies in general.

6.2. Performance Relative to Stochastic Dynamic Programming and Approximate Dynamic Programming

In this section, we benchmark our proposed heuristic approach with two advanced variations of dynamic programming approach, both of which solve for optimal liquidation strategy directly under stochastic demand. The first approach, namely stochastic dynamic programming (SDP), was used by Bitran and Mondschein (1997) to solve the liquidation problem with periodic pricing under stochastic demand. Their liquidation problem setup is very similar to ours. The second approach is based on approximate dynamic programming (ADP, discussed in Farias and Van Roy 2003), which is similar to the standard formulation we have discussed in Section 3.3 but uses linear functions to approximate the revenue function. Importantly, because the SDP approach directly solves for the optimal liquidation strategy under stochastic demand and does not rely on any approximation (unlike ADP), the revenue produced by SDP represents the *optimal* expected revenue under stochastic demand. As will be discussed in greater detail later, both SDP and ADP can naturally accommodate adjustable daily prices, making them good benchmarks to evaluate the performance bound of our proposed PDS (which also allows for adjustable daily prices). Below we first lay out the theoretical formulation of both SDP and ADP and then present the simulation results.

6.2.1. SDP Formulation. Let $REV(n,d)$ denote the maximum *expected* revenue from selling n items starting on day d of the liquidation period. Suppose price on day d is set to be p_d, then for an arriving customer, the probability of buying is $1 - F_d(p_d)$, where $F_d(.)$ is the valuation distribution CDF of customers arriving on day d. Take arrival process to be Poisson (consistent with Bitran and Mondschein 1997) with rate λ; then the arrival rate for buying customers is $\lambda_d = \lambda(1 - F_d(p_d))$. In other words, the probability that j buying customers arrive on day d is $\Pr(j) = e^{-\lambda_d} \cdot \lambda_d^j / j!$. The recurrence equation of this SDP problem is

$$REV(n,d) = \max_{p_d} \left\{ \sum_{j=0}^{\infty} \Pr(j) \cdot \left[\min(j,n) \cdot p_d + REV(n - \min(j,n), d+1) \right] \right\}.$$

After some algebraic manipulations (see Bitran and Mondschein 1997), the above equation can be written as

$$REV(n,d) = \max_{p_d} \left\{ n \cdot p_d + \sum_{j=0}^{n} e^{-\lambda_d} \cdot \lambda_d^j / j! \cdot \left[(j-n) \cdot p_d + REV(n-j, d+1) \right] \right\}.$$

The boundary conditions are $\forall d$, $REV(0,d) = 0$ and $\forall n$, $REV(n, D+1) = 0$. The problem is solved backward to find $REV(N,1)$. Each intermediate problem is a nonlinear univariate (p_d) optimization task, and we solve it using the Scipy implementation of the *L-BFGS-B* algorithm (Zhu et al. 1997), because it can find the optimal p_d more efficiently than alternatives (including brute force grid search) in the Scipy library.

Once the SDP is solved, it is straightforward to find the optimal price that should be set for each day. Specifically, suppose there are M_d remaining items starting on day d; then p_d should be set by *looking up* the price value that maximizes $REV(M_d, d)$, which has already been computed as part of the SDP calculations. Because solving SDP once is sufficient to access the optimal price on a given day for any starting inventory levels, it is natural to use SDP with adjustable daily prices, rather than setting all daily prices in advance. In other words, SDP is straightforwardly suitable to the dynamic price adjustments based on realized sales without incurring any additional computational overhead. Although it is possible to derive all daily prices at once from SDP results and set those prices in advance, doing so will only disadvantage SDP's revenue performance, because it forfeits SDP's natural capability to adjust prices in reaction to stochastic demand.

6.2.2. ADP Formulation. Next, we turn to the second algorithm based on the approximate dynamic programming approach. Our ADP formulation is similar to that of Topaloglu and Kunnumkal (2006), which approximates the value function with linear basis functions and converts the optimization problem into a linear program (see Farias and Van Roy 2003 for a discussion of this method). Let state space $\mathcal{R} = \{0, 1, \ldots, N\}$ represent all possible inventory levels, and let M_d be the state variable on day d. Let $REV_d(M_d)$ be the value function that represents the optimal revenue that can be obtained from selling M_d items starting on day d. To generate the constraints needed to solve the ADP using linear programming, we consider a finite set $\mathcal{P}_d$ of candidate prices on day d. Suppose $p_d \in \mathcal{P}_d$ is the price set on day d; then, denote $S_d = \mathbb{E}(Sales|p_d, M_d)$ as the expected sales on day d. The dynamic programming formulation of the problem has the following recurrence equation: $REV_d(M_d) = \max_{p_d}\{p_d \cdot S_d + REV_{d+1}(M_d - S_d)\}$. We approximate the value function by a linear function: $\forall d \in \{1, \ldots, D\}$, $REV_d(M_d) = \theta_d + \gamma_d M_d$, where $\{\theta_d, \gamma_d\}, d \in \{1, \ldots, D\}$ are parameters to be fitted ($2D$ parameters in total). Accordingly, solving this ADP problem is equivalent to solving the following linear programming problem (Topaloglu and Kunnumkal 2006):

$$\begin{aligned}&\min(\theta_1 + \gamma_1 N),\\ &\text{s.t. } \theta_d + \gamma_d M_d \geq p_d \cdot S_d + \theta_{d+1} + \gamma_{d+1}(M_d - S_d),\\ &\qquad \forall p_d \in \mathcal{P}_d, M_d \in \mathfrak{R}, \text{ and } d \neq D\\ &\theta_D + \gamma_D M_D \geq p_D \cdot S_D, \forall p_D \in \mathcal{P}_D, M_D \in \mathfrak{R}.\end{aligned}$$

The objective function to be optimized is the approximation of $REV_1(N)$ (i.e., the expected revenue of the entire liquidation problem.) Note that it is formulated as a minimization problem, rather than a maximization problem, simply because of the directions of linear constraints (the same formulation was used in Topaloglu and Kunnumkal 2006). On day d, the candidate price set, $\mathcal{P}_d$, consists of 100 expected order statistics of the underlying valuation distribution, because these candidate prices naturally correspond to different expected sale levels. Further simulations show that having more granular $\mathcal{P}_d$ increased computation time considerably without significant improvements in revenue. In total, there are $|\mathcal{P}_d| \cdot N \cdot D$ constraints. We solved the linear programming using the "CVXOPT" package in Python, a commonly used library for optimization tasks.

Once the linear programming problem is solved, the optimal liquidation strategy (i.e., daily prices) can be computed as $\forall d \neq D, M_d \in \mathfrak{R}$, $p_d = \text{argmax}_p\{p \times S_d + \theta_{d+1} + \gamma_{d+1}(M_d - S_d)\}$, and $p_D = \text{argmax}_p\{p \times S_D\}$ (Farias and Van Roy 2003). Similar to SDP, optimal prices for all possible days and inventory levels are obtained by solving ADP once. Therefore, ADP also supports adjustable daily prices naturally, without incurring any additional computational cost.

6.2.3. Simulation Experiments Comparing PDS with SDP and ADP. We set up the illustrative benchmarking simulations as follows. Daily arrival follows *Poisson*(100), and customer valuations on the first day follow either a uniform $U[0, 1]$, an exponential $Exp(1)$, or a Weibull distribution with several different shape parameters. A constant decay factor α determines valuation decay, such that customer valuations on day d follow either $U[0, \alpha^{d-1}]$, $Exp(1/\alpha^{d-1})$, or Weibull distribution with scale parameter α^{d-1}. Because both SDP and ADP are much more computationally expensive to solve, we simulate much smaller liquidation problems compared with the configurations in the previous section. Specifically, for each valuation distribution, we simulate a basic configuration with ($N = 100, D = 10, \alpha = 0.99$) and then vary inventory size, liquidation period length, and decay rate respectively in three separate simulations:(1) holding $D = 10$, $\alpha = 0.99$, and varying $N \in \{50,60,70,80,90,100\}$; (2) holding $N = 100$, $\alpha = 0.99$, and varying $D \in \{10,12,14,16,18,20\}$; and (3) holding $N = 100$, $D = 10$, and varying $\alpha \in \{0.99,0.98,0.96,0.94,0.92,0.9\}$. In addition, we also include a set of simulations whereby we vary the shape parameter of Weibull valuations, that is, $\rho \in \{2.0,2.5,3.0,3.5,4.0,4.5,5.0\}$ under the basic configuration ($N = 100, D = 10, \alpha = 0.99$). For each configuration, we generate 10,000 demand instances (i.e., 10,000 sets of customer arrivals and valuations) to calculate the average revenue obtained by PDS, SDP, and ADP, respectively. We report the average revenue and running time comparisons among PDS, SDP, and ADP. Because of space restrictions, we report a representative subset of results in Table 5 and include the complete set of simulation results in the online supplement (Appendix 6) in both table and figure formats.

According to Table 5, PDS is able to produce average total revenues that are very close to SDP, while maintaining a very significant computational efficiency advantage. According to the complete results in Online Appendix 6, across all problem configurations, PDS incurs only 0.03%–1.12% revenue loss but is 3,000–86,000 times faster, compared with SDP. Notably, when customer valuations follow *Weibull*(4) (similar to the setup of Bitran and Mondschein 1997), PDS is 25,500–86,000 times faster than SDP. We observe that the speed advantage of PDS over SDP increases as decay becomes faster (i.e., as α becomes smaller) as well as with larger inventory sizes.

Compared with ADP, PDS is associated with slightly higher or equal average revenue in the majority of cases or slightly lower revenue (0.01%–0.02%) in a few cases, while being 14,500–55,000 times faster (calculated on the basis of complete results in Online Appendix 6). Notably, PDS can result in higher revenue on average than ADP in many cases, because the performance of ADP hinges on the quality of revenue function approximations (Farias and Van Roy 2003). It is likely that our linear basis function does not approximate expected revenue well enough. However, having more complicated approximation functions would further increase the already lengthy running time of ADP, because there will be generally many more parameters to optimize for.

Because SDP optimizes expected liquidation revenue directly under stochastic revenue and without any approximation (unlike ADP), the above simulations demonstrate that PDS is capable of producing

Table 5. Total Revenue and Running Time Comparisons among PDS, SDP, and ADP

	N	D	α	Total revenue			Running time (seconds)		
				PDS	SDP	ADP	PDS	SDP	ADP
Valuations ~ $U[0,1]$	100	10	0.99	84.99	85.58 (+0.69%)	84.89 (−0.12%)	0.0021	9.75 (4,642×)	66.80 (31,809×)
	100	20	0.99	87.37	87.61 (+0.27%)	86.59 (−0.90%)	0.0046	19.37 (4,210×)	136.70 (29,717×)
	50	10	0.99	44.90	45.37 (+1.04%)	44.91 (+0.02%)	0.0010	3.04 (3,040×)	17.30 (17,300×)
	100	10	0.9	66.54	66.56 (+0.03%)	66.29 (−0.38%)	0.0013	10.69 (8,223×)	65.38 (50,292×)
Valuations ~ $Exp(1)$	100	10	0.99	216.08	216.39 (+0.14%)	214.64 (−0.67%)	0.0021	17.02 (8,104×)	60.35 (28,738×)
	100	20	0.99	269.06	270.01 (+0.35%)	266.89 (−0.81%)	0.0056	35.42 (6,325×)	126.42 (22,575×)
	50	10	0.99	139.10	139.95 (+0.61%)	138.45 (−0.47%)	0.0011	5.00 (4,545×)	15.95 (14,499×)
	100	10	0.9	152.76	153.07 (+0.20%)	151.88 (−0.58%)	0.0019	17.11 (9,005×)	60.21 (31,689×)
Valuations ~ $Weibull(4)$	100	10	0.99	116.32	117.32 (+0.85%)	116.30 (−0.02%)	0.0020	94.40 (47,200×)	60.30 (30,149×)
	100	20	0.99	120.24	120.38 (+0.12%)	119.56 (−0.57%)	0.0047	221.99 (47,231×)	127.32 (27,089×)
	50	10	0.99	61.77	62.47 (+1.12%)	61.49 (−0.46%)	0.0011	28.06 (25,509×)	16.02 (14,563×)
	100	10	0.9	94.98	95.10 (+0.13%)	94.44 (−0.57%)	0.0011	94.76 (86,145×)	60.00 (54,545×)
Valuations ~ $Weibull(2)$	100	10	0.99	142.79	142.83 (+0.03%)	142.09 (−0.49%)	0.0021	94.21 (44,861×)	60.16 (28,647×)
Valuations ~ $Weibull(3)$	100	10	0.99	124.45	125.37 (+0.73%)	124.18 (−0.22%)	0.0021	93.26 (44,409×)	60.15 (28,642×)
Valuations ~ $Weibull(5)$	100	10	0.99	111.77	112.74 (+0.86%)	111.78 (+0.01%)	0.0020	96.02 (48,010×)	60.50 (30,250×)

Notes. Numbers in parentheses represent the differences in revenue or running speed between PDS and SDP or ADP. See complete results in the online supplement (Appendix 6).

near-optimal revenue on average, while being much more scalable than SDP. To further demonstrate the scalability of PDS, we calculate the maximum liquidation problem sizes that PDS and SDP can accommodate within specific timeframes. We simulate liquidation problems in which valuations follow *Weibull*(4), arrival follows *Poisson*(100), and $\alpha = 0.99$. We then vary N and D and report the largest problem size that PDS and SDP can handle, respectively within one second, one minute, one hour, and one day. For simplicity, we set $N = 30D$ for each choice of D. The results are summarized in Table 6. Clearly, within any given timeframe, PDS is able to handle a much larger liquidation problem than SDP, that is, a 1,000 times (37.5 × 37.5) larger-sized problem for one-second timeframe, and the difference is even more significant for larger timeframes. In reality, if the size of the liquidation problem is relatively small and there is sufficient lead time to make pricing decisions, then one should clearly use SDP, because it produces optimal liquidation strategies under stochastic demand. However, when decisions need to be made within shorter timeframes or for relatively large-scale inventory problems, PDS offers very substantial computational scalability advantages while still achieving near-optimal revenue levels.

Table 6. Maximum Liquidation Problem Sizes Under SDP and PDS Within Different Time Frames

Time frame	SDP	PDS
One second	$N = 60$, $D = 2$	$N = 2{,}250$ (37.5×), $D = 75$ (37.5×)
One minute	$N = 150$, $D = 5$	$N = 10{,}500$ (70×), $D = 350$ (70×)
One hour	$N = 540$, $D = 18$	$N = 42{,}000$ (77.8×), $D = 1{,}400$ (77.8×)
One day	$N = 1{,}800$, $D = 60$	$N = 140{,}000$ (80×), $D = 4{,}800$ (80×)

Notes. Valuations follow *Weibull*(4), arrival follows *Poisson*(100), and $\alpha = 0.99$. Numbers in parentheses represent the difference between PDS and SDP regarding problem size for each dimension.

Finally, consider the largest liquidation problem for SDP in Table 6, where valuations follow *Weibull*(4), arrival follows *Poisson*(100), $N = 1800$, $D = 60$, and $\alpha = 0.99$. Whereas SDP took 1 day to solve the problem, PDS took 0.3 seconds to run and was associated with only 0.45% revenue loss. On a scale of problems that we have simulated in Section 6.1, SDP would take days to solve, whereas PDS still takes less than 1 second.

6.3. Additional Simulation Experiments

Besides the above benchmarking analyses, we conduct two additional sets of simulations. First, we simulate an inventory liquidation problem with *time-variant* arrival, as discussed in Section 4.4, to illustrate the applicability and performance of our proposed approach in such scenario. Specifically, we consider liquidating 100 items over a seven-day period, in which the first five days are weekdays and the last two days represent the weekend. Whereas customer arrivals on each weekday are drawn from *Poisson*(80), we assume that the arrival on a weekend is likely to be more intense and, hence, is drawn from *Poisson*(100). We use this set of simulations to compare the revenue performance of greedy heuristic with SDP under time-variant arrival. Consistently with our previous simulation experiments, we keep a constant valuation decay rate of 0.99 and simulate three types of customer

Table 7. Total Revenue and Running Time Comparisons Between PDS and SDP Under a Time-Variant Arrival Process

	Total revenue		Running time (seconds)	
Problem configuration	SDP	PDS	SDP	PDS
$U[0,1]$	79.60	79.09 (−0.64%)	9.66 (4,830×)	0.002
Exp(1)	169.90	169.58 (−0.19%)	13.35 (6,675×)	0.002
Weibull(4)	111.87	109.96 (−0.82%)	27.89 (13,950×)	0.002

Note. Values in parentheses represent the difference between PDS with SDP.

valuation distributions: uniform, exponential, and Weibull. The simulation results are summarized in Table 7. Again, compared with SDP, our heuristic-based PDS approach is able to produce very close-to-optimal total revenue, while already demonstrating a significantly faster computational performance even in such a small inventory liquidation problem setting.

Second, we simulate an inventory liquidation problem in which customer valuations are drawn from a distribution that does *not* satisfy the NDMR property (which is atypical in the liquidation literature that commonly uses NDMR-compliant uniform, exponential, or Weibull distributions to model customer valuations). For the sake of experimental completeness, we use this set of simulations to illustrate the revenue performance of greedy heuristic under not only demand stochasticity but also NDMR violation. The simulation results are included in the online supplement (Appendix 2).

7. Discussions and Conclusion

In this paper, we examine the inventory liquidation problem, in which a retailer liquidates a fixed number of identical items over a time period by strategically setting prices periodically (e.g., every day) on the basis of the high-level, distributional knowledge about stochastic consumer demand. We propose to solve the liquidation problem by deriving a deterministic representation of stochastic demand. Assuming that customer arrival and valuations follow known statistical distributions (e.g., which can be readily estimated from past transaction data), the expected arrivals and expected order statistics of valuation distributions represent informative and advantageous approximations of customer demand. Under the deterministic representation of demand, we develop two computational approaches for finding the optimal liquidation strategy that result in maximum total revenue. The dynamic programming approach is a general-purpose approach that is robust with respect to any customer valuation characteristics. The other approach, based on a greedy heuristic, is considerably less computationally complex and still provides optimal solutions under deterministic demand representation when customer valuation distributions satisfy the NDMR property, which is indeed satisfied by a number of statistical distributions typically used to represent customer valuations, such as uniform, exponential, and Weibull distributions. In summary, one of the key contributions of this paper is the identification and exploration of a general underlying property (i.e., NDMR) that is applicable to a broad set of valuation distributions representative of a variety of real-world inventory liquidation phenomena and under which our proposed highly scalable approach has strong revenue performance guarantees.

We conduct a comprehensive set of simulation experiments to evaluate the performance of our heuristic approach under stochastic demand. First, we compare our proposed heuristic approach with two other heuristic strategies, namely the fixed-price strategy and the fixed-quantity strategy, and their variations that allow daily adjustable prices. We demonstrate that, across different valuation distributions and both with and without daily adjustable prices, our heuristic approach is consistently better than the alternative strategies. The heuristic approach (including PAS and PDS) is especially advantageous when the valuation decay is relatively fast. Second, we compare PDS with a stochastic dynamic programming approach (Bitran and Mondschein 1997), which can obtain optimal expected revenue, as well as an approximate dynamic programming approach (Topaloglu and Kunnumkal 2006). Although SDP and ADP can solve the liquidation problem directly under stochastic demand and can naturally accommodate adjustable daily prices, they are much slower than PDS, and only SDP yields marginally higher revenue. Overall, deriving deterministic representation of stochastic demand and using our proposed heuristic approach can generate near-optimal liquidation strategies with superior computational efficiency.

Our work has significant managerial implications. In particular, we confirm that two simple, commonly used liquidation strategies, the fixed-price strategy and the fixed-quantity strategy, generally do not yield optimal revenue. As a result, managers instead need to consider adopting more sophisticated dynamic pricing mechanisms, such as the method proposed in the paper. Through simulation experiments, we also demonstrate the use of proposed computational techniques to understand changes in optimal strategies and revenues with respect to several important parameters of the liquidation problem. This would allow a retailer to make informed decisions with respect to various strategic tradeoffs. For example, having a shorter liquidation window can save time for the retailer at the cost of losing some revenue, but having a longer window does not bring additional revenue after some point when customer valuations become too low. At the same time, if appropriate marketing activities can effectively influence the magnitude of decay rates of customer valuations, such activities can be very profitable. Our comprehensive simulation experiments in Section 6 further demonstrate that the proposed heuristic approach is capable of obtaining near-optimal revenue under stochastic demand. Therefore, the heuristic approach can serve as a useful tool for managers to make liquidation-related decisions in realistic, stochastic demand scenarios. Finally, the computational efficiency of our approach enables managers to quickly experiment with different liquidation parameters and examine the resulting revenues even for large scale liquidation problems.

Our proposed computational strategy is valuable in a number of real-world scenarios. For example, our heuristic approach provides a significant speed-up over alternatives (e.g., stochastic dynamic programming and approximate dynamic programming) without significant sacrifices in revenue performance. Such improvement can be crucial when liquidation decisions need to be made for large inventories and/or with limited lead time. It is also valuable for third-party liquidation service providers, such as Liquidity Services (liquidation.com), because for them the capability to liquidate large volumes of items for their clients on a real-time basis is a key source of competitive advantage. Finally, going beyond the posted-price mechanisms that we have discussed in this paper, our approach can also be useful for designing B2C online auctions. Specifically, Bapna et al. (2002, 2003) examined the multiunit sequential auction, where the auction revenue depends critically on the value of *marginal bid* (i.e., the highest bid that does not win the auction) and the choice of bid increment. For this type of auction, our heuristic approach can help identify the marginal bid as well as choose the appropriate bid increment. Essentially, one can treat the price on a given day as an indicator of marginal valuation that determines the number of items to sell using any ascending multiunit auction mechanism, whether it is discriminatory or uniform price.

Our work also provides several interesting directions for future research. One possible direction is to consider demand uncertainty (as in Farias and Van Roy 2010) and develop useful heuristics that incorporate demand learning. Furthermore, one can also take into account certain strategic behaviors of customers. For example, with access to information such as the remaining inventory levels, customers may strategically wait until a later period to purchase the items at a lower price. Future research should consider these factors and form more comprehensive liquidation models and strategies.

Endnotes

[1] In particular, the arrival process and valuation distribution can be approximated with respective empirical distributions learned from historical sales data. The decay rate can be estimated according to the valuation distributions. We discuss the estimation method for several common valuation distributions in the online supplement (Appendix 3).

[2] In this paper, for convenience, we use "day" as an example of the minimum time unit for which the retailer can make price adjustments. Depending on specific context, this time unit could be an hour, a week, etc.

[3] Note that we do not assume arrival process to be time-invariant (i.e., we allow $L_{t1} \neq L_{t2}$ for $t1 \neq t2$), although for notational simplicity, we present our theoretical results and computational simulations on the basis of a time-invariant arrival process. All of our results can extend to time-variant arrival, by considering $B_d = \mathbb{E}(L_d)$, as will be discussed in Section 4.4.

[4] As discussed in Section 3.1, the setup of our model ensures that $V_{bd} \geq V_{b+1\,d}$ and $V_{bd} \geq V_{b\,d+1}$. Therefore, V_{11} is the highest valuation among all customers across all days.

[5] This is because there are DB possible DMRs to compute, and each takes constant time. Additionally, because typically $B < N$ in large-scale liquidation scenarios, the overall complexity of the heuristic approach is $O(DN) + O(DB) = O(DN)$.

[6] Although valuation decay rates are objective reflections of the generally declining customer valuation as time goes by, their magnitude may be indirectly affected if marketing activities are successful at changing customers' perceptions toward the value of the item being liquidated.

[7] Although we consider the most natural constructions of FP/FQ and DP/DQ here, other variations are possible. As an example, under fixed-price strategy, the seller can potentially set the fixed price to be the $(N + i)$th expected order statistic, where i is a certain predefined quantity. We discuss these variations and their performances in the online supplement (Appendix 7).

[8] We do not simulate customer valuations from the general forms of distributions, such as $U[a, b]$, $Exp(\lambda)$, or $Weibull(\lambda, \rho)$, because the range parameters (a and b) and the scale parameter (λ) only change the magnitude of expected order statistics but not how many items are sold on each day.

[9] If the valuation decay is fast and the liquidation period is relatively long, price-/quantity-based strategies can result in highly suboptimal revenues. In practice, additional tuning (e.g., shorten the liquidation period) may provide some improvement to these two simple strategies. In contrast, our proposed heuristic handles such situation automatically (i.e., does not require such additional tuning).

References

Araman VF, Caldentey R (2009) Dynamic pricing for nonperishable products with demand learning. *Oper. Res.* 57(5):1169–1188.

Aviv Y, Pazgal A (2002) Pricing of short life-cycle products through active learning. Accessed January 21, 2019, https://pdfs.semanticscholar.org/6173/90fd39f1b4286aa5314b207e042c490a6d4f.pdf.

Aviv Y, Pazgal A (2008) Optimal pricing of seasonal products in the presence of forward-looking consumers. *Manufacturing Service Oper. Management* 10(3):339–359.

Bapna R, Chang SA, Goes P, Gupta A (2009) Overlapping online auctions: Empirical characterization of bidder strategies and auction prices. *MIS Quart.* 33(4):763–783.

Bapna R, Goes P, Gupta A (2003) Analysis and design of business-to-consumer online auctions. *Management Sci.* 49(1):85–101.

Bapna R, Goes P, Gupta A (2005) Pricing and allocation for quality-differentiated online services. *Management Sci.* 51(7):1141–1150.

Bapna R, Goes P, Gupta A, Karuga G (2002) Optimal design of the online auction channel: Analytical, empirical, and computational insights. *Decision Sci.* 33(4):557–578.

Bitran G, Caldentey R (2003) An overview of pricing models for revenue management. *Manufacturing Service Oper. Management* 5(3):203–229.

Bitran G, Mondschein S (1997) Periodic pricing of seasonal products in retailing. *Management Sci.* 43(1):64–79.

Bitran G, Caldentey R, Mondschein S (1998) Coordinating clearance markdown sales of seasonal products in retail chains. *Oper. Res.* 46(5):609–624.

Craig NC, Raman A (2015) Improving store liquidation. *Manufacturing Service Oper. Management* 18(1):89–103.

Elmaghraby W, Keskinocak P (2003) Dynamic pricing in the presence of inventory considerations: Research overview, current practices, and future directions. *Management Sci.* 49(10): 1287–1309.

Farias DP, Van Roy B (2003) The linear programming approach to approximate dynamic programming. *Oper. Res.* 51(6):850–865.

Farias VF, Van Roy B (2010) Dynamic pricing with a prior on market response. *Oper. Res.* 58(1):16–29.

Gallego G, Van Ryzin G (1994) Optimal dynamic pricing of inventories with stochastic demand over finite horizons. *Management Sci.* 40(8):999–1020.

Hevner AR, March ST, Park J, Ram S (2004) Design science in information systems research. *MIS Quart.* 28(1):75–105.

McAfee RP, McMillan J (1987) Auctions and bidding. *J. Econom. Literature* 25(2):699–738.

McGill JI, Van Ryzin GJ (1999) Revenue management: Research overview and prospects. *Transportation Sci.* 33(2):233–256.

Pearson K (1902) Note on Francis Galton's problem. *Biometrika* 1(4): 390–399.

Smith SAS, Achabal DDD (1998) Clearance pricing and inventory policies for retail chains. *Management Sci.* 44(3):285–300.

Su X (2007) Intertemporal pricing with strategic customer behavior. *Management Sci.* 53(5):726–741.

Topaloglu H, Kunnumkal S (2006) Approximate dynamic programming methods for an inventory allocation problem under uncertainty. *Naval Res. Logist.* 53(8):822–841.

Wood C, Alford B, Jackson R, Gilley O (2005) Can retailers get higher prices for 'end-of-life' inventory through online auctions? *J. Retailing* 81(3):181–190.

Zhao W, Zheng YS (2000) Optimal dynamic pricing for perishable assets with nonhomogeneous demand. *Management Sci.* 46(3): 375–388.

Zhu C, Byrd RH, Lu P, Nocedal J (1997) Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Software* 23(4):550–560.