Management Science

Robust Storage Assignment in Unit-Load Warehouses

Marcus Ang, Yun Fong Lim, Melvyn Sim,

Please scroll down for article—it is on subsequent pages

INFORMS is the largest professional society in the world for professionals in the fields of operations research, management
science, and analytics.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

Robust Storage Assignment in Unit-Load Warehouses

Marcus Ang, Yun Fong Lim
Lee Kong Chian School of Business, Singapore Management University, Singapore 178899, Republic of Singapore
{marcusang@smu.edu.sg, yflim@smu.edu.sg}

Melvyn Sim
NUS Business School, National University of Singapore, Singapore 119245, Republic of Singapore,
melvynsim@nus.edu.sg

Assigning products to and retrieving them from proper storage locations are crucial decisions in minimizing the operating cost of a unit-load warehouse. The problem becomes intractable when the warehouse faces variable supply and uncertain demand in a multiperiod setting. We assume a factor-based demand model in which demand for each product in each period is affinely dependent on some uncertain factors. The distributions of these factors are only partially characterized. We introduce a robust optimization model that minimizes the worst-case expected total travel in the warehouse with distributional ambiguity of demand. Under a linear decision rule, we obtain a storage and retrieval policy by solving a moderate-size linear optimization problem. Surprisingly, despite imprecise specification of demand distributions, our computational studies suggest that the linear policy achieves close to the expected value given perfect information and significantly outperforms existing heuristics in the literature.

1. Introduction

In a global economy, companies create a competitive advantage by paying substantial attention to their supply chain design and operations. A warehouse is a consolidation hub of various products in a supply chain. A large warehouse that supports a wide range of businesses may store thousands of different products, which pass through the warehouse in huge volumes daily. The operational efficiency of warehouses is crucial to the competence of a supply chain. For excellent reviews of warehouse design and operations, see Van den Berg (1999), de Koster et al. (2007), and Gu et al. (2007).

In a *unit-load* warehouse, all products are stored and retrieved in unit-load (pallet) quantities. Each pallet carries items of the same product and is generally handled singly at a time. Unit-load warehouses can be found upstream in a supply chain and are linked to production facilities. For example, §4 describes a case study that we have done with a unit-load warehouse owned by a logistics company. The warehouse stores pallets of products for a manufacturer located nearby and retrieves these pallets when they are requested. Unit-load warehouses can also be found in the reserve areas of large distribution centers, where

products are stored to replenish fast-pick areas in separate locations within the centers (Bartholdi and Hackman 2007). For the sake of presentation, each pallet is moved by a forklift in the following description. Our model is equally suitable for warehouses using very-narrow-aisle trucks or automated storage and retrieval systems (Roodbergen and Vis 2009).

Arrivals of products to a unit-load warehouse in each time period (say, every day) generally follow some predetermined schedule according to the suppliers' production plans. In contrast, the number of pallets of each product departing from the warehouse is less predictable because of uncertain demand. Arriving pallets are moved from a receiving dock to their storage locations. (In some cases, products need to be palletized before they are stored.) Each pallet is stored at its assigned location until it is requested and moved to a shipping dock. A key performance measure of a unit-load warehouse is the average travel time to move each pallet from a receiving dock to its storage location and then to a shipping dock. In the *storage assignment problem* of a unit-load warehouse the storage location of each pallet is determined so that the expected total travel time is minimized over a planning horizon.

To store a pallet, a forklift first moves the pallet from a receiving dock to the storage location. After inserting the pallet into its location, the forklift returns to the receiving dock. To retrieve a pallet, a forklift moves from a shipping dock to the pallet's location, extracts the pallet, and then moves it to the shipping dock. This travel pattern is known as *single-command* travel. We assume the time durations to insert and to extract each pallet at its storage location are constant and are ignored in deciding where to store the pallets.

A *storage assignment policy* is a set of rules that determines the storage locations of pallets. Two types of storage assignment policies are commonly used: *dedicated storage policies* and *shared storage policies*. A dedicated storage policy reserves each storage location for a specific product, and no other products can be stored in that location. Because products' locations are fixed, they can be memorized. A well-known dedicated storage policy is proposed by Heskett (1963, 1964). The author defines the cube-per-order index (COI) of a product as the ratio of its allocated storage space to its demand rate. Products are ranked in increasing COI and are assigned sequentially to locations with the smallest travel time. The inverse of COI of a product is called the turnover rate of the product. Thus, this policy is also known as the *full turnover policy* (de Koster et al. 2007, Roodbergen and Vis 2009).

Mallette and Francis (1972) consider multiple receiving and shipping docks and identify the optimal dedicated storage policy. They show that if all products have the same probability mass function for selecting a dock, the full turnover policy (or the COI policy) is optimal. Malmborg and Bhaskaran (1990) prove the optimality of the COI policy for more complicated situations.

Under a dedicated storage policy, the empty storage locations cannot be reassigned to other products when the inventory of a product is depleted. To overcome this problem, one can use a shared storage policy, which allows an empty location to be assigned to any product. Thus, a product may be assigned to different locations over time. A warehouse that implements a shared storage policy must rely on a computerized system to track products. An example of a shared storage policy is the *random policy*, which randomly assigns an arriving pallet to any empty location with equal probability.

Another example of a shared storage policy is the *class-based turnover policy* proposed by Hausman et al. (1976), Graves et al. (1977), and Schwarz et al. (1978). Under this policy, storage locations are grouped into several classes. Products with the highest turnover rate are assigned to the class with the smallest average travel time. A pallet is assigned randomly to any empty location within a class. Rosenblatt and Eynan (1989) and Eynan and Rosenblatt (1994) determine

the optimal boundaries of the classes in a rectangular warehouse for the class-based turnover policy. Thonemann and Brandeau (1998) derive the expected travel time for the random, COI, and class-based turnover policies under stochastic demand with a stationary distribution.

Goetschalckx and Ratliff (1990) study a warehouse that is perfectly balanced: For every time period, the number of arriving pallets equals the number of departing pallets, with identical duration of stay. They show that if a warehouse is perfectly balanced, an optimal shared storage policy is the *full duration-of-stay policy*, which assigns the pallets with the shortest duration of stay to the locations with the smallest travel time. Following the ideas of the class-based turnover policy, the authors also introduce the *class-based duration-of-stay policy*. This policy sorts the pallets in increasing duration of stay and sequentially assigns them to a predetermined number of classes. The pallets with the shortest duration of stay are assigned to the class with the smallest average travel time. If a warehouse is not perfectly balanced, the simulation results based on a deterministic environment suggest that the full duration-of-stay policy outperforms the class-based turnover policy with two classes, which in turn outperforms the class-based duration-of-stay policy with two classes.

Kulturel et al. (1999) compare the class-based turnover policy with the class-based duration-of-stay policy using computer simulations for a warehouse with three classes that face stochastic demand. Their results echo the findings of Goetschalckx and Ratliff (1990), which show that the class-based turnover policy generally outperforms the class-based duration-of-stay policy. All these papers assume demand for each product is stationary over the planning horizon, which is hardly true in reality because of seasonality or life cycles of products. A heuristic approach to handling nonstationary demand is to constantly reshuffle the products so that those with increased mean demand are relocated to more economic locations. See Table 4 of Gu et al. (2007) for references in this area of research.

In this paper, we study the storage assignment problem in a unit-load warehouse that faces uncertain demand over a multiperiod planning horizon. To capture the complexity of demand pattern, we consider a *factor-based demand model* in which demand for each product is affinely dependent on some uncertain factors. This allows us to model different seasonality effects. Furthermore, unlike most papers in the literature, we do not restrict the system to an economic order quantity (EOQ)-based replenishment policy with a fixed-order quantity for each product. Instead, we assume the number of arriving pallets of each product in each period is predetermined

according to the supplier's production plan. We adopt an approach based on *robust optimization* to solve the storage assignment problem.

Robust optimization is a promising approach to address optimization problems under uncertainty. This is justified by the significant growth in this area of research (see, for instance, Soyster 1973; Ben-Tal and Nemirovski 1998, 1999, 2000; Bertsimas and Sim 2003, 2004; Bertsimas et al. 2003; Chen et al. 2008; Chen and Sim 2009; Goh and Sim 2010; El-Ghaoui and Lebret 1997; El-Ghaoui et al. 1998; Erdoğan and Iyengar 2006). Robust optimization has also been implemented in a dynamic setting that involves decision making in stages. Bertsimas and Thiele (2006) propose a robust optimization solution to a multiperiod inventory control problem. Similarly, Adida and Perakis (2006) handle demand uncertainty in a dynamic pricing and inventory control problem by formulating a deterministic robust optimization problem.

To better adapt to a multistage decision process, Ben-Tal et al. (2004) introduce the concept of an *adjustable robust counterpart* that permits decisions to be delayed until information becomes available. Applications of the adjustable robust counterpart include Atamtürk and Zhang (2007) and Erera et al. (2009). Unfortunately, adjustable robust counterpart models are generally NP-hard. The authors propose a linear decision rule called *affinely adjustable robust counterpart*. Ben-Tal et al. (2005) demonstrate that an affinely adjustable robust counterpart can be remarkably effective in minimizing the worst-case objective of a multiperiod inventory control problem. Bertsimas et al. (2010) show that the affinely adjustable robust counterpart can be optimal in some situations. See and Sim (2010) demonstrate the effectiveness of piecewise linear decision rules in minimizing the expected objective of a multiperiod inventory control problem under stochastic demand with correlation.

Our approach to the storage assignment problem is similar to the affinely adjustable robust counterpart. We restrict the storage and retrieval decisions to a linear decision rule to obtain a tractable formulation. Specifically, our contributions can be summarized as follows:

1. We have developed a new method for storage assignment in unit-load warehouses. Our approach has the following three unique characteristics: (i) We can handle stochastic demand over multiple periods without specifying its exact probability distribution. (ii) We assume the number of pallets of each product arriving in each period can be of any integer. In contrast, almost all existing models adopt an EOQ-based replenishment policy with a fixed reorder quantity for each product. (iii) We consider the capacity constraint of each storage class to determine storage and retrieval decisions. In contrast, all existing approaches neglect these capacity constraints. A case

study with a company and numerical experiments based on realistic warehouse settings suggest that our method requires significantly less travel than the best-known methods in the literature.

2. We propose a factor-based demand model in which the demand for each product is affinely dependent on some uncertain factors. The support set of the uncertain factors is a polytope called the *factors support set*. The means of the uncertain factors are also uncertain. The support set of these uncertain means is also a polytope called the *factor means support set*. In our model we adopt the approach of Gilboa and Schmeidler (1989) of minimizing the worst-case *expected* total cost. In contrast, Ben-Tal et al. (2005) minimize the worst-case total cost, which is a special case of our model in which the factor means support set is the same as the factors support set.

3. We characterize the factors support set to ensure feasibility in the storage assignment problem under a linear decision rule. Such characterization is not required in the inventory control problem of Ben-Tal et al. (2005), where feasibility is guaranteed for any bounded support set, which is not the case for the storage assignment problem.

In this paper, let $\tilde{z}$ denote an uncertain variable and let $\tilde{\mathbf{z}}$ denote an uncertain vector. The support set W of an uncertain vector $\tilde{\mathbf{z}}$ is the smallest convex set containing all instances of $\tilde{\mathbf{z}}$. Let $f, g \colon W \to \Re^p$, for any integer p, denote function mappings. We use the notation $f(\tilde{\mathbf{z}}) \geq g(\tilde{\mathbf{z}})$ to represent statewise dominance: $f(\mathbf{z}) \geq g(\mathbf{z})$ for all $\mathbf{z} \in W$. Similarly, $f(\tilde{\mathbf{z}}) = g(\tilde{\mathbf{z}})$ denotes statewise equality: $f(\mathbf{z}) = g(\mathbf{z})$ for all $\mathbf{z} \in W$. We use $\mathbf{y}'$ to denote the transpose of vector $\mathbf{y}$.

2. Problem Formulation

We consider a unit-load warehouse with single command travel. We assume a single receiving dock and a single shipping dock whose locations may not coincide with each other. As explained later, our model can be generalized to warehouses with multiple receiving and shipping docks. For each storage location, we define its *store cost (retrieve cost)* as the travel time for a standard forklift to move from the receiving (shipping) dock to the location and then return to the receiving (shipping) dock.

We partition the storage locations using a grid into different *classes*. Figure E.1(a) in Appendix E shows an example. Each rectangle defined by four neighboring grid points corresponds to a class. Let s_j and r_j denote the average store cost and the average retrieve cost of all locations in class j, respectively, and we assume each location in class j has store cost s_j and retrieve cost r_j, for $j = 1, \ldots, N$. Each class j has capacity c_j, which represents the number of locations in the class. We assume the Nth class represents emergency storage, which has infinite capacity ($c_N = \infty$)

but incurs high storage and retrieval costs. When a pallet is assigned to a class, it is stored at an arbitrary location in the class.

Suppose there are M products indexed by $i = 1, \ldots, M$. We divide the planning horizon into T periods indexed by $t = 1, \ldots, T$. For each period, we assume all pallets from suppliers arrive at the start of the period and all pallets ordered by customers during the period are retrieved at the end of the period. Our goal is to minimize the total expected cost over the planning horizon. For convenience, let $\mathcal{N} = \{1, \ldots, N\}$, $\mathcal{N}^- = \{1, \ldots, N-1\}$, $\mathcal{M} = \{1, \ldots, M\}$, $\mathcal{T} = \{1, \ldots, T\}$, and $\mathcal{T}^+ = \{1, \ldots, T+1\}$.

2.1. Deterministic Demand

We begin with a deterministic model in which all information throughout the entire planning horizon is available at the start of the first period. Let a_i^t denote the number of pallets of product i arriving at the start of period t. Let v_{ij}^t be a decision variable determining the number of arriving pallets of product i that are assigned to class j in period t. Because all arriving pallets must be assigned to some classes, we have $\sum_{j \in \mathcal{N}} v_{ij}^t = a_i^t$, for $i \in \mathcal{M}$, $t \in \mathcal{T}$. Similarly, let d_i^t denote the number of pallets of product i that are ordered in period t. Let w_{ij}^t be a decision variable determining the number of pallets of product i that are retrieved from class j in period t. We have $\sum_{j \in \mathcal{N}} w_{ij}^t = d_i^t$, for $i \in \mathcal{M}$, $t \in \mathcal{T}$.

Let x_{ij}^t denote the number of pallets of product i in class j at the start of period t. We assume there is no initial inventory in the warehouse, so $x_{ij}^1 = 0$, for $i \in \mathcal{M}$, $j \in \mathcal{N}$. We do not allow backlog of orders at any time, even after the planning horizon. Thus, $x_{ij}^t \geq 0$, for $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}^+$. The inventory of product i in class j at the start of period $t+1$ is $x_{ij}^{t+1} = x_{ij}^t + v_{ij}^t - w_{ij}^t$, for $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}$. Because the inventory in each class j must not exceed its capacity, we have the capacity constraints $\sum_{i \in \mathcal{M}}(x_{ij}^t + v_{ij}^t) \leq c_j$, for $j \in \mathcal{N}^-$, $t \in \mathcal{T}$.

The decision variables should be restricted to integers. However, to yield a tractable formulation, we relax the integrality constraints and formulate a linear optimization problem to minimize the total cost of the warehouse, as follows:

$$Z_D = \min \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^t + r_j w_{ij}^t) \qquad (1)$$

$$\text{s.t. } \sum_{j \in \mathcal{N}} v_{ij}^t = a_i^t, \quad i \in \mathcal{M}, t \in \mathcal{T};$$

$$\sum_{j \in \mathcal{N}} w_{ij}^t = d_i^t, \quad i \in \mathcal{M}, t \in \mathcal{T};$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^t - w_{ij}^t, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T};$$

$$x_{ij}^1 = 0, \quad i \in \mathcal{M}, j \in \mathcal{N};$$

$$\sum_{i \in \mathcal{M}}(x_{ij}^t + v_{ij}^t) \leq c_j, \quad j \in \mathcal{N}^-, t \in \mathcal{T};$$

$$x_{ij}^t \geq 0, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}^+;$$

$$v_{ij}^t, w_{ij}^t \geq 0, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}.$$

We assume any shortage will be handled by the suppliers and will not incur any cost. Thus, there is always sufficient inventory to meet demand for every period. Equivalently,

$$\sum_{\tau=1}^{t} d_i^\tau \leq \sum_{\tau=1}^{t} a_i^\tau, \quad i \in \mathcal{M}, t \in \mathcal{T}. \qquad (2)$$

PROPOSITION 1. *Problem* (1) *is feasible if and only if inequalities* (2) *hold.*

PROOF. See Appendix A.

Note that the initial conditions $x_{ij}^1 = 0$, for $i \in \mathcal{M}$, $j \in \mathcal{N}$ are not overly restrictive. Proposition 1 can be extended to a more general initial setting in which inequalities (2) become $\sum_{\tau=1}^{t} d_i^\tau \leq \sum_{\tau=1}^{t} a_i^\tau + \sum_{j \in \mathcal{N}} x_{ij}^1$, for $i \in \mathcal{M}$, $t \in \mathcal{T}$, where $x_{ij}^1 \geq 0$ for $i \in \mathcal{M}$, $j \in \mathcal{N}$.

2.2. Factor-Based Demand Model

We adopt a factor-based demand model similar to that of See and Sim (2010) in which demand for each product in period t is affinely dependent on uncertain factors $\tilde{z}_k$, $k = 1, \ldots, K_t$, where K_t represents the number of such factors used to model demand up to period t. At the end of period t, the uncertain factors are realized and the values of $\tilde{z}_k$, $k = 1, \ldots, K_t$ are known. At the start of period $t+1$, new uncertain factors $\tilde{z}_k$, $k = K_t + 1, \ldots, K_{t+1}$ are introduced and they are realized at the end of period $t+1$. Thus, we have $1 \leq K_1 \leq K_2 \leq \cdots \leq K_T$. We define $\mathcal{K}_t \triangleq \{1, \ldots, K_t\}$, $\mathcal{K}_t^0 \triangleq \{0, \ldots, K_t\}$, $\tilde{\mathbf{z}}^t \triangleq (\tilde{z}_1, \ldots, \tilde{z}_{K_t})$, and $\tilde{\mathbf{z}} \triangleq \tilde{\mathbf{z}}^T$.

Demand for product i in period t is an affine function of $\tilde{\mathbf{z}}^t$: $d_i^t(\tilde{\mathbf{z}}^t) \triangleq d_i^{t,0} + \sum_{k \in \mathcal{K}_t} d_i^{t,k} \tilde{z}_k$, $i \in \mathcal{M}$, $t \in \mathcal{T}$, where $d_i^{t,k}$ are known coefficients for $k \in \mathcal{K}_t^0$. The factor-based demand model can capture correlation of demand for different products across different periods with appropriate values of $d_i^{t,k}$. For example, for a two-period, two-product case with $K_1 = 2$ and $K_2 = 4$, we have $d_1^1(\tilde{\mathbf{z}}^1) = d_1^{1,0} + d_1^{1,1}\tilde{z}_1 + d_1^{1,2}\tilde{z}_2$ and $d_2^1(\tilde{\mathbf{z}}^1) = d_2^{1,0} + d_2^{1,1}\tilde{z}_1 + d_2^{1,2}\tilde{z}_2$ for period 1; and $d_1^2(\tilde{\mathbf{z}}^2) = d_1^{2,0} + d_1^{2,1}\tilde{z}_1 + d_1^{2,2}\tilde{z}_2 + d_1^{2,3}\tilde{z}_3 + d_1^{2,4}\tilde{z}_4$ and $d_2^2(\tilde{\mathbf{z}}^2) = d_2^{2,0} + d_2^{2,1}\tilde{z}_1 + d_2^{2,2}\tilde{z}_2 + d_2^{2,3}\tilde{z}_3 + d_2^{2,4}\tilde{z}_4$ for period 2. If demand for product 1 is independent of demand for product 2, then we have $d_1^{1,2} = d_2^{1,1} = 0$ and $d_1^{2,2} = d_1^{2,4} = d_2^{2,1} = d_2^{2,3} = 0$. This implies $d_1^1(\tilde{\mathbf{z}}^1) = d_1^{1,0} + d_1^{1,1}\tilde{z}_1$ and $d_2^1(\tilde{\mathbf{z}}^1) = d_2^{1,0} + d_2^{1,2}\tilde{z}_2$ for period 1 and $d_1^2(\tilde{\mathbf{z}}^2) = d_1^{2,0} + d_1^{2,1}\tilde{z}_1 + d_1^{2,3}\tilde{z}_3$ and $d_2^2(\tilde{\mathbf{z}}^2) = d_2^{2,0} + d_2^{2,2}\tilde{z}_2 + d_2^{2,4}\tilde{z}_4$ for period 2. Furthermore, if demand for each product is independent across periods, then we have $d_1^{2,1} = d_2^{2,2} = 0$. This implies $d_1^1(\tilde{\mathbf{z}}^1) = d_1^{1,0} + d_1^{1,1}\tilde{z}_1$, $d_2^1(\tilde{\mathbf{z}}^1) = d_2^{1,0} + d_2^{1,2}\tilde{z}_2$, $d_1^2(\tilde{\mathbf{z}}^2) = d_1^{2,0} + d_1^{2,3}\tilde{z}_3$, and $d_2^2(\tilde{\mathbf{z}}^2) = d_2^{2,0} + d_2^{2,4}\tilde{z}_4$. In §4, we will demonstrate how

a warehouse manager could set up such a demand model using historical demand data.

In practice, it is often difficult to obtain the actual distribution of the uncertain factors. As a result, we need to handle this ambiguity and characterize the uncertain factors as follows.

ASSUMPTION U. *The uncertain factors $\tilde{\mathbf{z}}$ are random variables with an unknown distribution. They lie in a full dimensional polytope support set W called the factors support set. The factors have uncertain means with a support set $\hat{W}$ called the factor means support set, which is also a polytope. We define $\mathcal{U}$ as a family of distributions of $\tilde{\mathbf{z}}$ such that for all $\mathcal{P} \in \mathcal{U}$, we have $\mathrm{E}_{\mathcal{P}}(\tilde{\mathbf{z}}) \in \hat{W}$, where $\mathrm{E}_{\mathcal{P}}(\tilde{\mathbf{z}})$ represents the expected values of $\tilde{\mathbf{z}}$ under a distribution $\mathcal{P}$.*

Similar to inequalities (2), demand cannot exceed inventory for each product in any period. Thus, W and $\hat{W}$ are subsets of the set $G \triangleq \{\mathbf{z} \in \Re^{K_T}: \sum_{\tau=1}^{t} d_i^{\tau}(\mathbf{z}^{\tau}) \leq \sum_{\tau=1}^{t} a_i^{\tau}, d_i^t(\mathbf{z}^t) \geq 0, i \in \mathcal{M}, t \in \mathcal{T}\}$. Without loss of generality, we can define the factors support set as $W \triangleq \{\mathbf{z} \in \Re^{K_T}: \mathbf{z} \in G, \mathbf{z} \in S\}$, where S represents other constraints on the factors and can be expressed as $S = \{\mathbf{z} \in \Re^{K_T}: \exists \mathbf{u} \in \Re^{N_b}: \mathbf{Az} + \mathbf{Bu} \leq \mathbf{q}\}$, $\mathbf{A} \in \Re^{N_a \times K_T}$, $\mathbf{B} \in \Re^{N_a \times N_b}$, and $\mathbf{q} \in \Re^{N_a}$. Likewise, we can define the factor means support set as $\hat{W} \triangleq \{\mathbf{z} \in \Re^{K_T}: \mathbf{z} \in G, \mathbf{z} \in \hat{S}\}$, where $\hat{S} = \{\mathbf{z} \in \Re^{K_T}: \exists \mathbf{u} \in \Re^{\hat{N}_b}: \hat{\mathbf{A}}\mathbf{z} + \hat{\mathbf{B}}\mathbf{u} \leq \hat{\mathbf{q}}\}$, $\hat{\mathbf{A}} \in \Re^{\hat{N}_a \times K_T}$, $\hat{\mathbf{B}} \in \Re^{\hat{N}_a \times \hat{N}_b}$, and $\hat{\mathbf{q}} \in \Re^{\hat{N}_a}$. In classical robust optimization the uncertainty sets used are typically simple geometric sets such as boxes, ellipsoids, or their intersections. Such uncertainty sets are not always subsets of G and can render the problem infeasible. We assume W and $\hat{W}$ are nonempty. Note that $\hat{W} \subseteq W$; thus, we may assume $\hat{S} \subseteq S$. If the factor means are completely unknown, we have $\hat{W} = W$, which becomes the adjustable robust counterpart model of Ben-Tal et al. (2005).

3. A Robust Optimization Model

We consider a robust optimization model that takes adjustability into account as information unfolds. For each period t, the following sequence of events is repeated: At the start of period t, a decision on where to store the arriving pallets is made based on the information captured in $\tilde{\mathbf{z}}^{t-1}$. These pallets are then moved to their assigned storage locations. After the demand in period t is realized, $\tilde{\mathbf{z}}^t$ becomes available. A decision on where to retrieve pallets is made, and then pallets are retrieved from their storage locations. We define the following adjustable variables: (1) $v_{ij}^t(\tilde{\mathbf{z}}^{t-1})$ is the number of arriving pallets of product i assigned to class j at the start of period t after $\tilde{\mathbf{z}}^{t-1}$ is realized. This decision is made after the pallets for period t arrive at the warehouse. (2) $w_{ij}^t(\tilde{\mathbf{z}}^t)$ is the number of pallets of product i retrieved from class j at the end of period t after $\tilde{\mathbf{z}}^t$ is realized. This decision is

made after demand in period t is realized. (3) $x_{ij}^{t+1}(\tilde{\mathbf{z}}^t)$ is the number of pallets of product i in class j at the start of period $t+1$.

Because the actual demand distribution is not known, we consider a family of distributions of $\tilde{\mathbf{z}}$ under Assumption U. To address distributional ambiguity, we use the approach by Gilboa and Schmeidler (1989) that minimizes the worst-case expected total cost over the family of distributions as follows:

$$Z_R = \min \ \max_{\mathcal{P} \in \mathcal{U}} \mathrm{E}_{\mathcal{P}} \left[\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^t(\tilde{\mathbf{z}}^{t-1}) + r_j w_{ij}^t(\tilde{\mathbf{z}}^t)) \right] \quad (3)$$

$$\text{s.t.} \ \sum_{j \in \mathcal{N}} v_{ij}^t(\tilde{\mathbf{z}}^{t-1}) = a_i^t, \quad i \in \mathcal{M}, t \in \mathcal{T};$$

$$\sum_{j \in \mathcal{N}} w_{ij}^t(\tilde{\mathbf{z}}^t) = d_i^t(\tilde{\mathbf{z}}^t), \quad i \in \mathcal{M}, t \in \mathcal{T};$$

$$x_{ij}^{t+1}(\tilde{\mathbf{z}}^t) = x_{ij}^t(\tilde{\mathbf{z}}^{t-1}) + v_{ij}^t(\tilde{\mathbf{z}}^{t-1}) - w_{ij}^t(\tilde{\mathbf{z}}^t),$$
$$i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T};$$

$$x_{ij}^1(\tilde{\mathbf{z}}^0) = 0, \quad i \in \mathcal{M}, j \in \mathcal{N};$$

$$\sum_{i \in \mathcal{M}} (x_{ij}^t(\tilde{\mathbf{z}}^{t-1}) + v_{ij}^t(\tilde{\mathbf{z}}^{t-1})) \leq c_j, \quad j \in \mathcal{N}^-, t \in \mathcal{T};$$

$$x_{ij}^t(\tilde{\mathbf{z}}^{t-1}) \geq 0, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}^+;$$

$$v_{ij}^t(\tilde{\mathbf{z}}^{t-1}), w_{ij}^t(\tilde{\mathbf{z}}^t) \geq 0, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T};$$

$$v_{ij}^t, x_{ij}^t \in \mathcal{F}_{K_{t-1}}, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T};$$

$$w_{ij}^t \in \mathcal{F}_{K_t}, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T};$$

where $\mathcal{F}_p$ denotes a family of measurable functions that map $\Re^p$ to $\Re$ for any integer p. In problem (3), the optimal solutions v_{ij}^t and w_{ij}^t are functions representing the optimal storage and retrieval decisions.

3.1. Linear Storage-Retrieval Policy

It is generally intractable to determine the optimal storage and retrieval decisions for problem (3). To obtain a tractable formulation, we assume v_{ij}^t and w_{ij}^t are affine functions as follows:

$$v_{ij}^t(\tilde{\mathbf{z}}^{t-1}) = v_{ij}^{t,0} + \sum_{k \in \mathcal{K}_{t-1}} v_{ij}^{t,k} \tilde{z}_k, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}; \quad (4)$$

$$w_{ij}^t(\tilde{\mathbf{z}}^t) = w_{ij}^{t,0} + \sum_{k \in \mathcal{K}_t} w_{ij}^{t,k} \tilde{z}_k, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}. \quad (5)$$

Given the coefficients $v_{ij}^{t,k}, k \in \mathcal{K}_{t-1}^0$, and $w_{ij}^{t,k}, k \in \mathcal{K}_t^0$, for $i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}$, the functions v_{ij}^t and w_{ij}^t defined above constitute a *linear storage-retrieval policy* or a *linear decision rule*. Under Assumption U, the objective function of problem (3) becomes

$$\max_{\mathcal{P} \in \mathcal{U}} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^t(\mathrm{E}_{\mathcal{P}}(\tilde{\mathbf{z}}^{t-1})) + r_j w_{ij}^t(\mathrm{E}_{\mathcal{P}}(\tilde{\mathbf{z}}^t)))$$

$$= \max_{\mathbf{z} \in \hat{W}} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^t(\mathbf{z}^{t-1}) + r_j w_{ij}^t(\mathbf{z}^t)).$$

By limiting to linear decision rules, problem (3) becomes the following optimization problem:

$$Z_{LR} = \min \max_{z \in \hat{W}} \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^t(\mathbf{z}^{t-1}) + r_j w_{ij}^t(\mathbf{z}^t)) \quad (6)$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{N}} v_{ij}^t(\tilde{\mathbf{z}}^{t-1}) = a_i^t, \quad i \in \mathcal{M}, \, t \in \mathcal{T};$$

$$\sum_{j \in \mathcal{N}} w_{ij}^t(\tilde{\mathbf{z}}^t) = d_i^t(\tilde{\mathbf{z}}^t), \quad i \in \mathcal{M}, \, t \in \mathcal{T};$$

$$x_{ij}^{t+1}(\tilde{\mathbf{z}}^t) = x_{ij}^t(\tilde{\mathbf{z}}^{t-1}) + v_{ij}^t(\tilde{\mathbf{z}}^{t-1}) - w_{ij}^t(\tilde{\mathbf{z}}^t),$$
$$i \in \mathcal{M}, \, j \in \mathcal{N}, \, t \in \mathcal{T};$$

$$x_{ij}^1(\tilde{\mathbf{z}}^0) = 0, \quad i \in \mathcal{M}, \, j \in \mathcal{N};$$

$$\sum_{i \in \mathcal{M}} (x_{ij}^t(\tilde{\mathbf{z}}^{t-1}) + v_{ij}^t(\tilde{\mathbf{z}}^{t-1})) \le c_j, \quad j \in \mathcal{N}^-, \, t \in \mathcal{T};$$

$$x_{ij}^t(\tilde{\mathbf{z}}^{t-1}) \ge 0, \quad i \in \mathcal{M}, \, j \in \mathcal{N}, \, t \in \mathcal{T}^+;$$

$$v_{ij}^t(\tilde{\mathbf{z}}^{t-1}), w_{ij}^t(\tilde{\mathbf{z}}^t) \ge 0, \quad i \in \mathcal{M}, \, j \in \mathcal{N}, \, t \in \mathcal{T};$$

$$v_{ij}^t, x_{ij}^t \in \mathcal{L}_{K_{t-1}}, \quad i \in \mathcal{M}, \, j \in \mathcal{N}, \, t \in \mathcal{T};$$

$$w_{ij}^t \in \mathcal{L}_{K_t}, \quad i \in \mathcal{M}, \, j \in \mathcal{N}, \, t \in \mathcal{T};$$

where $\mathcal{L}_p$ denotes a family of affine functions that map $\mathfrak{R}^p$ to $\mathfrak{R}$ for any integer p. Clearly, if problems (3) and (6) are feasible, we have $Z_R \le Z_{LR}$. Chen et al. (2008) show that a feasible stochastic optimization problem can become infeasible under a linear decision rule. Even if problem (3) is feasible, it is not clear whether there exists a linear decision rule that is feasible. Fortunately, this is not the case for the storage assignment problem.

THEOREM 1. *Under Assumption U, problem (6) is feasible and its objective function Z_{LR} is finite. The coefficients of the optimal linear storage-retrieval policy can be computed by solving the following optimization problem:*

$$Z_{LR} = \min \quad g^0 + \max_{z \in \hat{W}} \sum_{k \in \mathcal{K}_T} g^k z_k \quad (7)$$

$$\text{s.t.} \quad g^k = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^{t,k} + r_j w_{ij}^{t,k}), \quad k \in \mathcal{K}_T^0;$$

$$\sum_{j \in \mathcal{N}} v_{ij}^{t,k} = \begin{cases} a_i^t & \text{if } k=0, \\ 0 & \text{otherwise}, \end{cases}$$
$$i \in \mathcal{M}, \, k \in \mathcal{K}_T^0, \, t \in \mathcal{T};$$

$$\sum_{j \in \mathcal{N}} w_{ij}^{t,k} = \begin{cases} d_i^{t,k} & \text{if } k \in \mathcal{K}_t^0, \\ 0 & \text{otherwise}, \end{cases}$$
$$i \in \mathcal{M}, \, k \in \mathcal{K}_T^0, \, t \in \mathcal{T};$$

$$x_{ij}^{t+1,k} = x_{ij}^{t,k} + v_{ij}^{t,k} - w_{ij}^{t,k},$$
$$i \in \mathcal{M}, \, j \in \mathcal{N}, \, k \in \mathcal{K}_T^0, \, t \in \mathcal{T};$$

$$x_{ij}^1 = 0, \quad i \in \mathcal{M}, \, j \in \mathcal{N};$$

$$h_j^{t,k} = \sum_{i \in \mathcal{M}} (x_{ij}^{t,k} + v_{ij}^{t,k}), \quad j \in \mathcal{N}, \, k \in \mathcal{K}_T^0, \, t \in \mathcal{T};$$

$$h_j^{t,0} + \sum_{k \in \mathcal{K}_T} h_j^{t,k} z_k \le c_j,$$
$$\forall \mathbf{z} \in W, \, j \in \mathcal{N}^-, \, t \in \mathcal{T};$$

$$v_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} v_{ij}^{t,k} z_k \ge 0,$$
$$\forall \mathbf{z} \in W, \, i \in \mathcal{M}, \, j \in \mathcal{N}, \, t \in \mathcal{T};$$

$$w_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} w_{ij}^{t,k} z_k \ge 0,$$
$$\forall \mathbf{z} \in W, \, i \in \mathcal{M}, \, j \in \mathcal{N}, \, t \in \mathcal{T};$$

$$x_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} x_{ij}^{t,k} z_k \ge 0,$$
$$\forall \mathbf{z} \in W, \, i \in \mathcal{M}, \, j \in \mathcal{N}, \, t \in \mathcal{T}^+;$$

$$v_{ij}^{t,k} = x_{ij}^{t,k} = h_j^{t,k} = 0,$$
$$i \in \mathcal{M}, \, j \in \mathcal{N}, \, k \in \mathcal{K}_T \backslash \mathcal{K}_{t-1}, \, t \in \mathcal{T};$$

$$w_{ij}^{t,k} = 0, \quad i \in \mathcal{M}, \, j \in \mathcal{N}, \, k \in \mathcal{K}_T \backslash \mathcal{K}_t, \, t \in \mathcal{T}.$$

PROOF See Appendix B.

For a special case where the factor means are known, we have $\hat{W} = \{E(\tilde{\mathbf{z}})\}$. We can translate the factors such that $E(\tilde{\mathbf{z}}) = \mathbf{0}$. The objective function of problem (7) reduces to $Z_{LR} = \min g^0$, which becomes the optimal expected total cost under a linear decision rule.

Note that the objective function and some of the constraints of problem (7) involve the parameters $\mathbf{z}$ over the support sets $\hat{W}$ and W. These are known as robust counterparts (see, for instance, Ben-Tal and Nemirovski 1998). Problem (7) can be represented as a linear optimization problem. For brevity, we present the derivation of the constraints corresponding to the robust counterparts involving the set W in the resultant linear optimization problem. Derivation for the robust counterpart involving the set $\hat{W}$ is similar.

PROPOSITION 2. *The variables $\mathbf{y}$ and r are feasible in the robust counterpart $\mathbf{z}'\mathbf{y} \le r$, $\forall \mathbf{z} \in W$, or equivalently $\max_{z \in W} \mathbf{z}'\mathbf{y} \le r$, if and only if there exist $\boldsymbol{\gamma} \in \mathfrak{R}^{N_a}$ and $\boldsymbol{\alpha}^t, \boldsymbol{\beta}^t \in \mathfrak{R}^M$, for $t \in \mathcal{T}$, that are feasible in*

$$\sum_{t \in \mathcal{T}} (\bar{\mathbf{a}}^{t'} \boldsymbol{\alpha}^t + \mathbf{d}^{t,0'} \boldsymbol{\beta}^t) + \boldsymbol{\gamma}' \mathbf{q} \le r,$$

$$\sum_{t \in \mathcal{T}} (\bar{\mathbf{D}}^{t'} \boldsymbol{\alpha}^t - \mathbf{D}^{t'} \boldsymbol{\beta}^t) + \mathbf{A}' \boldsymbol{\gamma} = \mathbf{y},$$

$$\mathbf{B}' \boldsymbol{\gamma} = \mathbf{0},$$

$$\boldsymbol{\gamma} \ge \mathbf{0}, \quad \boldsymbol{\alpha}^t, \boldsymbol{\beta}^t \ge \mathbf{0}, \, t \in \mathcal{T},$$

where $\bar{\mathbf{a}}^t = \sum_{\tau=1}^{t}(\mathbf{a}^\tau - \mathbf{d}^{\tau,0})$, $\mathbf{a}^{t\prime} = (a_1^t \cdots a_M^t)$, $\mathbf{d}^{t,0\prime} = (d_1^{t,0} \cdots d_M^{t,0})$, $\bar{\mathbf{D}}^t = \sum_{\tau=1}^{t}\mathbf{D}^\tau$, *and*

$$\mathbf{D}^t = \begin{pmatrix} d_1^{t,1} & \cdots & d_1^{t,K_t} & 0 & \cdots & 0 \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ d_M^{t,1} & \cdots & d_M^{t,K_t} & 0 & \cdots & 0 \end{pmatrix} \in \Re^{M \times K_T}.$$

PROOF. See Appendix C.

We use Proposition 2 to transform problem (7) to a linear optimization problem. Because the resultant formulation is heavy in notation, we present it in Appendix D.

3.2. Restricted Linear Storage-Retrieval Policy

The large number of decision variables needed to characterize the linear storage-retrieval policy in problem (7) impedes practical implementations. According to Equation (5), the number of variables needed to determine the retrieval decisions $w_{ij}^t(\tilde{\mathbf{z}}^t)$ under the linear decision rule is $|\mathcal{K}_t| + 1$, where $|\mathcal{K}_t| = Mt$. The full implementation of the linear decision rule is memory intensive and limits the problem size we could handle. To address this issue, we propose the following *restricted linear storage-retrieval policy* or *restricted linear decision rule*:

$$v_{ij}^t(\tilde{\mathbf{z}}^{t-1}) = v_{ij}^{t,0} + \sum_{k \in \mathcal{K}_{i,t-1}} v_{ij}^{t,k} z_k, \quad i \in \mathcal{M}, j \in \mathcal{N}; \quad (8)$$

$$w_{ij}^t(\tilde{\mathbf{z}}^t) = w_{ij}^{t,0} + \sum_{k \in \mathcal{K}_{i,t}} w_{ij}^{t,k} z_k, \quad i \in \mathcal{M}, j \in \mathcal{N}; \quad (9)$$

where $\mathcal{K}_{i,t} = \{k \in \mathcal{K}_t: d_i^{t,k} \neq 0\}$. Note that under the factor-based model, the demand for product i in period t can be written as $d_i^t(\tilde{\mathbf{z}}^t) = d_i^{t,0} + \sum_{k \in \mathcal{K}_{i,t}} d_i^{t,k} \tilde{z}_k$, $i \in \mathcal{M}, t \in \mathcal{T}$. In practice, most of the coefficients $d_i^{t,k}$ are zeros; thus, $|\mathcal{K}_{i,t}|$ is small compared with $|\mathcal{K}_t|$. For example, if demands are independent across products and periods, then $|\mathcal{K}_{i,t}| = 1$. To change from the linear decision rule to the restricted linear decision rule, some of the decision variables are forced to zeros. Specifically, the decisions $v_{ij}^t(\tilde{\mathbf{z}}^{t-1})$ and $w_{ij}^t(\tilde{\mathbf{z}}^t)$ under the restricted linear decision rule only respond to factors associated with the realized demand of product i and are not affected by factors of other products. Therefore, the restricted linear decision rule can result in a much smaller optimization problem and greatly improve scalability. Although the restricted linear decision rule may be inferior to the linear decision rule, the following result ensures that the restricted linear decision rule remains feasible in problem (3).

PROPOSITION 3. *Under Assumption U, there exists a restricted linear storage-retrieval policy in the form of Equations* (8) *and* (9) *that is feasible for problem* (3).

Table 1 Warehouse Example

(a) Warehouse layout

Class	Storage cost	Retrieval cost	Capacity
1	10	10	300
2	50	50	500
3	1,000	1,000	∞

(b) Number of arrivals and demand for each product in each period

	$t = 1$		$t = 2$	
i	a_i^1	$\mathbf{d}_i^1(\tilde{\mathbf{z}}^1)$	a_i^2	$\mathbf{d}_i^2(\tilde{\mathbf{z}}^2)$
1	300	$100 + \tilde{z}_1$	50	$50 + \tilde{z}_3$
2	300	$10 + \tilde{z}_2$	0	$200 + \tilde{z}_4$

PROOF. The proof of feasibility is the same as that of Theorem 1 and is omitted for brevity. □

Our numerical studies suggest that the restricted linear decision rule greatly extends the problem size we could handle. Furthermore, it significantly outperforms existing heuristics in the literature and achieves close to the expected value, given perfect demand information.

3.3. An Example

We illustrate our approach using a small example with two products and three classes. Table 1(a) shows the layout of the warehouse with class 3 representing emergency storage. We assume demand for each product in each period is independent of other products and periods. Table 1(b) gives a problem instance for a planning horizon of two periods. Assume the uncertain factors $\tilde{z}_1, \ldots, \tilde{z}_4$ have support set $W = \{\mathbf{z}: \|\mathbf{z}\|_\infty \leq 10\}$ and zero means so that $\hat{W} = \{\mathbf{0}\}$. It is easy to verify that there is enough inventory to meet demand for all $\mathbf{z} \in W$ in each period. Thus, we have $W \subseteq G$.

The robust counterpart $\mathbf{z}'\mathbf{y} \leq r$, $\forall \mathbf{z} \in W$ is simply $10 \sum_{k=1}^{4} |y_k| \leq r$, which can be easily transformed to linear constraints. The storage assignment problem under linear storage-retrieval policies is

$$\min \sum_{t=1}^{2} \sum_{i=1}^{2} \sum_{j=1}^{3} (s_j v_{ij}^{t,0} + r_j w_{ij}^{t,0}) \quad (10)$$

$$\text{s.t. } \sum_{j=1}^{3} v_{ij}^{t,k} = \begin{cases} a_i^t & \text{if } k = 0, \\ 0 & \text{otherwise,} \end{cases}$$

$$i = 1, 2, k = 0, \ldots, 4, t = 1, 2;$$

$$\sum_{j=1}^{3} w_{ij}^{t,k} = \begin{cases} d_i^{t,k} & \text{if } k \in \mathcal{K}_t^0, \\ 0 & \text{otherwise,} \end{cases}$$

$$i = 1, 2, k = 0, \ldots, 4, t = 1, 2;$$

$$x_{ij}^{t+1,k} = x_{ij}^{t,k} + v_{ij}^{t,k} - w_{ij}^{t,k}, \quad i = 1, 2, j = 1, \ldots, 3,$$

$$k = 0, \ldots, 4, t = 1, 2;$$

$$x_{ij}^1 = 0, \quad i = 1, 2, \; j = 1, \dots, 3;$$

$$\sum_{i=1}^{2}\left((x_{ij}^{t,0} + v_{ij}^{t,0}) + 10\sum_{k=1}^{4}|x_{ij}^{t,k} + v_{ij}^{t,k}| \right) \le c_j,$$
$$j = 1, 2, \; t = 1, 2;$$

$$v_{ij}^{t,0} - 10\sum_{k=1}^{4}|v_{ij}^{t,k}| \ge 0,$$
$$i = 1, 2, \; j = 1, \dots, 3, \; t = 1, 2;$$

$$w_{ij}^{t,0} - 10\sum_{k=1}^{4}|w_{ij}^{t,k}| \ge 0,$$
$$i = 1, 2, \; j = 1, \dots, 3, \; t = 1, 2;$$

$$x_{ij}^{t,0} - 10\sum_{k=1}^{4}|x_{ij}^{t,k}| \ge 0,$$
$$i = 1, 2, \; j = 1, \dots, 3, \; t = 1, \dots, 3;$$

$$v_{ij}^{t,k} = x_{ij}^{t,k} = 0, \quad i = 1, 2, \; j = 1, \dots, 3,$$
$$k = 2(t-1)+1, \dots, 4, \; t = 1, 2;$$

$$w_{ij}^{1,k} = 0, \quad i = 1, 2, \; j = 1, \dots, 3, \; k = 3, 4.$$

We only need to solve problem (10) once to obtain the optimal $v_{ij}^{t,k}$ and $w_{ij}^{t,k}$, which represent an optimal linear policy. Note that we can reduce the size of the problem by removing the variables that are assigned to zeros. The corresponding problem under restricted linear policies would require the variables $v_{1j}^{2,2}$, $v_{2j}^{2,1}$, $w_{1j}^{1,2}$, $w_{2j}^{1,1}$, $w_{1j}^{2,2}$, $w_{2j}^{2,1}$, $w_{1j}^{2,4}$, and $w_{2j}^{2,3}$ for all j to be set to zeros. In this example, the optimum solutions for both linear and restricted linear policies coincide.

Let $\mathbf{v}^{t,k}$ and $\mathbf{w}^{t,k}$ denote 2×3 matrices with $v_{ij}^{t,k}$ and $w_{ij}^{t,k}$ as their (i, j) entries respectively. Solving problem (10) gives

$$\mathbf{v}^{1,0} = \begin{pmatrix} 90 & 210 \\ 210 & 90 \\ 0 & 0 \end{pmatrix}', \quad \mathbf{v}^{1,k} = \mathbf{0}, \quad k = 1, \dots, 4. \quad (11)$$

The matrix $\mathbf{v}^{1,0}$ determines the number of pallets of product i assigned to class j in period 1. For example, there are 90, 210, and 0 pallets of product 1 assigned to classes 1, 2, and 3, respectively.

The solution to problem (10) also determines the coefficients $w_{ij}^{1,k}$ as follows:

$$\mathbf{w}^{1,0} = \begin{pmatrix} 90 & 0 \\ 10 & 10 \\ 0 & 0 \end{pmatrix}', \quad \mathbf{w}^{1,1} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}',$$

$$\mathbf{w}^{1,2} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}', \quad \mathbf{w}^{1,k} = \mathbf{0}, \quad k = 3, 4. \quad (12)$$

Given these coefficients, once demand in period 1 is realized, we can determine the retrieval decisions

for period 1 according to Equation (5): $w_{ij}^1(\tilde{\mathbf{z}}^1) = w_{ij}^{1,0} + w_{ij}^{1,1}\tilde{z}_1 + w_{ij}^{1,2}\tilde{z}_2$. Suppose in period 1 the realized demands for products 1 and 2 are 95 and 18, respectively. According to Table 1(b), the corresponding realized factors are $\tilde{\mathbf{z}}^1 = (\tilde{z}_1, \tilde{z}_2) = (-5, 8)$. Using product 1 for illustration, we have

$$\begin{pmatrix} w_{11}^1(\tilde{\mathbf{z}}^1) \\ w_{12}^1(\tilde{\mathbf{z}}^1) \\ w_{13}^1(\tilde{\mathbf{z}}^1) \end{pmatrix} = \begin{pmatrix} 90 \\ 10 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}(-5) + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}(8) = \begin{pmatrix} 90 \\ 5 \\ 0 \end{pmatrix}.$$

Thus, in period 1 we should retrieve 90, 5, and 0 pallets of product 1 from classes 1, 2, and 3, respectively.

Similarly, given the realized factors $\tilde{\mathbf{z}}^1$, we can determine the storage decisions for period 2 according to Equation (4): $v_{ij}^2(\tilde{\mathbf{z}}^1) = v_{ij}^{2,0} + v_{ij}^{2,1}\tilde{z}_1 + v_{ij}^{2,2}\tilde{z}_2$. From the solution of problem (10), we have

$$\mathbf{v}^{2,0} = \begin{pmatrix} 50 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}', \quad \mathbf{v}^{2,k} = \mathbf{0}, \quad k = 1, \dots, 4. \quad (13)$$

The storage decisions for product 1 in period 2 are

$$\begin{pmatrix} v_{11}^2(\tilde{\mathbf{z}}^1) \\ v_{12}^2(\tilde{\mathbf{z}}^1) \\ v_{13}^2(\tilde{\mathbf{z}}^1) \end{pmatrix} = \begin{pmatrix} 50 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}(-5) + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}(8) = \begin{pmatrix} 50 \\ 0 \\ 0 \end{pmatrix}.$$

Thus, we should store 50, 0, and 0 pallets of product 1 in classes 1, 2, and 3, respectively.

The coefficients $w_{ij}^{2,k}$ for the retrieval decisions are given as follows:

$$\mathbf{w}^{2,0} = \begin{pmatrix} 45 & 200 \\ 5 & 0 \\ 0 & 0 \end{pmatrix}', \quad \mathbf{w}^{2,k} = \mathbf{0}, \quad k = 1, 2,$$

$$\mathbf{w}^{2,3} = \begin{pmatrix} 0.5 & 0 \\ 0.5 & 0 \\ 0 & 0 \end{pmatrix}', \quad \mathbf{w}^{2,4} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}'. \quad (14)$$

We can determine the retrieval decisions for period 2 based on these coefficients according to Equation (5): $w_{ij}^2(\tilde{\mathbf{z}}^2) = w_{ij}^{2,0} + w_{ij}^{2,1}\tilde{z}_1 + w_{ij}^{2,2}\tilde{z}_2 + w_{ij}^{2,3}\tilde{z}_3 + w_{ij}^{2,4}\tilde{z}_4$. Suppose in period 2 the realized demands for products 1 and 2 are 56 and 198, respectively. According to Table 1(b), the corresponding realized factors are $\tilde{\mathbf{z}}^2 = (\tilde{z}_1, \tilde{z}_2, \tilde{z}_3, \tilde{z}_4) = (-5, 8, 6, -2)$. Using product 1 for illustration, we have

$$\begin{pmatrix} w_{11}^2(\tilde{\mathbf{z}}^2) \\ w_{12}^2(\tilde{\mathbf{z}}^2) \\ w_{13}^2(\tilde{\mathbf{z}}^2) \end{pmatrix} = \begin{pmatrix} 45 \\ 5 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}(-5) + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}(8)$$
$$+ \begin{pmatrix} 0.5 \\ 0.5 \\ 0 \end{pmatrix}(6) + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}(-2) = \begin{pmatrix} 48 \\ 8 \\ 0 \end{pmatrix}.$$

Thus, in period 2 we should retrieve 48, 8, and 0 pallets of product 1 from classes 1, 2, and 3, respectively.

The procedure of using the optimal restricted linear policy is summarized as follows: We first obtain the coefficients $v_{ij}^{t,k}$ and $w_{ij}^{t,k}$ by solving a linear optimization problem. After demand is realized in each period t, we derive the factors $\tilde{\mathbf{z}}^t$ and then use them to determine the storage and retrieval decisions according to Equations (8) and (9).

4. Implementation in Practice: A Case Study

We perform a case study with a major third-party logistics provider in Singapore to demonstrate the applicability of our method. The company owns a unit-load warehouse that provides storage services for its client. All other activities, such as demand forecasting, customer order processing, and production scheduling, are done by the client. All pallets are handled by a fleet of 15 forklifts. The warehouse pays its employees by the hour and charges the client by volume for storing and handling the products.

The warehouse operates in two shifts per day. It receives and puts away products during the day shift between 8:00 A.M. and 5:30 P.M. Sixty percent of the arriving pallets are from the client's manufacturing plant located nearby, and the rest are imported from foreign countries. The warehouse is informed one week in advance about the arrivals of pallets, with 98% accuracy. The day shift is supported by 30 employees who not only receive and put away pallets, but also batch customer orders that are transmitted from the client. All orders arriving during the day shift are retrieved in the following night shift, which has 10 employees working between 8:00 P.M. and 6:00 A.M. In our model, we set each period as a day so that the warehouse's business processes are in line with our assumption: All arriving pallets in each period are stored before any pallets are retrieved for demand occurring in the period.

Figure E.1(a) in Appendix E shows the warehouse's layout. The storage area is 90 meters wide with 10 aisles. Each aisle is 65 meters long. There are 18 single-deep racks. Every aisle has a rack on each side except the end aisles. Each rack contains 48 sections, and each section has four to five levels. Only one pallet can be stored in each level of a section. All levels of the same section belong to the same class because they have identical storage cost and identical retrieval cost. The warehouse is partitioned into 10 classes (thus, $N = 11$) with a grid shown in Figure E.1(a).

We have collected the actual numbers of arriving and departing pallets of each product in each period for 48 weeks. Figure 1(a) sorts the 410 products in the warehouse according to their annual demands. The first 122 products account for about 80% of the total annual demand. The warehouse operates six days a

Figure 1   Product Characteristics
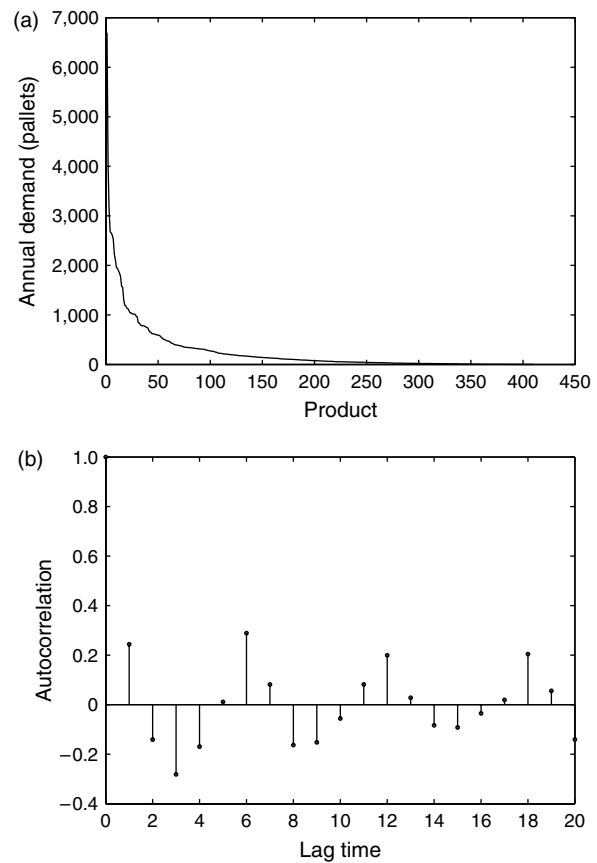
Notes. (a) The distribution of annual demands is strongly skewed. (b) The autocorrelation of the total daily demand suggests a weekly seasonality pattern.

week, so a natural choice for the length of planning horizon is six periods (that is, weekly planning). Figure 1(b) shows the autocorrelation of the total daily demand for all products over time. The peaks at lags 6, 12, and 18 suggest a weekly seasonality pattern.

We assume no correlation between different products. Using the factor-based demand model, we assume the demand for product i in period t is $d_i^t(\tilde{z}_i^t) = d_i^{t,0} + \tilde{z}_i^t$, where $d_i^{t,0}$ represents the sample mean of demand. We assume the uncertain factor $\tilde{z}_i^t$ falls in the range $[\max\{-d_i^{t,0}, -3\sigma_i^t\}, 3\sigma_i^t]$, where σ_i^t is the sample standard deviation of demand.[1]

We compare the restricted linear decision rule (RLR) with the static class-based turnover policy (TOS), the dynamic class-based turnover policy (TOD), and the class-based duration-of-stay policy (DOS). We evaluate the actual cost of each policy using the data given. The implementation of each policy requires the means and standard deviations of daily demands.

[1] Note that it is difficult to estimate the support of demand statistically. Nevertheless, in our computational studies, we found that the solution is rather insensitive to the size of the support set. Thus, we only report the results of this case.

To achieve this, we use the first 16 weeks of data to estimate these parameters in each period of week 17.[2] We then implement the policy for week 17 to evaluate its actual cost. Following a rolling-horizon principle, the demand parameters for week 18 can be estimated based on the data from week 2 to week 17, and so on.

For the RLR, we round any nonintegral v_{ij}^t and w_{ij}^t to the nearest integers. If the rounded solution does not match the actual number of arriving (departing) pallets, then we store (retrieve) any additional pallets in (from) the most economic class available. For example, if demand for a product is 20 but the rounded solution is to retrieve 19 pallets from class 1 and 0 from other classes, then we retrieve 1 more pallet from the most economic class that contains the product.

For the TOS policy, define the *static* turnover rate as $[\sum_{t=1}^{T}(a_i^t + d_i^{t,0})/\delta_i^t]/T$ for each product i, where δ_i^t is the average number of pallets of product i in period t. Products are ranked according to their static turnover rates, and the product with the highest static turnover rate is assigned to the most economic class. When a product is requested, it is retrieved from the most economic class containing the product. The TOD policy is similar to the TOS policy, except in each period t the former ranks products according to their *dynamic* turnover rates $(a_i^t + d_i^{t,0})/\delta_i^t$. For the DOS policy, we use the ADAPTIVE algorithm by Goetschalckx and Ratliff (1990). We also compare with a deterministic policy (DET), which is based on the optimal solution of problem (1). This policy is implemented with a rolling horizon using the realized demands of the past period and the mean demands of the coming periods. We implement all policies in Java programming language and solve the linear programs using CPLEX 10.2 on a personal computer with a 3.06 GHz Intel Core 2 Duo processor with 4 GB of SDRAM. For the RLR, it takes about 20 minutes to create a weekly plan, which is acceptable for the company.

The cost of each policy is computed based on the storage and retrieval costs of each class. Table 2 shows that the average daily cost under the RLR is significantly lower than that of other heuristics in this case study from week 17 to week 48. The table also shows that our method has the lowest 85%, 90%, and 95% quantiles of daily cost among all the policies. Figure 2 shows the percentage of days that have cost less than a value s under each policy. The daily cost of the RLR is first-order stochastically dominated by that of other heuristics. This strongly indicates that the RLR generally results in a lower daily cost than other heuristics.

[2] Because of the weekly seasonality pattern, we determine the mean $d_i^{t,0}$ and the standard deviation σ_i^t of demand based on the corresponding day of each of the previous weeks. For example, we use the actual demand for product i on Mondays of the first 16 weeks to estimate its mean demand on Monday of week 17.

Table 2   Daily Cost Profile Under Each Policy for the Case Study from Week 17 to Week 48

	Average daily cost	85% quantile of daily cost	90% quantile of daily cost	95% quantile of daily cost
RLR	9,220,164	15,388,911	15,832,972	17,573,184
TOS	10,014,759	16,367,066	17,157,081	18,968,993
TOD	10,044,606	16,437,146	17,240,289	18,993,355
DOS	9,951,695	16,415,298	16,883,752	18,855,040
DET	9,730,629	15,926,992	16,705,780	18,243,095

We will see that this superiority of the RLR in daily cost results in substantial savings in the long run.

We benchmark all policies against the *expected value given perfect information* (*EV* | *PI*), which is determined by resolving problem (1) every time we enter a new week. For example, to find *EV* | *PI* for week 17, we solve problem (1) with the actual demands in week 17. As we enter week 18, we resolve problem (1) with the actual demands in weeks 17 and 18 to obtain a new *EV* | *PI*, and so on. We define *percentage efficiency* of a policy as (*EV* | *PI*)/$Z \times 100\%$, where Z represents the cost given by the policy. An efficient policy would have high percentage efficiency. Figure 3(a) shows that the cumulative cost of the RLR is consistently lower than that of other heuristics and is close to *EV* | *PI*. Figure 3(b) shows that the RLR significantly outperforms other heuristics in percentage efficiency. For example, in week 48 it is about 8% more efficient than all the existing heuristics, which have similar costs. This strongly suggests that the RLR can generate substantial savings over other heuristics in the long run. Note that the DET policy is significantly worse than the RLR but more efficient than other heuristics. We also try different starting dates and durations to estimate the demand parameters and find that the RLR consistently outperforms other heuristics.

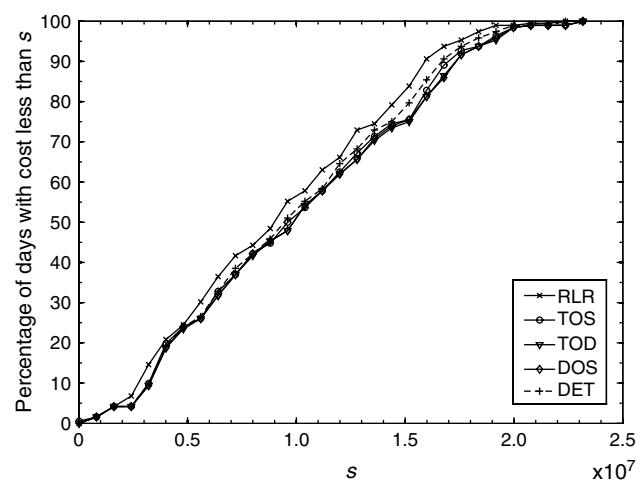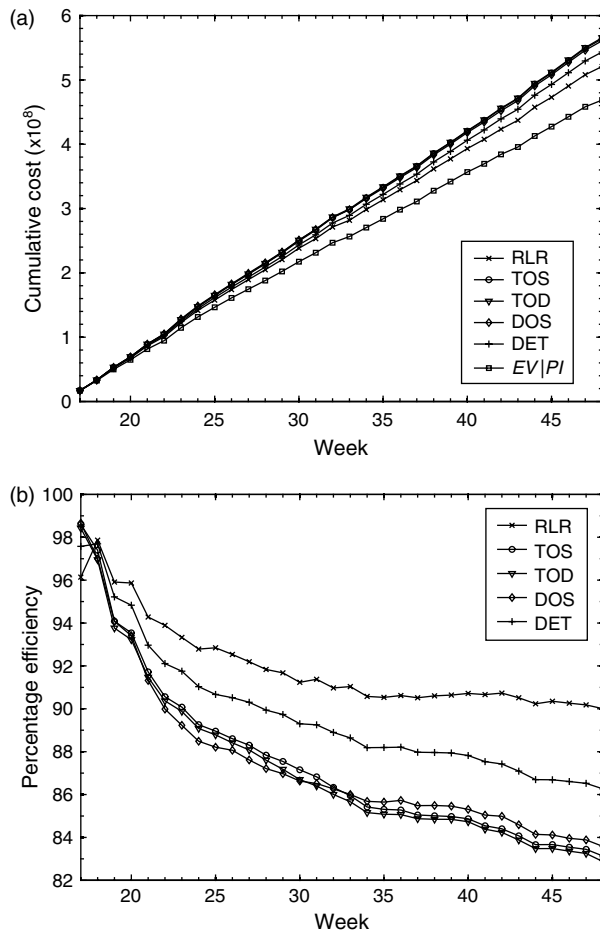**Figure 2**   Restricted Linear Decision Rule's Lower Daily Cost

Figure 3   Cumulative Costs and Percentage Efficiencies

(a)

(b)

Notes. (a) The RLR outperforms other heuristics in cumulative cost. (b) The RLR is significantly more efficient than other heuristics.

It is noteworthy that replenishments and fulfillments may occur simultaneously and continuously over time in other warehouses. We can approximate continuous time using shorter periods. To obtain a solution quickly, we can set a shorter planning horizon and solve the problem more frequently.

5. Numerical Studies on More General Cases

We test the performance of the policies in more general settings by considering a warehouse shown in Figure E.1(b) in Appendix E. It has 50 single-deep racks, and each rack contains 80 sections. Depending on the number of products in the warehouse, the number of levels in each section ranges from one to six. Only one pallet can be stored in each level of a section. We consider three layouts. The first is a U-flow layout shown in Figure E.1(b), in which the receiving dock R and shipping dock S coincide. The second and third layouts are shown in Figures E.1(c) and (d), respectively. The receiving and shipping docks in the latter two layouts are not located at the same point.

We perform simulations to compute the average cost of each policy for $T = 7$. We create 500 replications of arrival and demand data, which is sufficient to ensure that the standard error is within 2% of the average cost for each policy. To compute $EV \mid PI$, we first use the realized demand in each replication to solve problem (1) and then use the average cost over all replications as $EV \mid PI$. In the simulations we assume the uncertain factors $\tilde{z}_k$ follow a uniform distribution $U(-q, q)$, where q is a parameter. Thus, we have $W = \{\mathbf{z}: \|\mathbf{z}\|_\infty \le q\}$ and $\hat{W} = \{\mathbf{0}\}$. Table 3 shows the average cost and percentage efficiency of each policy for $q = 5$. The subscript of each cost value represents the standard error in percentage of the average cost. The fourth column shows the computational times of the RLR. The RLR significantly outperforms other heuristics and gives results close to $EV \mid PI$. The savings given by the RLR over other heuristics can be

Table 3   Average Cost and Percentage Efficiency of Each Policy Under Different Layouts

Layout	M	N	Time (sec)	Average cost ($\times 10^8$)/percentage efficiency (%)				
				RLR	TOS	TOD	DOS	DET
1	1,500	11	44,669	$30.2_{(0.29)}/91.0$	$45.3_{(0.27)}/60.6$	$34.8_{(0.71)}/79.1$	$35.8_{(0.29)}/76.8$	$31.5_{(0.19)}/87.3$
	600	19	19,793	$11.9_{(0.98)}/92.3$	$18.7_{(0.70)}/58.6$	$15.6_{(0.97)}/70.5$	$14.7_{(1.87)}/74.6$	$12.6_{(0.73)}/87.2$
	300	33	21,699	$8.83_{(0.68)}/92.1$	$14.6_{(0.77)}/55.7$	$10.5_{(0.66)}/77.5$	$10.4_{(0.67)}/77.9$	$9.36_{(1.69)}/86.9$
	80	111	17,387	$2.11_{(0.27)}/97.5$	$2.84_{(0.08)}/72.5$	$2.23_{(0.80)}/92.3$	$2.53_{(1.14)}/81.5$	$2.23_{(0.16)}/92.4$
2	1,500	11	45,718	$30.1_{(0.57)}/91.2$	$44.9_{(0.32)}/61.1$	$34.4_{(0.17)}/79.6$	$35.6_{(0.01)}/77.0$	$31.2_{(0.37)}/88.0$
	600	19	15,498	$11.4_{(0.79)}/90.7$	$18.5_{(1.20)}/55.8$	$15.2_{(1.75)}/68.0$	$14.4_{(1.20)}/71.6$	$12.1_{(0.86)}/85.9$
	300	33	25,353	$8.85_{(0.94)}/91.8$	$14.5_{(0.86)}/56.1$	$10.4_{(0.54)}/77.9$	$10.4_{(0.47)}/78.1$	$9.23_{(0.76)}/88.0$
	80	111	17,633	$2.14_{(0.64)}/97.3$	$2.81_{(0.49)}/74.1$	$2.24_{(0.98)}/92.8$	$2.53_{(0.74)}/82.3$	$2.22_{(0.10)}/93.5$
3	1,500	11	43,319	$30.7_{(0.01)}/92.5$	$45.7_{(0.20)}/62.1$	$35.4_{(0.57)}/80.2$	$36.5_{(1.24)}/77.9$	$32.5_{(0.16)}/87.6$
	600	19	17,150	$12.5_{(1.03)}/93.1$	$19.0_{(1.90)}/61.4$	$16.0_{(0.64)}/72.8$	$15.2_{(1.13)}/76.6$	$13.1_{(1.51)}/88.5$
	300	33	19,166	$9.10_{(0.04)}/92.3$	$14.8_{(0.01)}/56.9$	$10.7_{(1.46)}/78.4$	$10.7_{(1.29)}/78.8$	$9.58_{(0.70)}/87.7$
	80	111	17,470	$2.21_{(0.85)}/97.0$	$2.9_{(1.40)}/73.8$	$2.31_{(1.82)}/92.5$	$2.59_{(1.73)}/82.6$	$2.32_{(0.92)}/92.4$

as high as 36% (for layout 1, $M = 300$, and $N = 33$). In general, the DET policy outperforms the TOD and DOS policies, which outperform the TOS policy. An extensive set of experiments suggests that the RLR is consistently better than other heuristics and its costs are very close to $EV \mid PI$ for wide ranges of N, M, q, T, and different class generation methods. We do not include all the results here because of space limitations. Interested readers can refer to supplementary materials at http://www.mysmu.edu/faculty/yflim/sup.pdf.

6. Understanding the Policies

To explain the differences between the policies, we use the example in §3.3 to compare their decisions. Consider an extreme case, where $\tilde{z}_k = -10$, for $k = 1, \ldots, 4$. The realized demands in periods 1 and 2 are $(90, 0)$ and $(40, 190)$, respectively. Table 4 shows the inventory level of each product in each class after storage and retrieval are done at the start and the end, respectively, of each period under each policy. The costs of the storage and retrieval decisions associated with each class are also shown.

Under the TOS policy, product 2 has higher priority than product 1. Consequently, all arriving pallets of product 2 are stored in class 1 at the start of period 1. On the contrary, the TOD policy ranks product 1 higher in period 1 and stores all its arriving pallets in class 1. Both policies assign all locations of the most economic class to a single product in period 1. This strategy is myopic because variability in arrivals and demands means the storage and retrieval activities in the subsequent periods may

only involve the other product, which is stored at locations with higher costs. To absorb the variability, we should share the most economic storage locations between different products. The DOS policy partially addresses this issue by evenly allocating the locations of class 1 to the two products in period 1. However, as we can see from Table 4, this policy does not give the best solution.

Assuming that we know the above realized demands for products 1 and 2, what is a good policy to store and retrieve pallets? To minimize cost, one should store at least 90 pallets of product 1 in class 1 at the start of period 1 such that demand for this product in period 1 can be fully satisfied by class 1. This also ensures sufficient space in class 1 to accommodate all arrivals of product 1 in period 2. In contrast, because there is no arrival of product 2 in period 2, one should store at least 190 pallets of product 2 in class 1 at the start of period 1. Using this initial allocation, except the storage in period 1, class 2 is not involved for the rest of the planning horizon. All policies in Table 4 fail to satisfy this initial allocation requirement except the RLR, which stores 90 and 210 pallets of products 1 and 2, respectively, to class 1 in period 1. This indicates the superiority of the RLR over other heuristics.

Both the TOS and DOS policies consider only the aggregated arrival and demand information for each product, whereas the TOD policy relies on the detailed information in each individual period. In contrast, the RLR stores and retrieves pallets according to coefficients in Equations (11)–(14), which are obtained by solving problem (10). This

Table 4   **Inventory Levels and Costs of Each Class Under Different Policies**

Product\class	t = 1 Start 1	2	3	t = 1 End 1	2	3	t = 2 Start 1	2	3	t = 2 End 1	2	3	Total cost
TOS													
1	0	300	0	0	210	0	0	260	0	0	220	0	
2	300	0	0	300	0	0	300	0	0	110	0	0	
Cost	3,000	15,000	0	0	4,500	0	0	2,500	0	1,900	2,000	0	28,900
TOD													
1	300	0	0	210	0	0	260	0	0	220	0	0	
2	0	300	0	0	300	0	0	300	0	0	110	0	
Cost	3,000	15,000	0	900	0	0	500	0	0	400	9,500	0	29,300
DOS													
1	150	150	0	60	150	0	110	150	0	70	150	0	
2	150	150	0	150	150	0	150	150	0	0	110	0	
Cost	3,000	15,000	0	900	0	0	500	0	0	1,900	2,000	0	23,300
RLR													
1	90	210	0	0	210	0	50	210	0	10	210	0	
2	210	90	0	210	90	0	210	90	0	20	90	0	
Cost	3,000	15,000	0	900	0	0	500	0	0	2,300	0	0	21,700

optimization problem considers both aggregated information (over the entire planning horizon) and detailed information (in each period). In addition, it also takes the capacity constraint of each class and demand uncertainty into account. As we can see in Table 4, the RLR indeed gives the lowest total cost in this example.

7. Conclusions

Minimizing travel in a unit-load warehouse is non-trivial because replenishments may not follow a simple rule, as they typically depend on the production plans and status of the suppliers. The problem is further complicated by the fact that products face uncertain demand over multiple periods. It is therefore very challenging to find an efficient storage-retrieval policy for the warehouse.

Existing heuristics in the literature such as the turnover-based and the duration-of-stay-based policies do not consider variability of both inflow and outflow of products. Furthermore, these heuristics neglect the capacity of each storage class in the warehouse. Although these heuristics are easier to implement in practice, our results suggest that remarkable savings can be obtained by taking into account the variability of product flow and the capacity constraints of storage classes.

To handle variability of product flow, we consider a factor-based demand model in which demand for each product in each period is affinely dependent on some uncertain factors. We only require information on the mean and the bounds of each uncertain factor. We formulate the storage assignment problem in a unit-load warehouse as a robust optimization problem that minimizes the worst-case expected total cost subject to the capacity constraints of storage classes. By limiting to restricted linear decision rules, we obtain a storage-retrieval policy by solving a moderate-size linear program.

A case study with a logistics company suggests that the restricted linear storage-retrieval policy can be obtained in a reasonable amount of time for weekly planning and it generates substantial savings over other heuristics. A detailed analysis on daily costs shows that the restricted linear policy, on average, significantly outperforms other heuristics. The case study strongly supports the claim that our method is implementable and promising for practical use.

The restricted linear policy outperforms the class-based turnover and the class-based duration-of-stay policies in *all* numerical experiments that we conduct based on realistic warehouse settings. In some experiments, the savings by the restricted linear policy can be as high as 36%. Surprisingly, despite imprecise specification on demand distributions, the restricted linear policy attains close to the expected value, given perfect information (a lower bound of the optimal expected cost) in many cases.

A detailed observation of the storage and retrieval decisions reveals that the restricted linear policy outperforms other heuristics by carefully allocating the storage space of economic classes to different products. This is accomplished by considering variable arrivals and stochastic demands for products and the capacity constraints of storage classes, which are ignored by the existing heuristics.

Our model can be generalized to a warehouse with multiple receiving and multiple shipping docks as follows: Change a_i^t to $a_{i,\nu}^t$, where $a_{i,\nu}^t$ represents the number of arriving pallets of product i at dock ν in period t. Likewise, we can do so for the demand d_i^t and the decision variables v_{ij}^t and w_{ij}^t.

Although our numerical results are based on a storage class generation method that partitions the warehouse using a grid, further numerical studies reveal that the performance of the restricted linear policy relative to other heuristics is not sensitive to the class generation method used. Details can be found at http://www.mysmu.edu/faculty/yflim/sup.pdf.

Finally, we emphasize that the computational time to solve large linear programs could be further reduced in the future through continuing improvements in linear optimization. We would also like to highlight that although the formulation of robust optimization models can be rather tedious, it has been made easier because of the availability of software such as ROME (Goh and Sim 2011) and AIMMS (http://www.aimms.com/).

Appendix A. Proof of Proposition 1

($\Rightarrow$) Suppose that (1) is feasible; then there exists a feasible solution $\mathbf{x}$, $\mathbf{v}$, $\mathbf{w}$ such that

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^t - w_{ij}^t, \quad i \in \mathcal{M}, j \in \mathcal{N}, t \in \mathcal{T}. \quad (A1)$$

From the third constraint when $t = 1$ and summing all the jth terms for $j \in \mathcal{N}$, we have

$$\sum_{j \in \mathcal{N}} x_{ij}^2 = \sum_{j \in \mathcal{N}} (x_{ij}^1 + v_{ij}^1 - w_{ij}^1)$$

$$= a_i^1 - d_i^1 \text{ (from the first and second constraint of (1)).}$$

$$(A2)$$

From the constraint $x_{ij}^t \geq 0$, $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}^+$, we conclude that inequality (2) is true for $t = 1$. Similarly, for $t = 2, \ldots, T$, we have

$$\sum_{\tau=1}^{t} \sum_{j \in \mathcal{N}} x_{ij}^{\tau+1} = \sum_{\tau=1}^{t} \sum_{j \in \mathcal{N}} (x_{ij}^{\tau} + v_{ij}^{\tau} - w_{ij}^{\tau})$$

$$= \sum_{\tau=1}^{t} \sum_{j \in \mathcal{N}} x_{ij}^{\tau} + \sum_{\tau=1}^{t} \sum_{j \in \mathcal{N}} (v_{ij}^{\tau} - w_{ij}^{\tau})$$

$$= \sum_{\tau=1}^{t} \sum_{j \in \mathcal{N}} x_{ij}^{\tau} + \sum_{\tau=1}^{t} (a_i^{\tau} - d_i^{\tau}). \qquad (A3)$$

Hence,

$$\sum_{\tau=1}^{t} (a_i^{\tau} - d_i^{\tau}) = \sum_{\tau=1}^{t} \sum_{j \in \mathcal{N}} x_{ij}^{\tau+1} - \sum_{\tau=2}^{t} \sum_{j \in \mathcal{N}} x_{ij}^{\tau}$$

$$= \sum_{j \in \mathcal{N}} x_{ij}^{t+1}$$

$$\geq 0. \qquad (A4)$$

Similarly, the last inequality is deduced from the nonnegativity of x_{ij}^t for $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}^+$. Thus, we proved inequality (2).

($\Leftarrow$) Suppose that $\sum_{\tau=1}^{t} a_i^{\tau} - \sum_{\tau=1}^{t} d_i^{\tau} \geq 0$ for $i \in \mathcal{M}$, $t \in \mathcal{T}$. Using the assumption that there is an emergency storage class, N, with infinite storage, the fifth constraint (capacity constraint) is satisfied; we have

$$v_{ij}^t = \begin{cases} a_i^t & \text{for } j = N, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$w_{ij}^t = \begin{cases} d_i^t & \text{for } j = N, \\ 0 & \text{otherwise,} \end{cases}$$

for $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}$. Hence, $v_{ij}^t, w_{ij}^t \geq 0$ for $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}$. We now prove the nonnegativity of x_{ij}^t. For $j \neq N$, it is clear that $x_{ij}^t = 0$ for all $i \in \mathcal{M}$, $t \in \mathcal{T}$. We now prove the case $j = N$. For $t = 1$,

$$x_{iN}^2 = x_{iN}^1 + v_{iN}^1 - w_{iN}^1$$

$$= a_i^1 - d_i^1$$

$$\geq 0 \text{ (from inequality (2) when } \tau = 1\text{).}$$

For any $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}$,

$$x_{iN}^{t+1} = x_{ij}^t + v_{iN}^t - w_{iN}^t$$

$$= x_{iN}^{t-1} + \sum_{\tau=t-1}^{t} (v_{iN}^{\tau} - w_{iN}^{\tau})$$

$$= x_{iN}^{t-1} + \sum_{\tau=t-1}^{t} (a_i^{\tau} - d_i^{\tau})$$

$$= x_{iN}^{t-2} + \sum_{\tau=t-2}^{t} (a_i^{\tau} - d_i^{\tau})$$

$$= \cdots$$

$$= x_{iN}^1 + \sum_{\tau=1}^{t} (a_i^{\tau} - d_i^{\tau})$$

$$\geq 0 \text{ (from inequality (2)).}$$

Inductively, $x_{iN}^t \geq 0$, , $i \in \mathcal{M}$, $t \in \mathcal{T}$. Thus, $x_{ij}^t \geq 0$, $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}$. $\square$

Appendix B. Proof of Theorem 1

Under Assumption U, we have $\tilde{z}$ with support set W. For $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}$, let

$$v_{ij}^t(\mathbf{z}^{t-1}) = \begin{cases} a_i^t & \text{for } j = N \\ 0 & \text{otherwise,} \end{cases}$$

and

$$w_{ij}^t(\mathbf{z}^t) = \begin{cases} d_i^t(\mathbf{z}^t) & \text{for } j = N \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathbf{z} \in W$. Under the uncertainty set W, we have $d_i^t(\mathbf{z}^t) \geq 0$; hence, $w_{ij}^t(\mathbf{z}^t) \geq 0$. The nonnegativity of $v_{ij}^t(\mathbf{z}^{t-1})$ follows from the nonnegativity of a_i^t. We now prove the constraint $x_{ij}^{t+1}(\mathbf{z}^t) \geq 0$. For $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}$,

$$x_{ij}^{t+1}(\mathbf{z}^t) = x_{ij}^t(\mathbf{z}^{t-1}) + v_{ij}^t(\mathbf{z}^{t-1}) - w_{ij}^t(\mathbf{z}^t)$$

$$= \begin{cases} x_{ij}^t(\mathbf{z}^{t-1}) + a_i^t - d_i^t(\mathbf{z}^t) & \text{for } j = N, \\ x_{ij}^t(\mathbf{z}^{t-1}) & \text{otherwise.} \end{cases}$$

Since $x_{ij}^1 = 0$, it is clear that $x_{ij}^{t+1}(\mathbf{z}^t) = 0$ for $j \neq N$. For $i \in \mathcal{M}$, $j = N$ and $t \in \mathcal{T}$,

$$x_{iN}^{t+1}(\mathbf{z}^t) = x_{iN}^t(\mathbf{z}^{t-1}) + v_{iN}^t(\mathbf{z}^{t-1}) - w_{iN}^t(\mathbf{z}^t)$$

$$= x_{iN}^t(\mathbf{z}^{t-1}) + a_N^t - d_N^t(\mathbf{z}^t)$$

$$= x_{iN}^{t-1}(\mathbf{z}^{t-2}) + \sum_{\tau=t-1}^{t} (a_N^{\tau} - d_N^{\tau}(\mathbf{z}^{\tau}))$$

$$= \cdots$$

$$= x_{iN}^1 + \sum_{\tau=1}^{t} (a_N^{\tau} - d_N^{\tau}(\mathbf{z}^{\tau}))$$

$$\geq 0 \text{ (since } \mathbf{z} \in W\text{).}$$

Thus, $x_{ij}^{t+1}(\mathbf{z}^t) \geq 0$ for $i \in \mathcal{M}$, $j \in \mathcal{N}$, $t \in \mathcal{T}$, and we have found a feasible solution for Z_{LR}. It is clear that the objective value is also finite.

Using the linear decision rule on the objective function of (6), we have

$$\sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^t(\mathbf{z}^{t-1}) + r_j w_{ij}^t(\mathbf{z}^t))$$

$$= \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} \left(\left(s_j v_{ij}^{t,0} + \sum_{k \in \mathcal{K}_{t-1}} s_j v_{ij}^{t,k} z_k \right) \right.$$

$$\left. + \left(r_j w_{ij}^{t,0} + \sum_{k \in \mathcal{K}_t} r_j w_{ij}^{t,k} z_k \right) \right)$$

$$= \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} \left((s_j v_{ij}^{t,0} + r_j w_{ij}^{t,0}) + \sum_{k \in \mathcal{K}_T} (s_j v_{ij}^{t,k} + r_j w_{ij}^{t,k}) z_k \right),$$

in which $v_{ij}^{t,k} = 0$ for $i \in \mathcal{M}$, $j \in \mathcal{N}$, $k \in \mathcal{K}_T \backslash \mathcal{K}_t$, $t \in \mathcal{T}$ and $w_{ij}^{t,k} = 0$ for $i \in \mathcal{M}$, $j \in \mathcal{N}$, $k \in \mathcal{K}_T \backslash \mathcal{K}_{t-1}$, $t \in \mathcal{T}$. Setting $g^k = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^{t,k} + r_j w_{ij}^{t,k})$ for $k \in \mathcal{K}_T^0$, the objective function becomes

$$\min_{z \in \hat{W}} \max \left(g^0 + \sum_{k \in \mathcal{K}_T} g^k z_k \right) = \min \left(g^0 + \max_{z \in \hat{W}} \sum_{k \in \mathcal{K}_T} g^k z_k \right).$$

For $i \in \mathcal{M}$,

$$\sum_{j \in \mathcal{N}} v_{ij}^t(\tilde{\mathbf{z}}^{t-1}) = a_i^t, \quad t \in \mathcal{T}$$

$$\Leftrightarrow \sum_{j \in \mathcal{N}} \left(v_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} v_{ij}^{t,k} z_k \right) = a_i^t \quad \forall \mathbf{z} \in W, \, t \in \mathcal{T}$$

$$\Leftrightarrow \sum_{j \in \mathcal{N}} v_{ij}^{t,k} = \begin{cases} a_i^t & \text{if } k = 0, \\ 0 & \text{otherwise,} \end{cases} \quad k \in \mathcal{K}_T^0, \, t \in \mathcal{T}.$$

The last equivalence holds because the set W is full dimensional. Similarly, for $i \in \mathcal{M}$,

$$\sum_{j \in \mathcal{N}} w_{ij}^t(\tilde{\mathbf{z}}^t) = d_i^t(\tilde{\mathbf{z}}^t), \quad t \in \mathcal{T}$$

$$\Leftrightarrow \sum_{j \in \mathcal{N}} \left(w_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} w_{ij}^{t,k} z_k \right)$$
$$= d_i^{t,0} + \sum_{k \in \mathcal{K}_T} d_i^{t,k} z_k \quad \forall \mathbf{z} \in W, \, t \in \mathcal{T}.$$

$$\Leftrightarrow \sum_{j \in \mathcal{N}} w_{ij}^{t,k} = d_i^{t,k}, \quad k \in \mathcal{K}_T^0, \, t \in \mathcal{T}.$$

For $i \in \mathcal{M}, j \in \mathcal{N}$,

$$x_{ij}^{t+1}(\tilde{\mathbf{z}}^t) = x_{ij}^t(\tilde{\mathbf{z}}^{t-1}) + v_{ij}^t(\tilde{\mathbf{z}}^{t-1}) - w_{ij}^t(\tilde{\mathbf{z}}^t), \quad t \in \mathcal{T}$$

$$\Leftrightarrow x_{ij}^{t+1,0} + \sum_{k \in \mathcal{K}_t} x_{ij}^{t+1,k} z_k$$

$$= \left(x_{ij}^{t,0} + \sum_{k \in \mathcal{K}_t} x_{ij}^{t,k} z_k \right) + \left(v_{ij}^{t,0} + \sum_{k \in \mathcal{K}_{t-1}} v_{ij}^{t,k} z_k \right)$$

$$- \left(w_{ij}^{t,0} + \sum_{k \in \mathcal{K}_t} w_{ij}^{t,k} z_k \right), \quad t \in \mathcal{T}$$

$$\Leftrightarrow x_{ij}^{t+1,0} + \sum_{k \in \mathcal{K}_T} x_{ij}^{t+1,k} z_k$$

$$= \left(x_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} x_{ij}^{t,k} z_k \right) + \left(v_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} v_{ij}^{t,k} z_k \right)$$

$$- \left(w_{ij}^{t,0} + \sum_{k \in \mathcal{K}_T} w_{ij}^{t,k} z_k \right), \quad t \in \mathcal{T}$$

$$\Leftrightarrow x_{ij}^{t+1,k} = x_{ij}^{t,k} + v_{ij}^{t,k} - w_{ij}^{t,k}, \quad k \in \mathcal{K}_T^0, \, t \in \mathcal{T}.$$

For $i \in \mathcal{M}, j \in \mathcal{N}$, since $v_{ij}^{t,k} = 0, w_{ij}^{t,k} = 0$ for $k \in \mathcal{K}_T \backslash \mathcal{K}_t, t \in \mathcal{T}$, we have $x_{ij}^{t+1,k} = x_{ij}^{t,k}$ for $k \in \mathcal{K}_T \backslash \mathcal{K}_t, t \in \mathcal{T}$. From the constraint $x_{ij}^1 = 0$ for $i \in \mathcal{M}, j \in \mathcal{N}$, we conclude that $x_{ij}^{t,k} = 0$ for $i \in \mathcal{M}, j \in \mathcal{N}, k \in \mathcal{K}_T \backslash \mathcal{K}_t, t \in \mathcal{T}$.

By the definition that $h_j^{t,k} = \sum_{i \in \mathcal{M}} x_{ij}^{t,k} + v_{ij}^{t,k}$, for $j \in \mathcal{N}, k \in \mathcal{K}_T^0, t \in \mathcal{T}$, we can draw the similar deduction that $h_j^{t,k} = 0$ for $j \in \mathcal{N}, k \in \mathcal{K}_T \backslash \mathcal{K}_t, t \in \mathcal{T}$.

Finally, for inequality constraints involving linear decision rule, we note that given $y \in \mathcal{L}_{K_t}$ the constraint

$$y(\tilde{\mathbf{z}}^t) \leq r$$

is equivalent the following robust counterpart

$$y^0 + \sum_{k \in \mathcal{K}_T} y^k z_k \leq r, \quad \forall \mathbf{z} \in W.$$

Hence, we prove that under the linear storage-retrieval policy, we have problem (7). □

Appendix C. Proof of Proposition 2
Note that with

$$\mathbf{d}^{t,0} = \begin{pmatrix} d_1^{t,0} \\ \vdots \\ d_M^{t,0} \end{pmatrix}, \quad \mathbf{D}^t = \begin{pmatrix} d_1^{t,1} & \cdots & d_1^{t,K_t} & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & 0 & \cdots & 0 \\ d_M^{t,1} & \cdots & d_M^{t,K_t} & 0 & \cdots & 0 \end{pmatrix} \in \mathfrak{R}^{M \times K_T},$$

we can concisely represent the vector of uncertain demands as

$$\mathbf{d}^t(\tilde{\mathbf{z}}^t) = \begin{pmatrix} \mathbf{d}_1^t(\tilde{\mathbf{z}}^t) \\ \vdots \\ \mathbf{d}_M^t(\tilde{\mathbf{z}}^t) \end{pmatrix} = \mathbf{d}^{t,0} + \mathbf{D}^t \mathbf{z} \quad \text{for } t \in \mathcal{T},$$

where $\mathbf{z} \triangleq \mathbf{z}^T$. Hence, we can express the factor support set, W as follows:

$$W = \left\{ \mathbf{z} \,\middle|\, \exists \mathbf{u}: \sum_{\tau=1}^t \mathbf{d}^\tau(\mathbf{z}^\tau) \leq \sum_{\tau=1}^t \mathbf{a}^\tau, \mathbf{d}^t(\mathbf{z}^t) \geq \mathbf{0}, t \in \mathcal{T}, \mathbf{Az} + \mathbf{Bu} \leq \mathbf{q} \right\}$$

$$= \left\{ \mathbf{z} \,\middle|\, \exists \mathbf{u}: \sum_{\tau=1}^t \mathbf{D}^\tau \mathbf{z} \leq \sum_{\tau=1}^t (\mathbf{a}^\tau - \mathbf{d}^{\tau,0}), \mathbf{d}^{t,0} + \mathbf{D}^t \mathbf{z} \geq \mathbf{0}, \right.$$
$$\left. t \in \mathcal{T}, \mathbf{Az} + \mathbf{Bu} \leq \mathbf{q} \right\}$$

$$= \{ \mathbf{z} \mid \exists \mathbf{u}: \bar{\mathbf{D}}^t \mathbf{z} \leq \bar{\mathbf{a}}^t, -\mathbf{D}^t \mathbf{z} \leq \mathbf{d}^{t,0}, t \in \mathcal{T}, \mathbf{Az} + \mathbf{Bu} \leq \mathbf{q} \},$$

where $\mathbf{a}^t = (a_1^t \cdots a_M^t)$, $\bar{\mathbf{a}}^t = \sum_{\tau=1}^t (\mathbf{a}^\tau - \mathbf{d}^{\tau,0})$, and $\bar{\mathbf{D}}^t = \sum_{\tau=1}^t \mathbf{D}^\tau$. To obtain the robust counterpart, $\max_{\mathbf{z} \in W} \mathbf{z}'\mathbf{y} \leq y$, we note that by strong linear programming duality, the primal problem

$$\max_{u, z} \quad \mathbf{y}'\mathbf{z}$$
$$\text{s.t.} \quad \bar{\mathbf{D}}^t \mathbf{z} \leq \bar{\mathbf{a}}^t, \quad t \in \mathcal{T}$$
$$-\mathbf{D}^t \mathbf{z} \leq \mathbf{d}^{t,0}, \quad t \in \mathcal{T}$$
$$\mathbf{Az} + \mathbf{Bu} \leq \mathbf{q}$$

has the same objective value as the following dual problem:

$$\min \quad \sum_{t \in \mathcal{T}} (\bar{\mathbf{a}}^{t'} \boldsymbol{\alpha}^t + \mathbf{d}^{t,0'} \boldsymbol{\beta}^t) + \boldsymbol{\gamma}'\mathbf{q}$$
$$\text{s.t.} \quad \sum_{t \in \mathcal{T}} (\bar{\mathbf{D}}^{t'} \boldsymbol{\alpha}^t - \mathbf{D}^{t'} \boldsymbol{\beta}^t) + \mathbf{A}'\boldsymbol{\gamma} = \mathbf{y}$$
$$\mathbf{B}'\boldsymbol{\gamma} = \mathbf{0}$$
$$\boldsymbol{\gamma} \geq \mathbf{0}, \quad \boldsymbol{\beta}^t, \boldsymbol{\alpha}^t \geq \mathbf{0}, \quad t \in \mathcal{T}.$$

Hence, the robust counterpart, $\max_{\mathbf{z} \in W} \mathbf{z}'\mathbf{y} \leq y$ is feasible if and only if there exists $\boldsymbol{\gamma}, \boldsymbol{\alpha}^t, \boldsymbol{\beta}^t, t \in \mathcal{T}$ feasible in the dual problem and that $\sum_{t \in \mathcal{T}} (\bar{\mathbf{a}}^{t'} \boldsymbol{\alpha}^t + \mathbf{d}^{t,0'} \boldsymbol{\beta}^t) + \boldsymbol{\gamma}'\mathbf{q} \leq r$. □

Appendix D. Formulation for Computing the Optimal Linear Policy

$$\min \quad g^0 + \max_{\mathbf{z} \in \hat{W}} \sum_{k \in \mathcal{K}_t} g^k z_k$$

$$\text{s.t.} \quad g^k = \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{M}} \sum_{j \in \mathcal{N}} (s_j v_{ij}^{t,k} + r_j w_{ij}^{t,k}), \quad k \in \mathcal{K}_T^0;$$

$$\sum_{j \in \mathcal{N}} v_{ij}^{t,k} = \begin{cases} a_i^t & \text{if } k = 0, \\ 0 & \text{otherwise,} \end{cases} \quad i \in \mathcal{M}, k \in \mathcal{K}_T, t \in \mathcal{T};$$

$$\sum_{j\in\mathcal{N}} w_{ij}^{t,k} = d_i^{t,k}, \quad i\in\mathcal{M}, k\in\mathcal{K}_T^0, t\in\mathcal{T};$$

$$x_{ij}^{t+1,k} = x_{ij}^{t,k} + v_{ij}^{t,k} - w_{ij}^{t,k}, \quad i\in\mathcal{M}, j\in\mathcal{N}, k\in\mathcal{K}_T^0, t\in\mathcal{T};$$

$$x_{ij}^1 = 0, \quad i\in\mathcal{M}, j\in\mathcal{N};$$

$$h_j^{t,k} = \sum_{i\in\mathcal{M}} (x_{ij}^{t,k} + v_{ij}^{t,k}), \quad j\in\mathcal{N}, k\in\mathcal{K}_T^0, t\in\mathcal{T};$$

$$\sum_{\tau\in\mathcal{T}} (\bar{\mathbf{a}}^{\tau\prime}\boldsymbol{\alpha}_{h_j^t}^\tau + \mathbf{d}^{\tau,0\prime}\boldsymbol{\beta}_{h_j^t}^\tau) + \boldsymbol{\gamma}'_{h_j^t}\mathbf{q} \le c_j - h_j^{t,0}, \quad j\in\mathcal{N}^-, t\in\mathcal{T};$$

$$\sum_{\tau\in\mathcal{T}} (\bar{\mathbf{D}}^{\tau\prime}\boldsymbol{\alpha}_{h_j^t}^\tau - \mathbf{D}^{\tau\prime}\boldsymbol{\beta}_{h_j^t}^\tau) + \mathbf{A}'\boldsymbol{\gamma}_{h_j^t} = \mathbf{h}_j^{t,K_T}, \quad j\in\mathcal{N}^-, t\in\mathcal{T};$$

$$\mathbf{B}'\boldsymbol{\gamma}_{h_j^t} = \mathbf{0}, \quad j\in\mathcal{N}^-, t\in\mathcal{T};$$

$$\sum_{\tau\in\mathcal{T}} (\bar{\mathbf{a}}^{\tau\prime}\boldsymbol{\alpha}_{v_{ij}^t}^\tau + \mathbf{d}^{\tau,0\prime}\boldsymbol{\beta}_{v_{ij}^t}^\tau) + \boldsymbol{\gamma}'_{v_{ij}^t}\mathbf{q} \le -v_{ij}^{t,0}, \quad i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\sum_{\tau\in\mathcal{T}} (\bar{\mathbf{D}}^{\tau\prime}\boldsymbol{\alpha}_{v_{ij}^t}^\tau - \mathbf{D}^{\tau\prime}\boldsymbol{\beta}_{v_{ij}^t}^\tau) + \mathbf{A}'\boldsymbol{\gamma}_{v_{ij}^t} = \mathbf{v}_{ij}^{t,K_T},$$
$$i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\mathbf{B}'\boldsymbol{\gamma}_{v_{ij}^t} = \mathbf{0}, \quad i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\sum_{\tau\in\mathcal{T}} (\bar{\mathbf{a}}^{\tau\prime}\boldsymbol{\alpha}_{w_{ij}^t}^\tau + \mathbf{d}^{\tau,0\prime}\boldsymbol{\beta}_{w_{ij}^t}^\tau) + \boldsymbol{\gamma}'_{w_{ij}^t}\mathbf{q} \le -w_{ij}^{t,0},$$
$$i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\sum_{\tau\in\mathcal{T}} (\bar{\mathbf{D}}^{\tau\prime}\boldsymbol{\alpha}_{w_{ij}^t}^\tau - \mathbf{D}^{\tau\prime}\boldsymbol{\beta}_{w_{ij}^t}^\tau) + \mathbf{A}'\boldsymbol{\gamma}_{w_{ij}^t} = \mathbf{w}_{ij}^{t,K_T},$$
$$i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\mathbf{B}'\boldsymbol{\gamma}_{w_{ij}^t} = \mathbf{0} \quad i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\sum_{\tau\in\mathcal{T}} (\bar{\mathbf{a}}^{\tau\prime}\boldsymbol{\alpha}_{x_{ij}^t}^\tau + \mathbf{d}^{\tau,0\prime}\boldsymbol{\beta}_{x_{ij}^t}^\tau) + \boldsymbol{\gamma}'_{x_{ij}^t}\mathbf{q} \le -x_{ij}^{t,0},$$
$$i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\sum_{\tau\in\mathcal{T}} (\bar{\mathbf{D}}^{\tau\prime}\boldsymbol{\alpha}_{x_{ij}^t}^\tau - \mathbf{D}^{\tau\prime}\boldsymbol{\beta}_{x_{ij}^t}^\tau) + \mathbf{A}'\boldsymbol{\gamma}_{x_{ij}^t} = \mathbf{x}_{ij}^{t,K_T},$$
$$i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\mathbf{B}'\boldsymbol{\gamma}_{x_{ij}^t} = \mathbf{0}, \quad i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$v_{ij}^{t,k} = x_{ij}^{t,k} = h_j^{t,k} = 0, \quad i\in\mathcal{M}, j\in\mathcal{N}, k\in\mathcal{K}_T\backslash\mathcal{K}_t, t\in\mathcal{T};$$

$$w_{ij}^{t,k} = 0, \quad i\in\mathcal{M}, j\in\mathcal{N}, k\in\mathcal{K}_T\backslash\mathcal{K}_{t-1}, t\in\mathcal{T}$$

$$\boldsymbol{\gamma}_{h_j^t}, \boldsymbol{\gamma}_{v_{ij}^t}, \boldsymbol{\gamma}_{w_{ij}^t}, \boldsymbol{\gamma}_{x_{ij}^t} \ge \mathbf{0}, \quad i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\boldsymbol{\alpha}_{h_j^t}, \boldsymbol{\alpha}_{v_{ij}^t}, \boldsymbol{\alpha}_{w_{ij}^t}, \boldsymbol{\alpha}_{x_{ij}^t} \ge \mathbf{0}, \quad i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

$$\boldsymbol{\beta}_{h_j^t}, \boldsymbol{\beta}_{v_{ij}^t}, \boldsymbol{\beta}_{w_{ij}^t}, \boldsymbol{\beta}_{x_{ij}^t} \ge \mathbf{0}, \quad i\in\mathcal{M}, j\in\mathcal{N}, t\in\mathcal{T};$$

where

$$\mathbf{h}_j^{t,K_T} = \begin{pmatrix} h_j^{t,1} \\ \vdots \\ h_j^{t,K_T} \end{pmatrix}, \quad \mathbf{v}_{ij}^{t,K_T} = \begin{pmatrix} v_{ij}^{t,1} \\ \vdots \\ v_{ij}^{t,K_T} \end{pmatrix},$$

$$\mathbf{w}_{ij}^{t,K_T} = \begin{pmatrix} w_{ij}^{t,1} \\ \vdots \\ w_{ij}^{t,K_T} \end{pmatrix}, \quad \text{and} \quad \mathbf{x}_{ij}^{t,K_T} = \begin{pmatrix} x_{ij}^{t,1} \\ \vdots \\ x_{ij}^{t,K_T} \end{pmatrix}.$$

Appendix E. Warehouse Layouts

Figure E.1 **(a) Layout of the Case Study in §4: Storage Locations Are Grouped Into Different Classes by Partitioning the Warehouse with a Grid; (b) Layout 1 (U-Flow Layout): Receiving and Shipping Docks Are Located at the Same Point; (c) Layout 2 (Flow-Through Layout): Receiving and Shipping Docks Are Located at the Centers of the Opposite Sides of the Warehouse; (d) Layout 3: Receiving and Shipping Docks Are Located at Arbitrary Points on the Opposite Sides of the Warehouse**
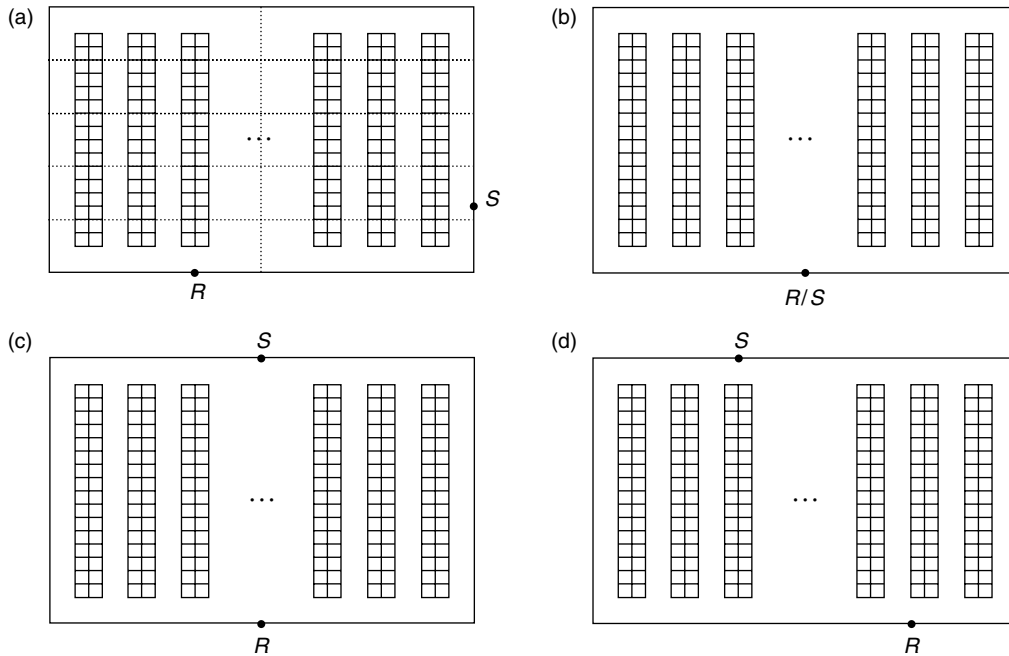
References

Adida E, Perakis G (2006) A robust optimization approach to dynamic pricing and inventory control with no backorders. *Math. Programming* 107(1–2):97–129.

Atamtürk A, Zhang M (2007) Two-stage robust network flow and design under demand uncertainty. *Oper. Res.* 55(4):662–673.

Bartholdi JJ III, Hackman ST (2007) *Warehouse and Distribution Science* (Supply Chain & Logistics Institute, Atlanta). Accessed July 4, 2012, http://www.warehouse-science.com/.

Ben-Tal A, Nemirovski A (1998) Robust convex optimization. *Math. Oper. Res.* 23(4):769–805.

Ben-Tal A, Nemirovski A (1999) Robust solutions to uncertain programs. *Oper. Res. Let.* 25(1):1–13.

Ben-Tal A, Nemirovski A (2000) Robust solutions of linear programming problems contaminated with uncertain data. *Math. Programming* 88(3):411–424.

Ben-Tal A, Golany B, Nemirovski A, Vial J (2005) Supplier-retailer flexible commitments contracts: A robust optimization approach. *Manufacturing Service Oper. Management* 7(3):248–273.

Ben-Tal A, Goryashko A, Guslitzer E, Nemirovski A (2004) Adjustable robust solutions of uncertain linear programs. *Math. Programming* 99(2):351–376.

Bertsimas D, Sim M (2003) Robust discrete optimization and network flows. *Math. Programming* 98(1–3):49–71.

Bertsimas D, Sim M (2004) The price of robustness. *Oper. Res.* 52(1):35–53.

Bertsimas D, Thiele A (2006) A robust optimization approach to inventory theory. *Oper. Res.* 54(1):150–168.

Bertsimas D, Iancu DA, Parrilo PA (2010) Optimality of affine policies in multistage robust optimization. *Math. Oper. Res.* 35(2):363–394.

Bertsimas D, Pachamanova D, Sim M (2003) Robust linear optimization under general norms. *Oper. Res. Lett.* 32(6):510–516.

Chen W, Sim M (2009) Goal driven optimization. *Oper. Res.* 57(2):342–357.

Chen X, Sim M, Sun P, Zhang J (2008) A linear decision-based approximation approach to stochastic programming. *Oper. Res.* 56(2):344–357.

de Koster R, Le-Duc T, Roodbergen KJ (2007) Design and control of warehouse order-picking: A literature review. *Eur. J. Oper. Res.* 182(2):481–501.

El-Ghaoui L, Lebret H (1997) Robust solutions to least-square problems to uncertain data matrices. *SIAM J. Matrix Anal. Appl.* 18(4):1035–1064.

El-Ghaoui L, Oustry F, Lebret H (1998) Robust solutions to uncertain semidefinite programs. *SIAM J. Optim.* 9(1):33–52.

Erdoğan E, Iyengar G (2006) Ambiguous chance constrained problems and robust optimization. *Math. Programming* 107(1–2): 37–61.

Erera AL, Morales JC, Savelsbergh M (2009) Robust optimization for empty repositioning problems. *Oper. Res.* 57(2):468–483.

Eynan A, Rosenblatt MJ (1994) Establishing zones in single-command class-based rectangular AS/RS. *IIE Trans.* 26(1): 38–46.

Gilboa I, Schmeidler D (1989) Maximin expected utility theory with non-unique prior. *J. Math. Econom.* 18(2):141–153.

Goetschalckx M, Ratliff HD (1990) Shared storage policies based on the duration stay of unit loads. *Management Sci.* 36(9):1120–1132.

Goh J, Sim M (2010) Distributionally robust optimization and its tractable approximations. *Oper. Res.* 58(4):902–917.

Goh J, Sim M (2011) Robust optimization made easy with ROME. *Oper. Res.* 59(4):973–985.

Graves SC, Hausman WH, Schwarz LB (1977) Storage-retrieval interleaving in automatic warehousing systems. *Management Sci.* 23(9):935–945.

Gu J, Goetschalckx M, McGinnis LF (2007) Research on warehouse operation: A comprehensive review. *Eur. J. Oper. Res.* 177(1):1–21.

Hausman WH, Schwarz LB, Graves SC (1976) Optimal storage assignment in automatic warehousing systems. *Management Sci.* 22(6):629–638.

Heskett JL (1963) Cube-per-order index: A key to warehouse stock location. *Transportation Distribution Management* 3:27–31.

Heskett JL (1964) Putting the cube-per-order index to work in warehouse layout. *Transportation Distribution Management* 4:23–30.

Kulturel S, Ozdemirel NE, Sepil C, Bozkurt Z (1999) Experimental investigation of shared storage assignment policies in automated storage/retrieval systems. *IIE Trans.* 31(8):739–749.

Mallette AJ, Francis RL (1972) A generalized assignment approach to optimal facility layout. *AIIE Trans.* 4(2):144–147.

Malmborg CJ, Bhaskaran K (1990) A revised proof of optimality for the cube-per-order index rule for stored item location. *Appl. Math. Model.* 14(2):87–95.

Roodbergen KJ, Vis IFA (2009) A survey of literature on automated storage and retrieval systems. *Eur. J. Oper. Res.* 194(2):343–362.

Rosenblatt MJ, Eynan A (1989) Deriving the optimal boundaries for class-based automatic storage/retrieval systems. *Management Sci.* 35(12):1519–1524.

Schwarz LB, Graves SC, Hausman WH (1978) Scheduling policies for automatic warehousing systems: Simulation results. *AIIE Trans.* 10(3):260–270.

See C, Sim M (2010) Robust approximation to multi-period inventory management. *Oper. Res.* 58(3):583–594.

Soyster AL (1973) Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper. Res.* 21(5):1154–1157.

Thonemann UW, Brandeau ML (1998) Note. Optimal storage assignment policies for automated storage and retrieval systems with stochastic demands. *Management Sci.* 44(1):142–148.

Van den Berg JP (1999) A literature survey on planning and control of warehousing systems. *IIE Trans.* 31(8):751–762.