PRODUCTION AND OPERATIONS MANAGEMENT

PRODUCTION AND OPERATIONS MANAGEMENT

Vol. 22, No. 3, May–June 2013, pp. 555–570
ISSN 1059-1478 | EISSN 1937-5956 | 13 | 2203 | 0555
DOI 10.1111/j.1937-5956.2012.01380.x

Increasing the Revenue of Self-Storage Warehouses by Facility Design

Yeming (Yale) Gong
EM Lyon Business School, Lyon, France, gong@em-lyon.com

René B. M. de Koster
Rotterdam School of Management, Erasmus University, Rotterdam, The Netherlands, rkoster@rsm.nl

J. B. G. (Hans) Frenk
Faculty of Engineering and Natural Sciences, Sabanci University, Istanbul, Turkey, frenk@sabanciuniv.edu

Adriana F. Gabor
Econometric Institute, Erasmus University, Rotterdam, The Netherlands, gabor@ese.eur.nl

Self-storage is a booming industry. Both private customers and companies can rent temporary space from such facilities. The design of self-storage warehouses differs from other facility designs in its focus on revenue maximization. A major question is how to design self-storage facilities to fit market segments and accommodate volatile demand to maximize revenue. Customers that cannot be accommodated with a space size of their choice can be either rejected or upscaled to a larger space. Based on data of 54 warehouses in America, Europe, and Asia, we propose a new facility design approach with models for three different cases: an overflow customer rejection model and two models with customer upscale possibilities, one with reservation and another without reservation. We solve the models for several real warehouse cases, and our results show that the existing self-storage warehouses can be redesigned to generate larger revenues for all cases. Finally, we show that the upscaling policy without reservation generally outperforms the upscaling policy with reservation.

Key words: logistics and transportation; facility design and planning; self storage; warehouse management
History: Received: August 2011; Accepted: March 2012 by Eric Johnson, after 2 revisions.

1. Introduction

Self-storage warehousing is a considerable industry in the United States and a booming business in Europe and Asia. In the United States, for example, the Self Storage Association (SSA, the official association representing this industry in the United States) reported that while there were only 6,601 facilities at year-end 1984, the number of facilities rocketed to over 50,500 at year-end 2010 (Self Storage Association 2011). The SSA estimates that the gross revenues for 2010 of self storage facilities totaled approximately $22.0 billion, and "nearly 1 in 10 US households" rent a self-storage unit in the United States (Self Storage Association 2011). According to the 2008 industry annual report of the Self Storage Association of the United Kingdom, the number of self-storage warehouses increased by between 19% and 117% in different countries in Europe in the period 2007–2008 (Self Storage Association UK 2009). In the rest of the world, self-storage is also rapidly developing.

The reasons behind this rapid growth are related to the self-storage business model, which apparently caters to customers' needs. Public storage provides both private persons and small businesses a temporary storage opportunity at a centrally located facility. A typical self-storage warehouse contains storage spaces of different sizes and qualities, each with a specific number of storage units. A customer rents a storage unit of an appropriate size for one or multiple months.

With decades of management experiences in cost control, this industry has been able to control costs well. According to the SSA, at year-end 2010 the large self-storage facilities only employed "an average of 3.5 employees per facility" in the United States (see Self Storage Association 2011). Traditionally, labor cost is the largest cost component in warehouses (De Koster et al. 2007). With the self-storage operation mode, customers handle storage operations themselves, without interference of warehouse personnel (e.g., operational details can be found at http://www.

shurgard.eu). However, the existing storage sizes or the number of storage units available per size may not fit the needs of the market. The number of available units of some type may be insufficient, while other sizes are abundant. This results in either lost customers and revenue or inefficient utilization of capacity of one type, which may also bring potential loss in another type. Hence, the prime objective for these self-storage companies is to maximize expected revenue at a stable cost level. An obvious question is therefore whether it is possible to provide a facility design that it is a better fit between storage design (types and numbers) and market demand and that also maximizes the revenue.

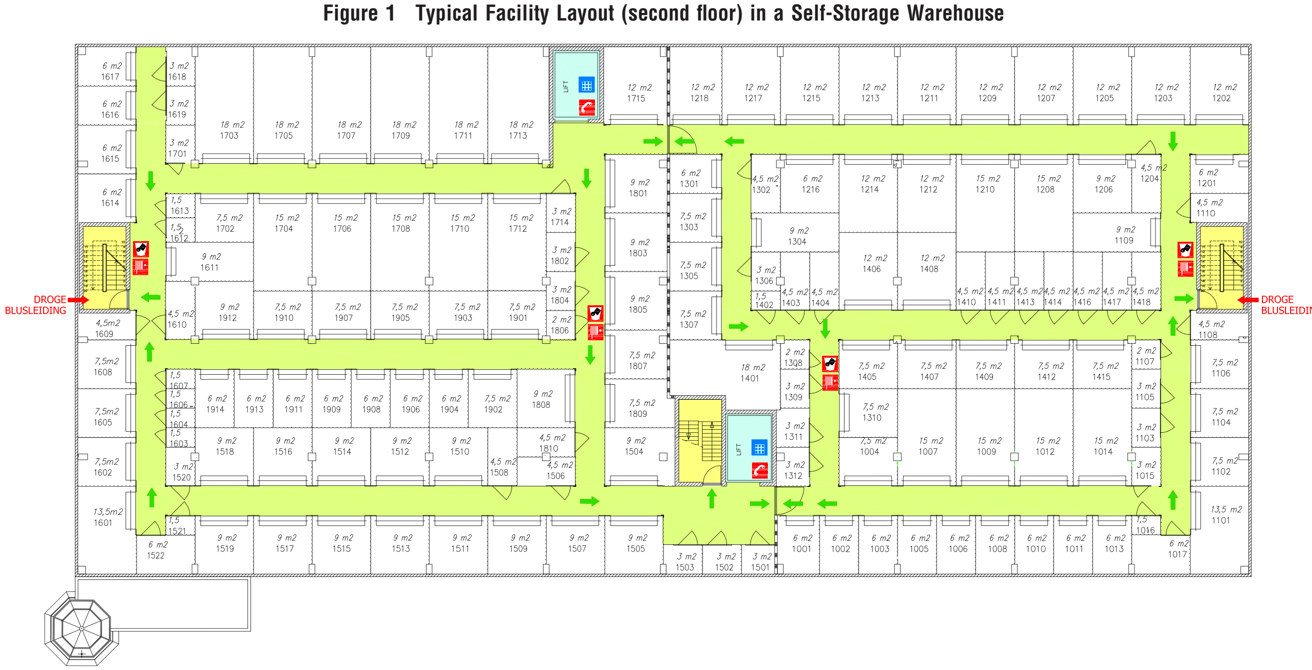
Our research has been inspired by interviews with Shurgard-Public Storage (the world's largest self-storage warehouse company) and other self-storage companies. Shurgard is an international corporation providing self-storage warehousing services to both private and business customers. Discussions with Shurgard's management team revealed the need for a warehouse design that maximizes revenues.

Figure 1 shows a typical example of a self-storage warehouse (only the second floor is shown), which contains storage rooms of 3, 4.5, 6, 7.5, 9, 12, 15, and 18 m^2 and in which many of the smaller spaces can be merged into larger ones. The height of these storage units is standard. In the United States, for example, the free height is 8 feet. Throughout this study, we therefore consider the two-dimensional facility design problem only. Our main focus is to determine the appropriate storage types and the number of storage units per type to fit market demand. We do not consider the remaining engineering problem, that is, designing a specific storage layout given storage types and the number of storage units per type. Many heuristics, optimization-based algorithms (see, e.g., chapters 5 and 6 in Francis et al. 1992; Tompkins et al. 2003), and commercial software tools (such as Factory Cad, FactoryFlow, Flow Planner) are available to solve such kinds of problems.

Warehouse facility design is a tactical decision: once a facility has been designed and built, it is difficult to adapt it to a changed environment. However, in self-storage, warehouse designs appear to be more flexible than in other warehouses as self-storage warehouses widely apply modular steel-base products like modular corridors, standardized internal wall panels, standardized swing doors, and roller doors. Particularly, the internal panel has a special patented "snap together" interlocking seam (see steelstorage.co.uk), rather than fixed jointing, which makes repartitioning of warehouse space easier. Most self-storage warehouses have a limited number of storage sizes (in the United States, usually 8 types) and most sizes are an integer multiple of a standard size. It is usually possible to remove or add nonsupporting walls to create, for example, one 9 m^2 room from a 3 m^2 and a 6 m^2 room. Admittedly, there are some constraints: it is impossible to merge split rooms while they are still occupied and by adding a wall there should still be an access door for both rooms. This space flexibility creates room for self-storage managers to adapt the layout of their facility to changing demand on a rather short term (like half yearly).

The design of a storage facility depends on the renting policy employed. To research the policies used in practice, we visited 54 self-storage warehouses (for

Figure 1 Typical Facility Layout (second floor) in a Self-Storage Warehouse



the data, see Table S1 in Appendix S1 in the Supporting Information), including 33 in the United States, 14 in Europe, and 7 in Asia, between December 2007 and July 2009. We collected price, design, and demand information and interviewed managers, customers, and employees. It appears that the operation modes are highly dependent on geographical location. Among the 54 warehouses, 22.2% are located in or close to city centers. These warehouses are inclined to reject customers when capacity is full as demand appears to be abundant. Some 64.8% of the warehouses try to upscale customers to larger storage sizes when the desired storage size is fully occupied, and 13.0% do not (or rarely) reject or upscale customers. Upscaling customers is possible in many cases if the next larger storage type is still available and the price difference is not too large. In practice, managers immediately show customers a larger storage type, when the smaller one is not available, as customers often have difficulties in expressing their exact space needs. Two situations can then be distinguished: *a priori* reserving some space for upscaled customers or no *a priori* reservations. The first policy is more convenient for warehouse employees: if a space type is fully occupied and the reserved space of the next larger type is not yet fully booked, they can offer it to the customer and ask whether he or she accepts this space at the, usually higher, price of the larger unit. If there is no *a priori* reservation, it is not always obvious for an employee whether upscale space should be offered, as this takes away space of the primary customers of the larger space type. However, a policy of no *a priori* reservations might generate more revenue. Note that upscaling by multiple types is unlikely, as it increases prices considerably. In addition, customers usually ask for the smallest unit type that fits their purpose. As customers store goods with a given volume, this gives a lower bound to the acceptable storage size. For this reason and because customers generally do not want to split their belongings over several smaller units, downscaling is rare in self-storage, unlike in other rental industries. This article therefore primarily focuses on facilities that reject customers or upscale them by one unit type when space is available. In all the policies we study, customers are only interested in one unit, and are not willing to split their belongings over more units. We build analytical models to provide designs that maximize the expected revenue for a given renting policy. We show that *a priori* reservation generates similar revenues to no *a priori* reservation in optimally sized facilities.

Allocating space to customers over a certain period of time has so far only been studied in a deterministic context (Zhang et al. 2008). However, in the self-storage environment, demand rates and renting periods are uncertain. The main contribution of this article is therefore to find a design that, under one of the commonly used renting policies, maximizes the expected revenue of a storage facility while taking rental demand and time uncertainty into account. This article deals only with short-term capacity management aspects, and we assume that prices are fixed. We do not address pricing aspects because storage prices are publicly announced in advance via print media, fare books, storage product brochures, or Internet and do not change on a short-term basis (for a related argument, see Talluri and Van Ryzin 2004, p. 176).

The remainder of this article is organized as follows. In the following section, we review the literature of related application areas and of related methods. Section 3 is devoted to a basic design model in an environment with high demand. In section 4, we incorporate the customer upscale problem by an overflow queue network. We do this for cases with and without *a priori* space reservation for upscaled customers. Section 5 shows results of the models for several warehouse cases. We conclude with final comments and directions for future research in section 6.

2. Literature Review

Facility planning and design helps organizations to achieve supply chain excellence in today's competitive global marketplace (Tompkins et al. 2003). Approximately 8% of the US gross national product (GNP) has been spent annually on new facilities since 1955 (Tompkins et al. 2003). Most facility design methods mainly consider cost control and focus on minimizing (internal) distance-based cost. Some of the research focuses on optimally sizing the warehouse and renting temporary storage space when demand is stochastic (e.g., Rao and Rao 1998). Zhang et al. (2008) model the problem of allocating customers to different warehouse spaces given deterministic demand, using a scheduling approach. Few facility design models aim at maximizing profits. However, without considering the market segment problem, which is viewed as an important common characteristic of revenue management research (see Weatherford and Bodily 1992), simple profit maximization can be reduced to cost minimization by, for example, the dual theory. To the best of our knowledge, no studies on facility design have focused on revenue management in the presence of stochastic demand with market segments for different space requirements.

There is some similarity between our problem and the literature on revenue management (see Talluri and Van Ryzin 2004). Several authors have written papers on revenue management, like Chiang et al. (2007), McGill and Van Ryzin (1999), and Weatherford and Bodily (1992). These works discuss applications of revenue management to several fields (but not warehousing). Solution techniques include mainly heuristics, dynamic programming, and mathematical programming. Airline management is one of main application fields of revenue management techniques (see Talluri and Van Ryzin 2004). Warehouse revenue management differs from airline revenue management in a number of ways. The major difference lies in the fact that in self-storage warehouses, customers may rent spaces of different sizes for multiple periods of time. In this respect, warehouse capacity management is closer to hotel operations management, where one is interested in renting strategies of different types of rooms to customers with the objective of profit maximization. However, the literature on hotel management only focuses on acceptance of customers and not on the capacity design of the hotel. The hotel revenue management literature typically uses (dynamic) Markov decision processes (Bitran and Gilbert 1996, Bitran and Mondschein 1995) and (static) mathematical programming models (Bitran et al. 1995) to derive optimal or near-optimal strategies for renting hotel rooms of different types to customers with random demands. The problem of optimally pricing units in a warehouse with a given configuration and without possibilities of upscaling is discussed in Chen and Sondhi (2009). Their goal is to establish prices that maximize the expected revenue for an already designed warehouse, unlike our goal to find a configuration that would maximize the profit when the price is given. Since upscaling seems to be a common practice in warehousing management, we are interested in models that can take this strategy into account. Note that upscaling introduces an extra dimension to the complexity of the facility design problem, as the optimization problem cannot be decomposed in separate optimization problems for each type of unit.

The problem studied in our paper may also be related to staffing problems in call centers (Koole and Mandelbaum 2002) or workplace training centers (Akcay et al. 2010). In a typical staffing problem in a call center, the question is to decide how many employees with a certain set of skills should be hired so that the quality of service is ensured and costs are minimized (Koole and Mandelbaum 2002). The problem that workplace training centers (or firms who offer service by using flexible resources) face is to design policies for accepting project requests and for assigning them to trainers with different skills to maximize the expected revenue (Akcay et al. 2010). As opposed to call centers, there may be a delay between a request for a project and the actual start of the project. A complicating aspect in the staffing problems mentioned (which is not inherent to the self-storage industry) is that a request (a call or a project) can be routed to agents with different skills. As such, considerable research has been dedicated to design optimal routing strategies for calls (Bhulai 2009) and assigning projects (Akcay et al. 2010). In the context of call centers, a question similar to the one we are trying to answer in this study would be the following: given the revenue obtained from a call type and the costs of hiring agents with different skills, how many agents from each skill category should be hired such that a maximum budget B is not exceeded and the revenue is maximized? In our search for related literature, we have not found papers studying this problem. Most of the papers on staffing analyze strategies for minimizing the costs of hiring agents while ensuring a certain quality of service. They do not consider a constraint on the total budget, since quality of service prevails over the costs of hiring an extra agent (Pot et al. 2008). For the self-storage industry, however, satisfying total capacity is a hard constraint which cannot be easily adjusted according to demand. In workplace training centers, assigning jobs to trainers is equally important as deciding whether to accept a request for a project. This results in acceptance policies that are dependent on the pool of available resources. In this article, however, we are interested in a good warehouse design under state independent customer acceptance policies that are common in practice.

3. Design Model without Upscaling

In this section, we present a design model without the possibility of upscaling, based on the settings of 12 warehouses as found in Chicago, Philadelphia, New York, Washington DC, Orlando, Hong Kong, Shanghai, and Brussels (for the data, see Table S1 in Appendix S1). To introduce this model, we assume that there are m different storage type units which can be requested by customers. A storage type i unit, $1 \leq i \leq m$, has an integer storage size area c_i and its rent is given by r_i per unit of time. In the remainder of this article, customers requesting a storage type i unit are called type i customers. Type i customers arrive according to a Poisson process with arrival rate λ_i and the different Poisson arrival processes are assumed to be independent. The following rental policy is assumed. An available type i storage unit is rented to any arriving type i customer. A type i customer who finds, upon arrival, that all storage type i units are occupied is rejected (no upscaling). We assume that the occupancy times of a storage type i unit are given by independent and identically distributed random variables with expected occupancy time β_i. The goal is to determine how many storage type i units should be built with the objective of revenue maximization.

To model this problem, we introduce the decision variable x_i denoting the number of constructed storage type i units. It is easy to see that the number of occupied storage type i units, $1 \leq i \leq m$, can be modeled as m independent $M/G/x_i/x_i$ queueing loss systems. Hence, the long-run average revenue is given by $\sum_{i=1}^{m} r_i L_i$ with L_i the long-run average number of customers in such a $M/G/x_i/x_i$ system. To calculate this, we use the following application of Little's law (see Tijms 1982, p. 345).

In a $G/G/x/x$ loss system with arrival rate λ and x servers, where customers pay at rate r for service, the long-run average revenue is equal to

$$rL = r\lambda(1 - P_{rej})\beta, \tag{1}$$

with L the long-run average number of customers in the system, β the expected service time, and P_{rej} the rejection probability.

By the PASTA (Poisson Arrivals See Time Averages) property, the rejection probability P_{rej} equals (see Cohen 1976, Gross and Harris 1998) the Erlang loss formula $B(x_i, \rho_i)$ with $\rho_i := \lambda_i \beta_i$ the load of the system and

$$B(x, \rho) := \frac{\rho^x}{x!}\left(\sum_{j=0}^{x} \frac{\rho^j}{j!}\right)^{-1}. \tag{2}$$

Notice that $B(0,\rho) = 1$ for every $\rho > 0$. Hence, by Equation (1), the long-run average revenue generated by accepted type-i customers equals $r_i\rho_i(1 - B(x_i, \rho_i))$.

Under the restriction that the total area of the warehouse is given by the integer C, the problem of maximizing the long-run average revenue can be formulated as

$$\max\left\{\sum_{i=1}^{m} r_i v(x_i, \rho_i) : \sum_{i=1}^{m} c_i x_i \leq C, x_i \in \mathbb{Z}_+, 1 \leq i \leq m\right\}, \tag{P1}$$

where

$$v(x, \rho) := \rho(1 - B(x, \rho)). \tag{3}$$

Problem (P1) can be solved by the following dynamic programming algorithm. Introduce for every $1 \leq k \leq m$ and $c \in \mathbb{Z}_+$ the feasible region $\mathcal{F}_k(c) := \{\mathbf{x} \in \mathbb{Z}_+^{m+1-k} : \sum_{i=k}^{m} c_i x_i \leq c\}$ and let

$$J_k(c) := \max\left\{\sum_{i=k}^{m} r_i v(x_i, \rho_i) : \mathbf{x} \in \mathcal{F}_k(c)\right\},$$

be the maximal long-run average revenue obtained from storage units of type $k,\ldots,m$, if the decision maker assigns to these units a total integer capacity c with $c \leq C$. Clearly, the optimal solution of problem (P1) is given by $J_1(C)$.

Since the function $x \mapsto v(x,\rho)$ is increasing, we first observe that $J_m(c)$, $c \in \{0,\ldots,C\}$, is given by

$$\begin{aligned} J_m(c) &= \max\{r_m v(x_m, \rho_m) : x_m \leq \lfloor c c_m^{-1} \rfloor\} \\ &= r_m v(\lfloor c c_m^{-1} \rfloor, \rho_m), \end{aligned}$$

with $\lfloor z \rfloor$ denoting the largest integer smaller than or equal to z. For type $1 \leq k \leq m - 1$, the value $J_k(c)$, $c \in \{0,\ldots,C\}$, can be iteratively calculated by the Bellman equation

$$J_k(c) = \max_{x_k \in \{0,\ldots,\lfloor c c_k^{-1} \rfloor\}} \{r_k v(x_k, \rho_k) + J_{k+1}(c - c_k x_k)\}.$$

Next, we investigate the monotonicity of the optimal objective function value of problem (P1) as a function of the load ρ_i, $i = 1,\ldots,m$ for each storage type.

LEMMA 1. *For every fixed* $x \in \mathbb{N}$, *the function* $v : \mathbb{N} \times [0, \infty) \to \mathbb{R}$ *given by* $v(x,\rho) = \rho(1 - B(x,\rho))$ *is increasing in the loads* ρ.

The proof of the above result can be found in Appendix S2 in the Supporting Information. Applying Lemma 1, it is immediately clear that the optimal objective function value of problem (P1) is increasing in ρ_i, $i = 1,\ldots,m$.

Generalization: A generalization of problem (P1) is to include a service restriction that arriving type i customers, $1 \leq i \leq m$, are rejected with a probability at most equal to σ_i (this parameter is set by the decision maker). In this case, we need to solve the problem

$$\max\left\{\sum_{i=1}^{m} r_i v(x_i, \rho_i) : \sum_{i=1}^{m} c_i x_i \leq C, x_i \in \mathbb{Z}_+, B(x_i, \rho_i) \leq \sigma_i, 1 \leq i \leq m\right\}.$$

Observe we always implicitly assume that the feasible region is nonempty. Since the function $x \mapsto B(x,\rho)$ is strictly decreasing, the above optimization problem reduces to

$$\max\left\{\sum_{i=1}^{m} r_i v(x_i, \rho_i) : \sum_{i=1}^{m} c_i x_i \leq C, x_i \geq B_{\rho_i}^{\leftarrow}(\sigma_i), x_i \in \mathbb{Z}_+, 1 \leq i \leq m\right\}, \tag{Q}$$

with

$$B_{\rho}^{\leftarrow}(u) := \min\{x \in \mathbb{Z}_+ : B(x, \rho) \leq u\}.$$

For solving optimization problem (Q), we can apply a dynamic programming algorithm similar to the one used for problem (P1). Rejected customers that ask for a larger size unit can be incorporated in models with upscaled customers, but with a modified arrival rate. We examine such models in the next section.

4. Design Model with Upscaling

Based on the data of 35 of the 54 warehouses investigated (for the data, see Table S1 in Appendix S1 in the Supporting Information), we now consider the following, in some cases more realistic model: customers who are initially interested in a storage type i unit may accept with probability p_i to pay a price r_{i+1} for a storage type $i + 1$ unit when all storage type i units are occupied. A customer willing to accept this is called an *upscaled* customer. Note that having to pay the price of the larger unit is typical for the storage business. This differs from many other businesses where upscaled customers pay the lower price (Gallego and Stefanescu 2009). If a storage type $i + 1$ unit is available, an upscaled customer will be served, otherwise the customer is rejected. As before, a customer who is initially interested in a type i storage unit is called a type i customer. We again model the arrival process of type i customers as a Poisson process with rate λ_i. However, we now assume that the holding times of a storage type i unit are independent and exponentially distributed with mean β_i. The main reason behind this assumption is to be able to analytically calculate the loss probability of upscaled customers. According to van Dijk and Kortbeek (2009), this is a reasonable assumption, as the loss probabilities appear to be rather insensitive to changes in the service time distribution. Given the price r_i for each storage type i unit per unit of time, the goal is to decide how many units of each type to build (and reserve) such that the long-run average revenue is maximized.

Based on practical experience, we consider two upscaling models. In the first one, units are reserved *a priori* for upscaled customers, while in the second, upscaled customers of type i may rent, upon arrival, any available unit at level $i + 1$. For the first model, we present an exact method for finding the long-run average revenue. The second model, however, is more complex and less tractable, as we will see in section 4.2. The analysis of the model without *a priori* reservation will be based on a first moment approximation assumption regarding the overflow process of upscaled customers. Using the outcomes of the mathematical model and simulation, the validity of this approximation is tested in Appendix S3. As shown in this appendix, this assumption is reasonable. Intuitively, the model with *a priori* reservation may be more suitable when the mean occupancy time of upscaled customers is higher than that of customers originally interested in a certain type of unit. However, as we will see in the numerical section, *a priori* reservation generates similar revenues to no *a priori* reservation when arrival rates are high. Thus, due to its tractability, the model with *a priori* reservation may be used as an approximation of the model without *a priori* reservation, in high-demand scenarios.

4.1. Model with *a priori* Reservation for Upscaled Customers

In this subsection, we consider an upscaling policy that reserves units for upscaled customers in the construction phase. In our study, 28.6% warehouses with upscale operations use this operational policy or a similar one with *a priori* reservations. The upscaling process can be described as follows. Let x_i, $1 \leq i \leq m$, be the number of storage type i units built for type i customers at level i and y_i, $1 \leq i \leq m - 1$, the type $i + 1$ units built and reserved for type i customers upscaled to level $i + 1$. The total number of type i units built is thus x_1 for type 1 units and $x_i + y_{i-1}$ for type i units, with $1 < i \leq m$. A type i customer who finds, upon arrival, that all the x_i storage type i units are occupied may choose to be upscaled and use one of the y_i reserved units, if one is available. If all y_i units are occupied, the customer is rejected. Customers of type m who find, upon arrival, that all x_m units are busy are directly rejected. A customer of type $i + 1$ is not allowed to occupy one of the y_i units reserved for upscaled type i customers, even if such a unit is available.

To analyze this model, we first look at the process followed by each type of customer separately. The long-run average revenue obtained from type i customers can be split in the long-run average revenue obtained from x_i and y_i units, respectively.

4.1.1. Long-Run Average Revenue from the x_i Units. The number of occupied storage type i units among the available x_i can again be modeled by a queueing loss system. Due to our assumption of exponentially distributed occupancy times, this is an $M/M/x_i/x_i$ loss queue. As before, the long-run average revenue obtained from the x_i units is given by $r_i v(x_i, \rho_i)$ with v listed in relation (3).

4.1.2. Long-Run Average Revenue from the y_i Units. To calculate the long-run average revenue obtained from the y_i units, we first need to characterize the arrival process of upscaled type i customers to level $i + 1$. Clearly, this is the same as the overflow process of rejected type i customers willing to upscale. Since the occupancy times of type i units are exponentially distributed, the arrival moments of type i customers finding all x_i storage type i units occupied are regeneration points of the $M/M/x_i/x_i$ loss queue (Wolff 1998). Hence, the overflow process of rejected type i customers is a (delayed) renewal process. By the PASTA property, the rate of this process is given by $\lambda_i B(x_i, \rho_i)$. Since each rejected type i customer is willing to upscale with probability p_i, the overflow process of rejected customers willing to upscale is therefore a delayed renewal process with rate $\eta_{i+1}(x_i)$ given by

$$\eta_{i+1}(x_i) = p_i \lambda_i B(x_i, \rho_i).$$

Again by Little's formula, the long-run average revenue generated by the y_i units is equal to

$$r_{i+1}\eta_{i+1}(x_i)\beta_i(1 - P_{rej}(x_i, y_i)),$$

with $P_{rej}(x_i, y_i)$ the rejection probability that an upscaled type i customer will find all the reserved y_i units occupied. The next result for the rejection probability $P_{rej}(x_i, y_i)$ is shown in Appendix S2 in the Supporting Information.

LEMMA 2. *If the sequence* $K_j, 0 \le j \le y_j$ *is given by* $K_0 := 1$ *and*

$$K_j = p_i^j \prod_{l=1}^{j} \frac{\gamma(x_i, l, \rho_i)}{1 - \gamma(x_i, l, \rho_i)}, \tag{4}$$

with

$$\gamma(x_i, l, \rho_i) = \frac{1 + \sum_{j=1}^{x_i} \binom{x_i}{j} \rho_i^{-j} \Pi_{k=0}^{j-1}(l+k)}{1 + \sum_{j=1}^{x_i+1} \binom{x_i+1}{j} \rho_i^{-j} \Pi_{k=0}^{j-1}(l+k)},$$

then

$$P_{rej}(x_i, y_i) = \frac{1}{\sum_{j=0}^{y_i} \binom{y_i}{j} K_j^{-1}}. \tag{5}$$

The optimization problem of deciding how many units of each type should be built or reserved so that the long-run average revenue obtained from all types of customers is maximized can now be written as

$$\begin{aligned} \max & \sum_{i=1}^{m} r_i v(x_i, \rho_i) + \sum_{i=1}^{m-1} r_{i+1}\eta_{i+1}(x_i)\beta_i(1 - P_{rej}(x_i, y_i)) \\ & \sum_{i=1}^{m-1} (c_i x_i + c_{i+1} y_i) + c_m x_m \le C \\ & x_i, y_i \in \mathbb{Z}_+. \end{aligned} \tag{P2}$$

Since the feasible solutions of (P2) with $y_i = 0$, $i = 1,\ldots,m-1$ correspond to the feasible solutions of the model without upscaling, it is clear that the optimal long-run average revenue attained by (P2) is at least the optimal long-run average revenue attained by (P1).

The above maximization problem can be solved via dynamic programming. Let again $J_k(c)$ be the maximal long-run average revenue obtained from storage units of type $k,\ldots,m$ if for those units a total integer capacity c with $c \le C$ is available. The optimal value $J_1(C)$ of (P2) can be computed recursively as follows. Since the function $x \mapsto v(x,\rho)$ is increasing, we first obtain for each capacity $c \in \{c_m, \ldots, C\}$ that $J_m(c)$ is given by

$$J_m(c) = \max_{x_m \in \{0,\ldots,\lfloor c c_m^{-1} \rfloor\}} r_m v(x_m, \rho_m) = r_m v(\lfloor c c_m^{-1} \rfloor, \rho_m).$$

Also for $c \in \{0, \ldots, c_m - 1\}$, it is obvious that $J_m(c) = 0$. Introducing for $1 \le k \le m-1$

$$S_k(c) := \{(x_k, y_k) \in \mathbb{Z}_+^2 : c_k x_k + c_{k+1} y_k \le c\},$$

we obtain that the Bellman recurrence relations for the functions J_k, $1 \le k \le m-1$ are given by

$$J_k(c) = \max_{(x_k, y_k) \in S_k(c)} \{f(x_k, y_k) + J_{k+1}(c - c_k x_k - c_{k+1} y_k)\}, \tag{6}$$

with

$$f(x_k, y_k) = r_k v(x_k, \rho_k) + r_{k+1}\eta_{k+1}(x_k)\beta_k(1 - P_{rej}(x_k, y_k)).$$

Using relations (5) and (4), it is easy to verify that the optimal objective function value of problem (P2) is an increasing function in each upscale acceptance probability p_i, $1 \le i \le m$. As for the model without upscaling, we are interested in the monotonicity of the optimal objective function value of problem (P2) with respect to the load ρ_i, $1 \le i \le m$ of each storage type. We conjecture that the optimal average revenue is increasing in ρ_i. We present numerical results in section 5.3 to support this conjecture.

4.2. Model without *a priori* Reservation for Upscaled Customers

In this subsection, we consider another operational policy in which capacity for upscaled customers is not reserved in advance. This policy is used by 71.4% warehouses with upscale operations (e.g., warehouses in Chicago, Philadelphia, Rotterdam, Brussels, and Lille). A type i customer finding, upon arrival, that all units of type i are occupied and choosing for upscaling may get any available unit of type $i+1$ (if such an available unit exists). This unit will be accepted with probability p_i. As before, we assume that type i customers arrive according to a Poisson process with rate λ_i and that the holding times for all storage units are independent and exponentially distributed with mean β_i for type i customers.

Under this assumption, the number of occupied storage type 1 units is a Markovian loss model. Moreover, we have seen in the previous subsection (see also Takacs 1982, Chapter 4) that the overflow process of rejected customers of an $M/M/c/c$ loss queue starting initially empty is a (delayed) renewal process with renewal epochs identified by the overflow epochs. The process at level 2 is, however, much more complex. The arrival process of type 2 and type 1 upscaled customers is the superposition of a delayed renewal process with a given arrival rate and a Poisson process with rate λ_2. Due to this, the loss system at level 2 with two different types of customers is impossible to analyze analytically. Since the overall arrival process becomes even more complicated at level $i \geq 3$, we introduce the following approximating Poisson assumption for the overflow process of rejected customers (in the sequel, H_2 denotes a two-phase hyper exponential distribution):

The overflow process of rejected customers in an $M/H_2/x/x$ loss queue with arrival rate λ and mean service time β is approximated by a Poisson process with rate $\lambda B(x,\rho)$, $\rho = \lambda\beta$ and $B(x,\rho)$ the Erlang loss probability that an arriving customer finds all x servers busy.

This approximation assumption is widely used in inventory models with lateral transshipments or emergency supplies (see Alfredsson and Verrijdt 1999, Axsäter 1990, Kranenburg and Van Houtum 2009). Koole and Talim (2000) show that this approximation also works well in the context of call centers. Note that in this study, the purpose of using the above assumption is to obtain a good approximation of the objective function. In Appendix S3 in the Supporting Information, we illustrate via simulations that this is indeed the case.

To analyze our model under the above approximation assumption, we also need the next result.

LEMMA 3. *Consider a queueing system with two types of customers and c servers and assume a customer who finds all the servers busy is rejected. The independent arrival processes of the two types of customers are Poisson with rates γ_1, γ_2 and the service times are exponentially distributed with mean ζ_1, ζ_2, respectively. For such a queueing system, the long-run probability that a rejected customer is of type i, i = 1,2 is given by $\frac{\gamma_i}{\gamma_1+\gamma_2}$.*

The above lemma also holds for independent and arbitrarily distributed service times with mean ζ_1, ζ_2, respectively. We give a proof of this more general result in Appendix S2.

As in the previous sections, we calculate the long-run average revenue for each type of storage unit by using Little's Law. For this, we first need to compute the average occupancy time of each type of storage unit and the probability that a customer will find all units of a certain type occupied.

To characterize the rental process of type i storage units, we proceed as follows. Let x_i be the number of type i storage units built, η_i the arrival rate of type $i - 1$ customers upscaled to units of type i, and $B(x_i, \rho_i, \eta_i)$ the probability that a customer interested in a type i unit (an upscaled type $i - 1$ customer or a type i customer) finds all type i storage units occupied. Clearly, $\eta_1 = 0$ and $B(x_1, \rho_1, \eta_1) = B(x_1, \rho_1)$, with $B(x_1, \rho_1)$ given by relation (2). As before, the number of occupied storage units of type 1 can be modeled as an $M/M/x_1/x_1$ queue. By the Poisson approximation assumption, the overflow process of rejected customers is a Poisson process with rate $\lambda_1 B(x_1, \rho_1)$. Hence, the rate of the arrival process of upscaled type 1 customers at level 2 is a Poisson process with rate $p_1\lambda_1 B(x_1, \rho_1) = \eta_2$.

Iteratively, for every $i \geq 2$, the arrival process is formed by upscaled customers of type $i - 1$ and type i customers. By the Poisson approximation assumption and induction, the arrival process is Poisson with rate $\lambda_i + \eta_i$. If there are storage units available, a type i customer will rent a type i storage unit for an exponential time with mean β_i, while an upscaled type $i - 1$ customer will rent such a unit for an exponential time with mean β_{i-1}. Since an arriving customer is of type $i - 1$ with probability $\frac{\eta_i}{\lambda_i+\eta_i}$ and of type i with probability $\frac{\lambda_i}{\lambda_i+\eta_i}$, the service time has a two-phase hyperexponential distribution with mean

$$\beta_i \frac{\lambda_i}{\lambda_i + \eta_i} + \beta_{i-1} \frac{\eta_i}{\lambda_i + \eta_i}.$$

We can thus conclude that the number of occupied storage units of type i can be modeled by an $M/H_2/x_i/x_i$ loss queue. Also the load of arriving type i and upscaled type $i - 1$ customers at level i is given by

$$\lambda_i\beta_i + \eta_i\beta_{i-1} = \rho_i + \eta_i\beta_{i-1}.$$

Hence, by the Poisson approximation assumption, the overflow process of rejected customers at this $M/H_2/x_i/x_i$ queue is a Poisson process with rate $(\lambda_i + \eta_i)B(x_i, \rho_i, \eta_i) = (\lambda_i + \eta_i)B(x_i, \rho_i + \eta_i\beta_{i-1})$. This implies, using also Lemma 3, that the arrival process of upscaled type i customers to storage units of type $i + 1$ is therefore a Poisson process with rate η_{i+1} given by

$$\begin{aligned}\eta_{i+1} &= p_i \frac{\lambda_i}{\lambda_i + \eta_i}(\lambda_i + \eta_i)B(x_i, \rho_i + \eta_i\beta_{i-1}) \\ &= p_i\lambda_i B(x_i, \rho_i + \eta_i\beta_{i-1}).\end{aligned} \tag{7}$$

Since by the PASTA property

$$\begin{aligned}&\mathbb{P}(\text{arriving customer not blocked at level } i) \\ &\quad = 1 - B(x_i, \rho_i, \eta_i),\end{aligned}$$

we also obtain again by Little's Law that the long-run average revenue generated by the x_i type i storage units is given by

$$r_i(\rho_i + \eta_i\beta_{i-1})(1 - B(x_i, \rho_i, \eta_i)) = r_i(\rho_i + \eta_i\beta_{i-1}) \times (1 - B(x_i, \rho_i + \eta_i\beta_{i-1})) = r_i v(x_i, \rho_i + \eta_i\beta_{i-1}), \tag{8}$$

with the function v defined in relation (3).

The problem of maximizing the long-run average revenue for the model without *a priori* reservation can now be formulated as follows:

$$\max\left\{\sum_{i=1}^{m} r_i v(x_i, \rho_i + \eta_i\beta_{i-1}) : \sum_{i=1}^{m} c_i x_i \le C, x_i \in \mathbb{Z}_+, 1 \le i \le m\right\}. \tag{P3}$$

First note that for each allocation $\mathbf{x}^\top = (x_1, \ldots, x_m) \in \mathbb{Z}_+^m$ satisfying the feasibility condition $\sum_{i=1}^m c_i x_i \le C$, the (approximated) long-run average revenue can be obtained by iteratively calculating $v(x_i, \rho_i + \eta_i\beta_{i-1})$ using relation (7).

Also observe that the above mathematical program is a nonlinear nonseparable integer programming problem. Solving such a mathematical programming model is extremely hard. Fortunately, by the interpretation of our problem, dynamic programming can be used to find the optimal allocation.

Let $M = \max_{1 \le i \le m}\{p_i\lambda_i\}$ (with $p_m = 0$). For every $1 \le k \le m$, introduce the optimal value function $J_k : \{0, 1, \ldots, C\} \times (0, M) \to \mathbb{R}, 1 \le k \le m$ given by

$$J_k(c, \eta) := \begin{cases} \text{the maximal long-run average revenue obtained from type } i \text{ storage units} \\ i \ge k, \text{if the available capacity for these units is } c \text{ and the arrival rate of} \\ \text{upscaled type } k-1 \text{ customers at level } k \text{ is } \eta. \end{cases}$$

Clearly, when a capacity of c is available for constructing type m storage units, it is most profitable to completely use this capacity and construct $\lfloor cc_m^{-1}\rfloor$ type m storage units, for any arrival rate $0 \le \eta \le M$ of upscaled type $m-1$ customers at level m. This implies by relation (8) that

$$J_m(c, \eta) = \begin{cases} r_m v(\lfloor cc_m^{-1}\rfloor, \rho_m + \eta\beta_{m-1}) & \text{if } c \in \{c_m, \ldots, C\} \\ 0 & \text{otherwise} \end{cases}. \tag{9}$$

By relations (7) and (8), we obtain for each $k < m$, that the maximal long-run average revenue $J_k(c, \eta)$ satisfies the (Bellman) equation

$$J_k(c, \eta) = \max_{x_k \in \{0,1,\ldots,\lfloor cc_k^{-1}\rfloor\}} \{J_{k+1}(c - c_k x_k, \lambda_k p_k B(x_k, \rho_k + \eta\beta_{k-1})) + r_k v(x_k, \rho_k + \eta\beta_{k-1})\}. \tag{10}$$

In other words, $J_k(c, \eta)$ is equal to the maximum long-run average revenue generated when $x_k \in \{0, 1, \ldots, \lfloor cc_k^{-1}\rfloor\}$ units are built for type k customers. In this case, according to (7), the revenue generated from type k units is equal to $r_k v(x_k, \rho_k + \eta\beta_{k-1})$. Additionally, revenue is generated from type i units, $i \ge k+1$. Since by (7), the arrival rate of upscaled type k customers is equal to $\lambda_k p_k B(x_k, \rho_k + \eta\beta_{k-1})$, the long-run average revenue obtained from units of type $i \ge k+1$ is equal to $J_{k+1}(c - c_k x_k, \lambda_k p_k B(x_k, \rho_k + \eta\beta_{k-1}))$. The maximum long-run average revenue is given by $J_1(C, 0)$.

Note that the main difficulty in solving the recurrent relation (10) is dealing with the continuity of the arrival rate η of updated customers. We therefore discretize the state space of η into multiples of a chosen parameter $h > 0$ and solve the recurrence relation for the discretized values. In Appendix S2 of the Online supplement, we show that we obtain a lower bound on $J_k(c, \eta)$ by solving (10) with all the rates η of upscaled customers rounded down to the closest multiple of h. Similarly, we obtain an upper bound on $J_k(c, \eta)$ by rounding up all the rates of upscaled customers to the closest multiple of h. In Theorem 1, we prove that the obtained lower and upper bounds are close to each other if the parameter h is small. A detailed description of the rounding procedure and all the proofs can be found in Appendix S2 in the Supporting Information. In presenting the results of our numerical experiments, we will approximate the long-run average revenue $J_1(C, 0)$ by the average between the lower and upper bounds.

5. Applications

In this section, we numerically investigate our design method for self-storage warehouses and explore its management insights. We first apply our method to warehouses with high demand and customer rejections, then we incorporate upscale operations, followed by a sensitivity analysis to test the robustness of the solution approach.

5.1. Application in Warehouses without Upscale Operations

Self-storage warehouses in downtown high-demand areas primarily follow a customer rejection policy when a particular storage type is not available. For

Table 1 Design for Warehouses without Upscale Operations

Warehouse	Items	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8	Revenue ($)
W. Chicago*	Size (ft^2)	5 × 5	5 × 10	7.5 × 10	10 × 10	10 × 15	10 × 20	10×25		
	Prices ($)	81	93	170	170	305	348	397		
	Demand	10 (2)‡	15 (2)	35 (2)	45 (2)	20 (2)	5 (3)	2 (3)		
	Old design	21	30	43	120	30	20	15		42,109
	New design	25	34	79	96	45	17	6		48,233
Van Buren†	Size (ft^2)	5×5	5×10	7.5 × 10	10 × 10	10 × 15	10 × 20	10×25	10 × 30	
	Prices ($)	66	78	102	139	281	348	398	480	
	Demand	19 (2)	61 (2)	48 (2)	80 (2)	11 (2)	10 (3)	8 (3)	9 (3)	
	Old design	41	63	52	126	43	41	0	0	45,649
	New design	40	103	23	101	19	24	16	19	52,693

*For W. Chicago, $C = 29,500\,\text{ft}^2$. †For Van Buren, $C = 35,325\,\text{ft}^2$. ‡A (B) means the average monthly demand (average single storage duration in months).

example, at a self-storage warehouse (W. Chicago) near the John Hancock center in Chicago, the average demand per month for small storage units is higher than its capacity. The Van Buren warehouse near the University of Illinois Chicago (UIC) has a similar experience: students store their personal belongings in this warehouse during the summer break. The receptionists directly reject customer requests when units of a storage type are fully occupied.

We used the summer 2008 prices, design, and demand data (the average monthly demand and the average single storage duration) of these two warehouses and applied the basic model with customer rejections. Results can be found in Table 1. Using the new design, the W. Chicago warehouse can increase its monthly average revenue by 14.5%. The number of 10′ × 25′ and 10′ × 10′ units should be reduced, whereas the number of 7.5′ × 10′ and 10′ × 15′ units should be increased.

The Van Buren warehouse can improve its monthly average revenue by 14.7% by adopting a new design. This warehouse has two major customer categories: UIC students and business customers from Chicago Loop (the Chicago central business district). UIC students typically rent size 5′ × 10′ storage units which are fit for families with "a studio or one-bed room" according to Public Storage's marketing brochure. Our experiments suggest that the number of these small rooms, in particular the 5′ × 10′ units, should be increased. At the same time, it is also beneficial to increase the number of 10′ × 25′ and 10′ × 30′ units to meet the demand of business customers who need large units.

5.2. Design Application for Warehouses with Upscale Operations

Some self-storage warehouses, often those with less abundant demand, try to upscale customers when they run out of space of a certain type. Spaanse Polder Rotterdam (S. P. Rotterdam) uses *a priori* space reservation, while N. Delaware Philadelphia warehouse (N. D. Philly) upscales customers without prior reservation. S. P. Rotterdam does not reject customers if the capacity in one storage type is full, but instead will try to upscale its customers if space in a higher storage type is available. As a general rule, if the number of units of a certain type is larger than twice the demand, they reserve half of the units for upscaled customers. The N. D. Philly warehouse is located near the B. Franklin bridge in Philadelphia, in the downtown area but without convenient transport access, so there is not such a big demand. Therefore, the management of this warehouse is not inclined to reject customers but, instead, will try to upscale them, if space of a storage type runs out. We applied the corresponding upscaling model in Section 4 to each warehouse, with data from summer 2008 for S. P. Rotterdam and from fall 2008 for N. D. Philly. Customer upscale acceptance probabilities were estimated by the warehouse manager. Note that the prices may be not proportional to the size of the units, due to, for example, the accessibility of the units. We present the results in Table 2.

The average revenue of both warehouses can be increased by 10.2% and 8.7%, respectively, by using our algorithms. Our results suggest that revenue growth at S. P. Rotterdam can be achieved by increasing the number of small units (type 1 units) and by reducing the number of type 5 units. It seems more beneficial to upscale customers of type 5 than to build type 5 units. This is probably due to the difference in price between type 5 and type 6 units and to the high probability of customers accepting the larger space. The number of type 6 units reserved for upscaled customers should be slightly decreased, while the number of type 6 units built for type 6 customers should be slightly increased. At the N. D. Philly warehouse, where upscaling is performed without *a priori* space reservations, an increase in revenue can be obtained by offering more type 4 units and by reducing the number of large units.

Table 2 Design for Warehouses with Upscale Operations

Warehouse	Items	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8	Revenue
S. P. Rotterdam*	Size (m^2)	3	6	9	12	15	18	22	27	
	Prices (€)	109	132	177	225	254	372	436	468	
	Demand	31 (2)†	31 (2)	33 (2)	13 (2)	7 (3)	8 (3)	2 (4)	2 (4)	
	Old design	34 (0)‡	44 (0)	58 (0)	25 (0)	18 (9)	27 (14)	3 (0)	4 (0)	37,480€
	New design	68 (0)‡	59 (0)	59 (0)	20 (0)	2 (0)	33 (12)	5 (0)	2 (0)	41,311€
N. D. Philly§	Size (ft^2)	5×5	5×10	10×10	10×15	10×20	10×25	10×30	10×40	
	Prices($)	65	79	132	227	222	255	326	396	
	Demand	34 (2)	90 (2)	70 (2)	50 (2)	40 (2)	30 (3)	9 (3)	2 (4)	
	Old design	78	180	144	22	54	22	24	4	67,017$
	New design	72	172	130	102	65	7	1	0	72,872$

*For S. P. Rotterdam, $C = 2118\text{m}^2$ and $\mathbf{p} = [0.8, 0.9, 0.9, 0.9, 0.8, 0.8, 0.9, 0]$. †A (B) means the average monthly demand (average single storage duration in months). ‡A (B) means the total number of storage units of a storage type (the reserved number of storage units for upscaled customers). §For N. D. Philly, $\mathbf{p} = [0.5, 0.5, 0.5, 0.7, 0.5, 0.5, 0.9, 0]$ and $C = 53750\,\text{ft}^2$.

5.3. Sensitivity Analysis

So far our results suggest that it is possible to create designs that improve revenue. Although facilities can be flexibly adapted to changing demand to some extent, we here investigate whether the proposed methods can consistently indicate how the design could be improved under varying demand. We use the S. P. Rotterdam warehouse as an example, since monthly demand data are available for a period of 2 years. We use sample-based sensitivity analysis (SBSA) for the W. Chicago warehouse, for which data are scarce.

The optimal design results for the S. P. Rotterdam warehouse are presented in Table 3. Although we use monthly demand data, for space reasons the table only summarizes average quarterly demand data. The table shows that a new design can result in an increase of about 11% in the monthly average revenue in all seasons. It suggests that more revenue can be obtained by increasing the number of small type 1 and 2 units, decreasing the number of moderate size type 3, 4, and 5 units, and increasing the number of large type 6 and 7 units. The analysis gives no recommendation for redesigning the largest type 8, since more units give increased revenue in 50% of the cases, and fewer units generate more revenue in 50% of the cases. However, redesigning storage type 8 is not a top priority, as it contributes rather little, both in terms of revenue and capacity, and its demand is small and volatile.

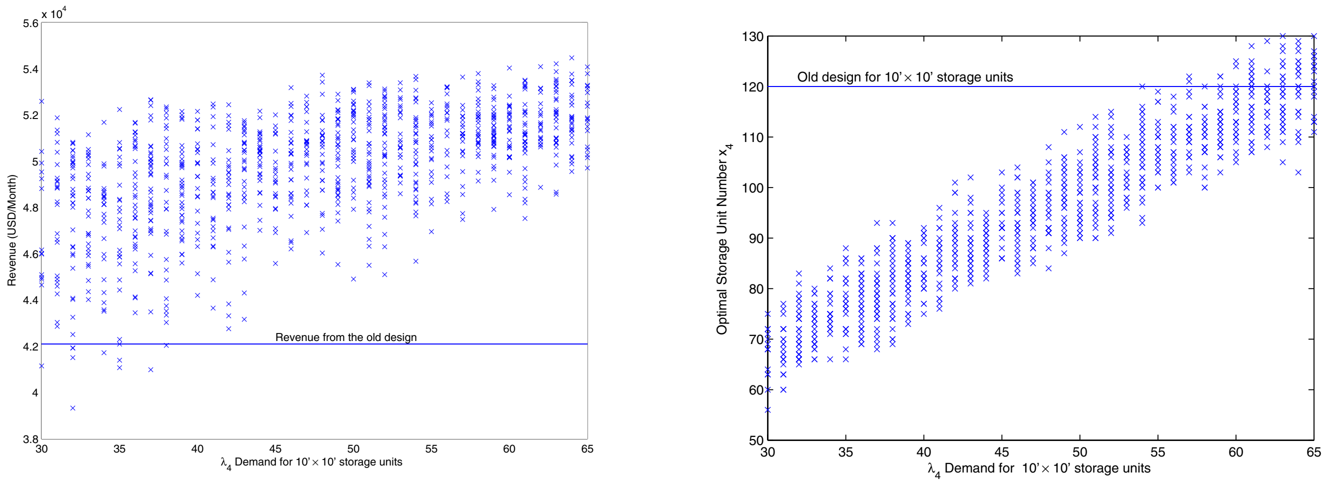
For the W. Chicago warehouse, for which detailed data were not available, we generated 1000 demand

Table 3 Sensitivity Analysis of the Design for S. P. Rotterdam Warehouse

Demand period	Items	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8	Revenue (€)	Impr.
	Size(m^2)	3	6	9	12	15	18	22	27		
	Prices (€)	109	132	177	225	254	372	436	468		
	Old design	34 (0)	44 (0)	58 (0)	25 (0)	18 (9)	27 (14)	3 (0)	4 (0)		
07 Spring	Demand	25 (2)*	24 (2)	25 (2)	11 (2)	6 (3)	7 (3)	3 (4)	2 (4)		
	New design	56 (0)†	49 (0)	48 (0)	20 (0)	2 (0)	32 (12)	11 (0)	5 (0)	38,938 (34,449)‡	13.03%§
07 Summer	Demand	21 (2)	19 (2)	18 (2)	11 (2)	8 (3)	6 (3)	2 (4)	2 (4)		
	New design	48 (0)	40 (0)	2 (0)	53 (31)	2 (0)	38 (19)	8 (0)	7 (0)	36,726 (32,612)	12.62%
07 Fall	Demand	25 (2)	26 (2)	30 (2)	10 (2)	7 (3)	6 (3)	3 (4)	1 (4)		
	New design	56 (0)	53 (0)	59 (0)	18 (0)	2 (0)	31 (14)	11 (0)	2 (0)	39,277 (35,157)	11.72%
07 Winter	Demand	22 (2)	23 (2)	23 (2)	11 (2)	8 (3)	7 (3)	3 (4)	2 (4)		
	New design	50 (0)	46 (0)	44 (0)	20 (0)	2 (0)	36 (16)	11 (0)	5 (0)	38,743 (34,940)	10.89%
08 Spring	Demand	33 (2)	33 (2)	34 (2)	14 (2)	8 (3)	9 (3)	3 (4)	2 (4)		
	New design	71 (0)	62 (0)	57 (0)	19 (0)	2 (0)	35 (13)	6 (0)	0 (0)	42,331 (38,354)	10.37%
08 Summer	Demand	31 (2)	31 (2)	33 (2)	13 (2)	7 (3)	8 (3)	2 (4)	2 (4)		
	New design	68 (0)	59 (0)	59 (0)	20 (0)	2 (0)	32 (12)	5 (0)	2 (0)	41,311 (37,303)	10.75%
08 Fall	Demand	28 (2)	31 (2)	36 (2)	13 (2)	7 (3)	7 (3)	1 (4)	1 (4)		
	New design	63 (0)	61 (0)	69 (0)	22 (0)	2 (0)	32 (13)	2 (0)	1 (0)	40,664 (36,875)	10.27%
08 Winter	Demand	21 (2)	23 (2)	28 (2)	9 (2)	6 (3)	6 (3)	2 (4)	2 (4)		
	New design	48 (0)	47 (0)	2 (0)	63 (47)	2 (0)	31 (13)	7 (0)	6 (0)	37,605 (33,787)	11.30%
	Suggestion	↑	↑	↓	↓	↓	↑	↑	—		

*A (B) means the average monthly demand (average single storage duration in months). †A (B) means the total number of storage units of a storage type (the reserved number of storage units of a storage type). ‡A (B) means the revenue based on the new design (the revenue based on the old design). §Let R_n denote the new revenue from the new design and R_o denote the old revenue. The improvement proportion (Impr.) is measured by $R_n/R_o - 1$.

Figure 2 A Sample-Based Sensitivity Analysis for the W. Chicago Case



rates for all seven storage types from uniform distributions with given demand rate intervals, based on the few demand realizations available. The duration of rent per storage type is kept at the level indicated in Table 1. We calculate the optimal storage design for every demand rate. The optimal revenue and layout based on 1000 demand rates are presented in Figure 2 by a scatter plot graph. We present only the results of storage type 4; scatter plots of the other 6 storage types have a similar pattern. The plot on the left shows that the probability of improving the layout by the proposed algorithm is 98.9%. The revenues are lower than the old revenue in 1.1% of the samples (see the bottom left corner). These results stem from cases with very low demand. The plot on the right shows that the probability that the old design has too many $10' \times 10'$ storage units is 95.1%, indicating that the new design is robust for demand fluctuations.

5.3.1. Sensitivity Analysis of the Optimal Solution with Respect to the Load of a Single Customer Type. So far, we have based our sensitivity analysis on real demand data. Since the demand rates of more types of customers change each season, it is difficult to draw conclusions on how the optimal revenue is affected by these changes. To gain a better insight into the effect of individual storage types on revenue and to check whether effects of different storage types cancel out each other, we created several artificial instances in which only the load of one storage type changes. As a test case, we take the S. P. Rotterdam warehouse, in which we calculate the optimal revenue when the load $\rho_i = \lambda_i \beta_i$ of each storage type i, $1 \leq i \leq m$, is 0.7, 0.8, 1.2, 1.3 of the load given in Table 2.

Our results show that the optimal long-run average revenue increases with the load for each storage type, thus confirming our conjecture in Section 4.1. The numerical results also suggest that a change in the load of a single storage type will not affect the optimal long-run average revenue considerably. In all the cases, the optimal long-run average revenue of the perturbed loads ranged between 90% and 103% of the optimal long-run average revenue with unperturbed loads. The percentage of units that change type when the loads are perturbed ranges between 1% and 10%. The number of these type 6 units ranges between 75% and 133% of the number of reserved units in the optimal solution of the unperturbed problem. Simultaneously, we can compare the optimal layout for the unperturbed load vector with the perturbed load (perturbed in one component). This yields an average revenue ranging between 90% and 98% of the (optimal) long-run average revenue of the perturbed load. Clearly, the optimal configuration for the unperturbed case might not be optimal for the perturbed case, and so this multiplication factor is always bounded above by one.

We use the S. P. Rotterdam case to test the influence of the upscale acceptance probabilities. We calculate the optimal revenue when the upscale acceptance probability of each customer type i is 0.7, 0.8, or 1.2 times the original upscale acceptance probability with a maximum of 1. The optimal revenue appears to be rather insensitive to changes in one upscale acceptance probability with a difference of less than 1%. The optimal configuration seems quite robust to changes in the upscale acceptance probabilities. Less than 3.2% of the units change their type. The type of the units reserved remains unchanged (type 6).

5.4. Comparison of the Upscaling Models

In the practical situations we encountered, managers were interested in the design of the warehouse under a given upscaling policy. The choice for a specific upscaling policy may, however, affect revenue. In this section, we compare the two upscaling algorithms with respect to revenue, with the goal of identifying scenarios where one policy performs better than the other. Since the model without upscaling is a special case of the model with *a priori* reservations (when no units are reserved for upscaled customers), we implicitly include this policy in our analysis.

In order to make a thorough analysis, we create a set of cases that cover many possible scenarios. In all our cases, we consider a warehouse of capacity $C = 8050\,\text{ft}^2$, with 5 types of possible units. The values of the chosen parameters are described in Table 4. In all the cases, the size c of the units and the revenue r have the same values.

We consider three possible upscale acceptance vectors of the form $0.1\mathbf{e}$, $0.5\mathbf{e}$, and $0.9\mathbf{e}$ corresponding to low, moderate and high upscale acceptance probabilities for customers of all types, with **e** denoting the (1, 1, 1, 1, 1, 0) vector. The fourth probability vector corresponds to a willingness to be upscaled that increases as the difference in price between the requested unit $r(j)$ and the offered unit $r(j + 1)$ decreases. In this case, the vector of upscale acceptance probabilities has components equal to $p_j^{pr} = 0.3 + 4/(r(j + 1) - r(j))$, for $j = 1, 2, 3, 4$.

We consider several arrival scenarios, in which the arrival rates of all types are chosen from the same uniform distribution. For a vector $\boldsymbol{\lambda} \in \mathbb{R}^5$, we will indicate by $\lambda \sim U[a,b]$ that the arrival rate of each type is chosen uniformly on $[a,b]$. The first scenario, given by $\boldsymbol{\lambda} \sim U[1,5]$, corresponds to the case when all types have low arrival rates, the second and third scenarios, given by $\boldsymbol{\lambda} \sim U[10,20]$ and $\boldsymbol{\lambda} \sim U[20,40]$, correspond to the case of a moderate arrival rate, the fourth scenario, given by $\boldsymbol{\lambda} \sim U[40,60]$, corresponds to a high arrival rate, and the last scenario, given by $\boldsymbol{\lambda} \sim U[1,60]$, corresponds to the case when some types have a high arrival rate and some types have a low arrival rate. We consider two types of occupancy durations. The first, $\beta_1 = [2, 2, 2, 3, 4]$, corresponds to the pattern encountered in practice. The second, $\beta_2 = [5, 4, 3, 2, 1]$, was created to verify the hypothesis that the reservation policy outperforms the policy without *a priori* reservation when an upscaled customer brings more profit than a regular customer of a certain type.

We generated 10 random instances for each combination of upscale acceptance probabilities and arrival rates. For each case, we calculate the optimal long-run average revenue for the upscaling policy with *a priori* reservation and the lower and upper bounds on the optimal revenue for the case without *a priori* reservation. In all the cases, the upscaled customers pay the price of the unit to which they are upscaled.

Table 4 Input Parameters for the Comparison of Upscaling Models

Parameter	Type 1	Type 2	Type 3	Type 4	Type 5
c (ft^2)	25	50	100	150	200
r (\$)	65	79	132	227	240

The results for $\beta_1 = [2, 2, 2, 3, 4]$ are presented in Table 5. The first and second columns of the table indicate the combination of arrival rate and upscale acceptance probabilities used. Columns 3, 4, and 5 represent the average, minimum, and maximum differences between the revenue gained without *a priori* reservation and the revenue with *a priori* reservation, expressed in percentages of the optimal revenue with *a priori* reservation. We take the average of the lower and upper bounds derived in section 4.2 as a reference for the optimal revenue in the case without *a priori* reservation. Columns 6 and 7 show the number of instances, out of 10, in which the policy without *a priori* reservation yielded a larger (lower) revenue than the policy with *a priori* reservation. The results are based on the observation that if the lower bound of the model without *a priori* reservation is higher than the optimal revenue with *a priori* reservation is applied, we can conclude that no prior reservation results in a higher long-run average revenue. Note that the average between the lower and upper bounds in the model without *a priori* reservation may be lower than the optimal revenue of the model with *a priori* reservation. If the optimal revenue with *a priori* reservation is lower than the upper bound of the revenue without *a priori* reservation, we cannot draw any conclusions due to insufficient information and we indicate this in the table by the symbol $\sim$. Finally, the last column of the table shows the average number of units reserved by the policy with *a priori* reservation.

The results in column 6 suggest that no prior reservation generally outperforms *a priori* reservation. Since the occupancy durations are nondecreasing, this result is to be expected, as upscaled customers are less profitable than regular customers of certain type of units. For low demand, the revenue of the *a priori* reservation policy was between the lower and upper bounds of the revenue without *a priori* policy in half of the cases. The third column suggests that the largest difference in revenue between the two policies is when the arrival rate is relatively low and moderate, namely, for $\boldsymbol{\lambda} \sim U[1,5]$ and $\boldsymbol{\lambda} \sim U[10,20]$. At the same time, more units are reserved when the upscale acceptance probability is high. Apparently, when arrival rates are low, the algorithm with *a priori* reservation tries to increase revenue by ensuring that upscaled customers can be served. The average number of reserved units decreases when arrival rates are high because there are enough regular customers to ensure a high revenue (see the rows corresponding to $\boldsymbol{\lambda} \sim U[20,40]$

Table 5 Optimal Revenues Obtained by the Upscaling Models for $\beta_1 = [2, 2, 2, 3, 4]$

λ	p	Av. rel. diff. (%)*	Min (%)†	Max (%)‡	$OPT_r \leq OPT_{nr}$§	$OPT_r > OPT_{nr}$§	Av. no. res. units¶
U[1,5]	0.1**e**	2.61	1.08	5.91	0	0	0
	0.5**e**	3.57	2.90	5.91	6	~**	0
	0.9**e**	2.48	−1.89	5.65	5	~	21.3
	e	1.02	−1.74	3.75	4	~	24.3
	$\mathbf{p}^{\mathbf{pr}}$	3.15	2.58	5.91	6	~	0
U[10,20]	0.1**e**	0.62	0.54	0.69	10	0	0
	0.5**e**	2.35	1.97	2.95	10	0	6.8
	0.9**e**	2.52	2.15	3.43	10	0	12.6
	e	1.93	1.54	3.08	10	0	15
	$\mathbf{p}^{\mathbf{pr}}$	2.13	1.69	2.61	10	0	3.6
U[20,40]	0.1**e**	0.11	0.04	0.22	10	0	0
	0.5**e**	0.40	0.19	0.70	10	0	0
	0.9**e**	0.56	0.31	0.98	10	0	1.5
	e	0.52	0.32	0.75	10	0	9
	$\mathbf{p}^{\mathbf{pr}}$	0.33	0.16	0.59	10	0	0
U[40,60]	0.1**e**	0.03	0.02	0.04	10	0	0
	0.5**e**	0.11	0.09	0.16	10	0	0
	0.9**e**	0.18	0.15	0.24	10	0	0
	e	0.19	0.16	0.21	10	0	0
	$\mathbf{p}^{\mathbf{pr}}$	0.11	0.10	0.14	10	0	0
U[1,60]	0.1**e**	0.35	0.02	1.08	10	0	0
	0.5**e**	1.04	0.11	2.38	10	0	7
	0.9**e**	0.96	0.18	2.62	10	0	12.1
	e	0.45	0.18	0.80	10	0	14.5
	$\mathbf{p}^{\mathbf{pr}}$	0.96	0.10	2.20	10	0	3.8

*Average, †Minimum, and ‡Maximum values of the relative differences between the revenue gained without *a priori* reservation and with *a priori* reservation expressed in percentages of the optimal revenue with *a priori* reservation. §OPT_r (OPT_{nr}) is the optimal objective function value for the model with (without) *a priori* reservation. ¶is the average number of reserved units. **means that OPT_r has values between the lower and upper bounds of OPT_{nr}.

and $\boldsymbol{\lambda} \sim U[40,60]$). When demand is higher, the differences in revenue obtained by the two policies are relatively small. Apparently, in this case, the arrival rate of upscaled customers is high enough and compensates for the loss in regular customers caused by reservation.

To see the influence of the occupancy duration on the results of the two policies, we perform the same experiments for $\beta_2 = [5, 4, 3, 2, 1]$. Table 6 presents the results for upscale acceptance probabilities $p = [1, 1, 1, 1, 0]$. The policy with *a priori* reservation performs slightly better when arrival rates are low (see the row for $\lambda \sim U[1,5]$) since upscaled customers are more profitable than regular customers, as they tend to stay in the system longer. However, the difference between the two policies disappears as arrival rates increase. Revenue from regular customers is preferred compared to the reservation of units which might remain unoccupied. The number of reserved units decreases as arrival rates increase, as is the case when occupancy rates go up.

The results in Tables 5 and 6 confirm the practice in the warehouse business. Our models justify why facilities with moderate demand tend to upscale customers, whereas facilities with high demand simply reject them. It also shows that *a priori* reservation with high acceptance probability (like in the S. P. Rotterdam warehouse case) can be close to optimal.

Table 6 Optimal Revenues Obtained by the Upscaling Models for $\beta_2 = [5, 4, 3, 2, 1]$, $p = [1,1,1,0]$

λ	Av.rel. diff. (%)*	Min (%)†	Max (%)‡	$OPT_r < OPT_{nr}$§	$OPT_r > OPT_{nr}$§	Av. no. res. units¶
U[1,5]	−0.048	−0.110	−0.002	0	6	58.2
U[10,20]	0.010	0.006	0.018	10	0	19.4
U[20,40]	0.001	0.001	0.002	10	0	4.6
U[40,60]	8×10^{-5}	2×10^{-5}	1×10^{-4}	10	0	0
U[1,60]	0.001	3×10^{-5}	0.004	10	0	7.3

*Average, †minimum, and ‡maximum values of the relative differences between the revenue gained without *a priori* reservation and with *a priori* reservation expressed in percentages of the optimal revenue with *a priori* reservation. §OPT_r (OPT_{nr}) is the optimal objective function value for the model with (without) *a priori* reservation. ¶is the average number of reserved units.

6. Concluding Remarks

This article outlines a design approach to improve revenue management of self-storage warehouses. Based on data of 54 of such facilities, we distinguished three groups, based on their demand level, which in turn is largely determined by geographical location. We investigated different policies used by warehouses with respect to upscaling customers. Warehouses with medium demand tend to upscale customers when storage space of their choice runs out, whereas those with large demand may simply reject customers.

We considered models with customer rejection (typical for high demand) and with customer upscaling (typical for medium demand). We distinguished two upscale models: with and without prior space reservation for upscaled customers. Models were solved using dynamic programming. For the case of upscaling without prior reservation, we used a first moment approximation of the overflow process of customers from one storage type to the next. Our experiments showed that the new facility design can improve the expected revenue of self-storage warehouses. Moreover, our sensitivity analysis suggests that the obtained design is rather insensitive to small changes in load and upscaling acceptance probabilities. Therefore, we recommend managers of self-storage warehouses to reconsider the configuration of the warehouse more frequently and to benefit from changing demand rates by adapting the space distribution. The experimental comparison of the two upscaling policies indicates that no prior reservation yields a larger revenue than prior reservation when upscaled customers pay the price of the larger unit to which they are upscaled and the occupancy times are nondecreasing in unit size, although the differences in revenue are slight. *A priori* reservation policy may slightly outperform the no reservation policy for low demand and occupancy times that increase with unit size. In both cases, the difference in revenue between the two policies decreases as the load increases. Hence, the choice for the *a priori* reservation policy may be justified by its simplicity and near-optimal revenue.

This article is one of the first to apply capacity management to facility design, taking into consideration market segmentation and uncertainty of data. Currently, this problem appears to be very relevant for self-storage warehouses. The design approach can be applied to other fields as well, particularly to hotel management. In hotel design, our method may be applied to decide which room types to build, based on market data. The research may further be applied to parking lot businesses, as parking lot layout has features similar to the layout of self-storage warehouses (e.g., different parking space sizes). In this case, the setting is slightly simpler than the one in this study, as the number of space classes is smaller. Our methods could also be applied to construction equipment leasing. This is a huge market, as most civil construction engineering companies do not purchase all equipment (such as bulldozers, shovels, and cranes) but rent it. Our algorithms can help equipment lease companies to determine the types and quantities of equipment they should possess. The approach can be applied to restaurant revenue management as well (see also Kimes and Thompson 2004, , 2005), in the context of determining the optimal table mix. There may also be applications in air cargo space design and car rental business (to determine the types and numbers of cars to be procured).

Acknowledgments

We thank the senior editor and the referees for giving comments which helped us improve the paper considerably. The authors gratefully acknowledge the support from Eric Schuurbiers, a marketing manager from Shurgard (Public Storage) corporation. We thank G. Koole (VU Amsterdam) and O. Boxma (TU Eindhoven) for helpful discussions. Yeming Gong acknowledges the financial support from NSFC (No. 70901028), the hosting of D. D. Einstein (U. Chicago), and the help of S. Xiong (Temple U.) to collect data in the United States and H. Yang (C. U. Hong Kong) to collect data in Asia.

References

Akcay, Y., A. Balakrishnan, S. H. Xu. 2010. Dynamic assignment of flexible service resources. *Prod. Oper. Manag.* **19**(3): 279–304.

Alfredsson, P., J. Verrijdt. 1999. Modeling emergency supply exibility in a two-echelon inventory system. *Manage. Sci.* **45**(10): 1416–1431.

Axsäter, S. 1990. Modelling emergency lateral transshipments in inventory systems. *Manage. Sci.* **36**(11): 1329–1338.

Bhulai, S., 2009. Dynamic routing policies for multi-skill call centers. *Probab. Engrg. Inform. Sci.* **23**(1): 75–99.

Bitran, G. R., S. M. Gilbert. 1996. Managing hotel reservations with uncertain arrivals. *Oper. Res.* **44**(1): 35–49.

Bitran, G. R., S. M. Gilbert, T. Y. Leong. 1995. Hotel sales and reservations planning. P. T. Harker, ed. *The Service Productivity and Quality Challenge*. Kluwer Academic Publishers, Dordrecht, 343–363.

Bitran, G. R., S. V. Mondschein. 1995. An application of yield management to the hotel industry considering multiple say stays. *Oper. Res.* **43**(3): 427–443.

Chen, L., S. Sondhi. 2009. Revenue management at extra space storage LLC. *Int. J. Revenue Manag.* **3**(2): 176–196.

Chiang, W.-C., J. C. H. Chen, X. Xu. 2007. An overview of research on revenue management: Current issues and the future research. *Int. J. Revenue Manag.* **1**(1): 97–128.

Cohen, J. W. 1976. *On Regenerative Processes in Queuing Theory*. Springer Verlag, Berlin.

De Koster, R., T. Le-Duc, K. J. Roodbergen. 2007. Design and control of warehouse order picking: A literature review. *Eur. J. Oper. Res.* **37**(5): 819–829.

van Dijk, N. M., N. Kortbeek. 2009. Erlang Loss Bounds for OT-ICU Systems. *Queueing Syst*. **63**(1-4): 253–280.

Francis, R. L., L. F. McGinnis, J. A. White. 1992. *Facility Layout and Location: An Analytical Approach*. Prentice Hall, Englewood Cliffs, NJ.

Gallego, G., C. Stefanescu. 2009. Upgrades, upsells and pricing in revenue management. Working paper, Columbia University, New York.

Gross, D., C. M. Harris. 1998. *Fundamentals of Queueing Theory*, 3rd edn. John Wiley & Sons, New York.

Kimes, S. E., G. M. Thompson. 2004. Restaurant revenue management at Chevys: Determining the best table mix. *Desicion Sci*. **35**(3): 371–392.

Kimes, S. E., G. M. Thompson. 2005. An evaluation of heuristic methods for determining the best table mix in full-service restaurants. *J. Oper. Manag*. **23**: 599–617.

Koole, G. M., A. Mandelbaum. 2002. Queueing models of call centers: An introduction. *Ann. Oper. Res.* **113**: 41–59.

Koole, G., J. Talim. 2000. Exponential approximation of multi-skill call centers architecture. *Proceedings of QNETs* **2000**(23): 1–10.

Kranenburg, A., G. Van Houtum. 2009. A new partial pooling structure for spare parts networks. *Eur. J. Oper. Res.* **199**(3): 908–921.

McGill, J. I., G. J. Van Ryzin. 1999. Revenue management: Research overview and prospects. *Transp. Sci.* **33**(2): 233–256.

Pot, A., S. Bhulai, G. Koole. 2008. A simple staffing method for multi-skill call centers. *Manuf. Serv. Oper. Manag.* **10**(3): 421–428.

Rao, A. K., M. R. Rao. 1998. Solution procedures for sizing of warehouses. *Eur. J. Oper. Res.* **108**: 15–26..

Self Storage Association. 2011. *Self Storage Association Fact Sheet*, June 2011. Self Storage Association, Alexandria, VA, USA.

Self Storage Association UK. 2009. *Annual Industry Report*, Self Storage Association of United Kingdom, Cheshire, UK.

Takacs, L. 1982. *Introduction to the Theory of Queues*. Greenwood Press Reprints, Westport, CT.

Talluri, K. T., G. J. Van Ryzin. 2004. *The Theory and Practice of Revenue Management*. Kluwer, Boston.

Tijms, H. C. 1982. *A First Course in Stochastic Models*. John Wiley & Sons, New York.

Tompkins, J. A., J. A. White, Y. A. Bozer, J. M. A. Tanchoco. 2003. *Facilities Planning*. John Wiley & Sons, New York.

Weatherford, L. R., S. E. Bodily. 1992. A taxonomy and research overview of perishable-asset revenue management: Yield Management, overbooking, and pricing. *Oper. Res.* **40**(5): 831–844.

Wolff, R. W. 1998. *Stochastic Modeling and The Theory of Queues*. Prentice Hall, Englewood Cliffs, NJ.

Zhang, X., Y. Gong, R. De Koster, S. Van de Velde. 2008. Storage scheduling decision models for revenue management of self storage warehouses. *TRAIL Selected Papers* **P2008/1**: 299–315.

Supporting Information

Additional Supporting Information may be found in the online version of this article:

Appendix S1: Data Collection for 54 Self-storage Warehouses.

Appendix S2: Proofs and Algorithms.

Appendix S3: Numerical Results for the Approximation Assumption.