CLASS-BASED STORAGE WITH A FINITE NUMBER OF ITEMS

Yugang Yu
Rotterdam School of Management, Erasmus University, the Netherlands

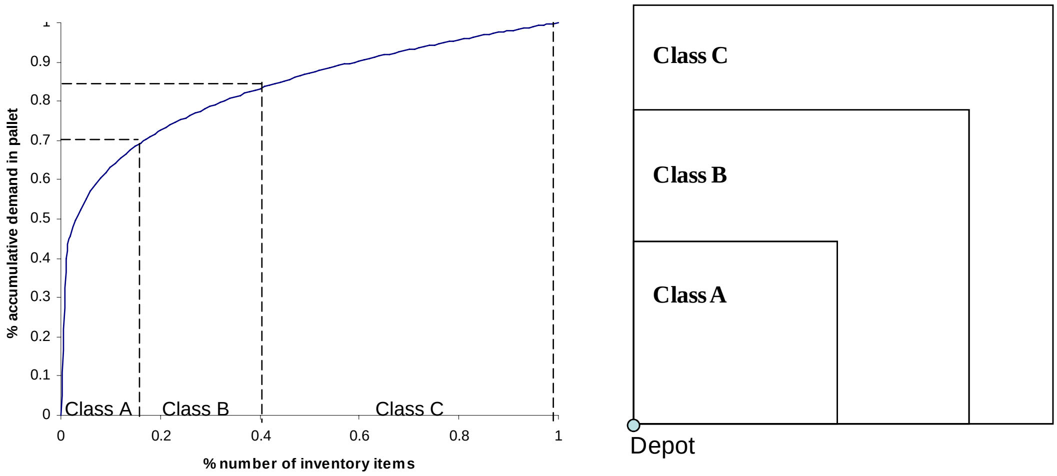
René B.M. de Koster
Rotterdam School of Management, Erasmus University, the Netherlands

ABC class-based storage is widely studied in literature and applied practice. It divides all stored items into a limited number of classes according to their demand rates (turnover per unit time). Classes of items with higher turnovers are stored in a region closer to the warehouse depot. In literature, it is commonly shown that the use of more storage classes leads to shorter travel time for storing and retrieving items. A basic assumption in this literature commonly is that the required storage space of items equals their average inventory levels, which is right if an infinite number of items are stored in each storage region. However, if a finite number of items are stored in the warehouse, more storage classes need more space to store the items: more classes lead to fewer items stored per class, which have less opportunity to share space with other items. This paper revisits ABC class-based storage by relaxing the common assumption that the total required storage space of all items is independent of the number of classes. We develop a travel time model and use it for optimizing the number and the boundaries of classes. Our numerical results illustrate that a small number of classes is optimal.

1. Introduction

ABC class-based storage is the most commonly used storage policy in practice and is widely discussed in many operations management textbooks (Adams et al., 1996; Heragu, 2006; Tompkins et al., 2003), and scientific papers (De Koster et al., 2007; Graves et al., 1977; Gu et al., 2007; Hausman et al., 1976; Rosenblatt and Eynan, 1989; Thonemann and Brandeau, 1998).

ABC class-based storage divides items into different classes (three is common in practice) according to the ABC demand curve (see Figure 1(a)). A relatively small number of highly demanded items are grouped as A-class items and are then stored in a warehouse region closest to the depot. Rarely demanded items grouped as C-class items are stored in the region farthest from the depot. Within each class, items are stored randomly. Figure 1(b) shows a side view of a storage rack with an example of ABC class-based storage in an automated storage and retrieval warehousing system. In such a system, the distance between a storage location and the depot is measured in Chebyshev distance (Hausman et al., 1976).



(a) An ABC Curve

(b) ABC class-based storage regions

Figure 1. ABC class-based storage. Figure (b) gives a side view of the storage rack

The seminal paper of Hausman et al. (1976) formulates a travel time model for class-based storage and addresses the benefits of two or three class-based storage under different ABC demand curves. Rosenblatt and Eynan (1989) generalize the model for *n*-class-based storage and develop an algorithm for finding optimal boundaries of all classes. Following these two papers, most research on class-based storage (Eynan and Rosenblatt, 1994; Gu et al., 2007; Kouvelis and Papanicolaou, 1995; Larson et al., 1997; Yu and De Koster, 2009) assumes that the total required storage space does not depend on the number of classes. The assumption is true if the number of items in each class is sufficiently large (infinite). A basic assumption in these models is that multiple items share a common storage space, and they are replenished to the system at different points in time. When an item is replenished, empty storage space for receiving it can be found by using part of the common space. As a result, if the number of items in a class is infinite, the required storage space of the class approximately equals the total average inventory level of all items in the class (Hausman et al., 1976, page 634). With this assumption, the numerical results in these papers show that an increase in number of classes can reduce average travel time for storing or retrieving items (see the curve indicating conventional research results in Figure 2).

However, if a finite number of items are stored in a warehouse, the total required storage space differs from that given by Hausman et al. (1976).

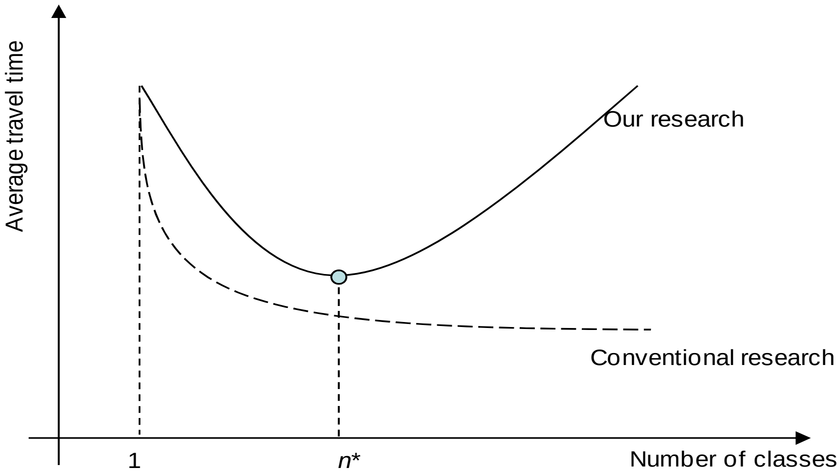


Figure 2. The travel times in two different lines of research

This paper extends Hausman et al. (1976) and Rosenblatt and Eynan (1989) by assuming a finite number of items are stored. The total required storage space therefore becomes a function of the number of classes and the number of items in each class. More classes with fewer items per class result in a larger required storage space per item. With an increase in the number of classes, the number of items per class decreases and more storage space is needed to store all items, which increases the average travel time for storing/retrieving items. However, simultaneously, the item turnovers can be classified more precisely and frequently requested items can be stored closer to the depot, which decreases the average travel time for storing/retrieving items.

A travel time model to describe the above tradeoff is missing in Hausman et al. (1976). In the literature, Eldemir et al. (2004) develop formulas to estimate the required storage space as a function of the number of items under random storage. They comprehensively analyze the impact of the skewness of the ABC curve on required storage space and then on the average travel time. However, the total required storage space is not given as a function of the number and boundaries of classes such that they can be optimized. Their model is also too difficult to be used for optimizing the number of classes and class boundaries.

This paper develops a new formula to estimate the required storage space for *n*-class-based storage. A travel time model and a corresponding algorithm are then developed to determine the optimal number of classes and their boundaries. Our results support that, if the number of classes is large, an increase in the number of classes lengthens the travel time for storing/retrieving items as shown in Figure 2, which may explain why in practice only a small number of classes are used.

This paper is organized as follows. Section 2 describes the problem as defined by Hausman et al. (1976) and Rosenblatt and Eynan (1989). Section 3 introduces the conventional travel time model from Hausman et al. (1976) and Rosenblatt and Eynan (1989). In Section 4, we extend the conventional research by assuming a finite number of items stored in the system. In Section 5, numerical examples are given to illustrate the main findings of this paper. Section 6 concludes the paper.

2. Problem Description

This section first describes the studied system. We then define mathematical notations and formulas used throughout the paper.

Following Hausman et al. (1976), we consider a basic automated warehousing system: an automated storage/retrieval system (AS/RS) consisting of a storage/retrieval (S/R) machine, a storage rack, and one depot where all items enter and leave the system. Items can be finished goods, work-in-process, or raw materials, and they are stored on standardized pallets before arriving at the AS/RS system. The system works as follows: when a storage pallet arrives at the depot of the system, the machine retrieves it and transports it (moving in horizontal and vertical directions simultaneously) to any given storage location in the rack. Upon request of a stored pallet, the machine picks up and moves the pallet to the depot.

In accordance with Hausman et al. (1976), the system has the following properties:

1) All storage locations are the same size as the pallets themselves. On each pallet, only one item is stacked.
2) The depot is at the lower-left side of the storage rack, as shown in Figure 1(b).
3) The storage rack is “square” in time; the time for the machine to move from the depot to the most distant column equals the time from the depot to the highest row.
4) The capacity of the machine is one unit load and it operates in a single-command mode; the machine either stores or retrieves a pallet each time.
5) The pickup/deposit time for the machine to load/unload a pallet is ignored.
6) The turnover of each item is known and constant through time. The turnover of an item equals the number of times the item is stored or retrieved in a unit-time period, such as a week, a month, or a year.
7) Item inventories are replenished according to the simple EOQ model, where the demand of every item (measured in number of pallets) is deterministic and determined by the ABC curve given in Eq. (1).
8) The items are ranked according to their marginal contribution to the total demand using the ABC curve. An item that has a smaller contribution is indexed with a larger number.

The ABC curve is a plot of ranked cumulative percentage demand per unit time, $G(i)$, and modeled by:

$$G(i) = i^s = \int_0^i D(j)dj \,/ \int_0^1 D(j)dj\,, \quad \text{for } 0 < s \le 1, \tag{1}$$

where i is the item at the i^{th} percentile in the ranked sequence of all items. $D(i)$ is the demand of item i per unit time. The smaller s is, the skewer the ABC curve is.

Without loss of generality, assuming that the total demand $\int_0^1 D(j)dj = 1$, Hausman et al. (1976) then have:

$$D(i) = dG(i)/di = si^{s-1}, \;\; 1 < i \le 1. \tag{2}$$

Given the above system properties and the item demands determined by Eq. (2), we want to find the average one-way travel time for storing/retrieval a pallet. A one-way travel time is the travel-time distance between the depot and a pallet storage position.

Class-based storage divides the storage space into *n* regions. Region *k* is dedicated to store items of class *k*, *k*=1,…,*n*. A region with items of higher turnover is located closer to the depot. As shown in Figure 1, the regions are L-shaped and square-in-time.

Storage locations and corresponding pallets are defined as follows.

j index of the j^{th} storage location (or pallet). A pallet closer to the depot has a smaller index.

The notations related to a region (class) *k* include:

i_k the item with the lowest turnover in class *k*.

j_k the storage location (a corresponding pallet) farthest from the depot in region (class) *k*. It corresponds to the total required storage space of items 1 to i_k .

t_k average one-way travel time for storing/retrieving a pallet of class *k*.

G_k 100×(cumulative demand for the first *k* classes)/(the total annual demand of all items)% .

R_k one-way travel time for storing/retrieving a pallet at the boundary of region(class) *k*.

$\Lambda(k)$ the total turnover (in the number of pallets) of all items stored in class *k* per unit time.

T_n average one-way travel time of a pallet for an *n*-class storage system.

With the above notations and according to Hausman et al.(1976) and Rosenblatt and Eynan (1989), the average one-way travel time in an *n*-class system, T_n can be formulated as:

$$T_n = \frac{\sum_{k=1}^{n} t_k \Lambda(k)}{\sum_{k=1}^{n} \Lambda(k)} = \sum_{k=1}^{n} t_k \left(\frac{\Lambda(k)}{\sum_{k=1}^{n} \Lambda(k)}\right) . \qquad (3)$$

$\Lambda(k)/(\sum_{k=1}^{n} \Lambda(k))$ is the weight of item turnover of class *k* in the total turnover of all items where $\Lambda(k)$ is the turnover of all the items in class *k*, and $\sum_{k=1}^{n} \Lambda(k)$ is the total turnover of items in the whole system. $\Lambda(k)/(\sum_{k=1}^{n} \Lambda(k))$ is calculated by using $\Lambda(k) = \int_{R_{k-1}^2}^{R_k^2} \lambda(j)dj$, *k*=1,…,*n*, where $\lambda(j)$ is the turnover of the *jth* pallet in the system in Hausman et al.(1976). Rosenblatt and Eynan (1989) simplify the calculation using Eq. (1) to obtain:

$$\Lambda(k) / \sum_{k=1}^{n} \Lambda(k) = G_k - G_{k-1} = i_k^s - i_{k-1}^s , \quad k = 1,...,n . \qquad (4)$$

In case of an SIT system, t_k equals (Eynan and Rosenblatt, 1994; Hausman et al., 1976; Rosenblatt and Eynan, 1989):

$$t_k = \frac{2(R_k^3 - R_{k-1}^3)}{3(R_k^2 - R_{k-1}^2)} \qquad k = 1,...,n, \tag{5}$$

Substituting Eq. (4) and (5) into (3), Eq. (3) can be rewritten as:

$$T_n = \sum_{k=1}^{n} \frac{2(R_k^3 - R_{k-1}^3)}{3(R_k^2 - R_{k-1}^2)} (i_k^s - i_{k-1}^s). \tag{6}$$

To minimize T_n in Eq.(6), we have to derive the relationship between R_k and i_k, k=1, …, n. If this relationship is obtained, the travel time in Eq. (6) can then be minimized by optimizing either R_k, k=1, …, n or i_k, k=1, …, n-1.

3. Conventional Travel Time Model for Class-based Storage

Using the results of Hausman et al.(1976) and Rosenblatt and Eynan (1989), this section summarizes the conventional relationship between R_k and i_k, k=1, …, n based on the derivation of the required storage space of each class. This leads to a travel time model, which can be used to optimize class boundaries.

3.1 Required Storage Space

In the classic economic order quantity (EOQ) model, the EOQ (in pallets) of item i is

$$Q(i) = (2KD(i))^{1/2} \tag{7}$$

where K is the ratio of order cost to holding cost and assumed to be the same for all items. The average inventory (in pallets) of item i is $Q(i)/2$. The average inventory level of all items, L, therefore is

$$L = \int_{i=0}^{1} Q(i)/2di = \int_{i=0}^{1} (2KD(i))^{1/2}/2)di = (2Ks)^{1/2}/(s+1). \tag{8}$$

Hausman et al. (1976, p. 634) assume the total required storage space (measured in number of pallet locations) for storing all items, A, equals L:

$$A = L = (2Ks)^{1/2}/(s+1). \tag{9}$$

Therefore, if full turnover based storage is considered, the relationship between pallet j, $j \in (0,\ A]$ and item i can be determined by (Hausman et al., 1976):

$$j = \int_{k=0}^{i(j)} (2KD(k))^{1/2}/2)dk, \tag{10}$$

that is, $$i(j) = [(s+1)^2 j^2/(2Ks)]^{1/(s+1)}. \tag{11}$$

Without loss of generality, j is rescaled from (0, L] to (0, 1] by replacing j with $j{*}L$ in Eq. (11), and we have

$$i(j) = [\frac{(s+1)^2[(2Ks)^{1/2}/(s+1)j]^2}{2Ks}]^{1/(s+1)} = j^{2/(s+1)}, \tag{12}$$

Correspondingly, j_k and i_k relate as follows:

$$i_k = j_k^{2/(s+1)}, \quad k=1,\ldots,n. \tag{13}$$

Note that the total required storage space, max {j}, is 1 when all items (i=100%) are stored due to the rescaling of j to (0, 1].

3.2 Relationship between R and i

Because the rack is SIT, we have $j_k = R_k^2$ (or $(R_k / R_n)^2$ as $R_n = \sqrt{\max\{j\}} = 1$ after rescaling) according to Rosenblatt and Eynan (1989). We then obtain

$$i_k = j_k^{2/(s+1)} = R_k^{4/(s+1)}, \quad k=1,\ldots,n, \tag{14}$$

where $i_n = j_n = R_n = 1$.

3.3 Travel Time Model and Its Solution

According to Eq. (4) and (14), we have

$$G_k - G_{k-1} = i_k^s - i_{k-1}^s = R_k^{4s/(s+1)} - R_{k-1}^{4s/(s+1)}. \tag{15}$$

Substituting Eq. (15) into (6), T_n becomes a function of R_k, k=1, …, n-1. The corresponding conventional model (Rosenblatt and Eynan, 1989) becomes:

Model CM(n):

$$\text{Min } T_n = \sum\nolimits_{k=1}^{n} \frac{2(R_k^3 - R_{k-1}^3)}{3(R_k^2 - R_{k-1}^2)} [R_k^{4s/(s+1)} - R_{k-1}^{4s/(s+1)}]. \tag{16}$$

Subject to: $0 < R_k < 1$ and $R_{k-1} < R_k$.

Decision variables: R_k, k=1, …, n-1. $R_0 = 0$ and R_n =1 are known.

To solve Model CM(n), an iterative Eq. (17) is derived:

$$T_n = \frac{2(1 - R_{k-1}^3)}{3(1 - R_{k-1}^2)} [1 - R_{k-1}^{4s/(s+1)}] + (R_{k-1}^{(5s+1)/(s+1)}) T_{n-1}. \tag{17}$$

Given $T_0 = 0$, both T_k, k=1,…,n, and R_k, k=1,…,n can be optimally found (Rosenblatt and Eynan, 1989). We omit the details here.

4. Our Travel Time Model for Class-based Storage

This section relaxes the assumption of Hausman et al. (1976, p. 634) represented in Eq. (9). Eq. (9) holds if the number of items in each class is infinite. Our relaxation allows a finite number of items to share storage space for each class. The required storage space of items that is then not simply equal to the average inventory level, L, but a function of the number of items and the number of classes. In subsection 4.1, we derive the required storage space of storage classes and in subsection 4.2 we find the relationship between R_k, and i_k, k=1, …, n. The travel time corresponding with the required storage space is given in subsection 4.3.

4.1 New required storage space function

If the number of items for sharing a storage space is finite, the average required storage space of an item may depend on many factors, such as the number of items sharing the space, the skewness of the ABC curve (s), inventory replenishment policies, and inventory cost (*K*). We first determine the required storage space of an item as a function of the number of items in the same shared space by fixing all the other factors.

The function $a_i(N_k)$ denotes the required space to store item *i* in class *k*, where N_k represents the total number of items sharing the common storage space of class *k*. We then have.

$$N_k = N(i_k - i_{k-1}) . \qquad (18)$$

We have determined $a_i(N_k)$ by simulation. It appears to be fairly independent of the shape of the ABC-curve *s*. The resulting function is depicted in Figure 3.

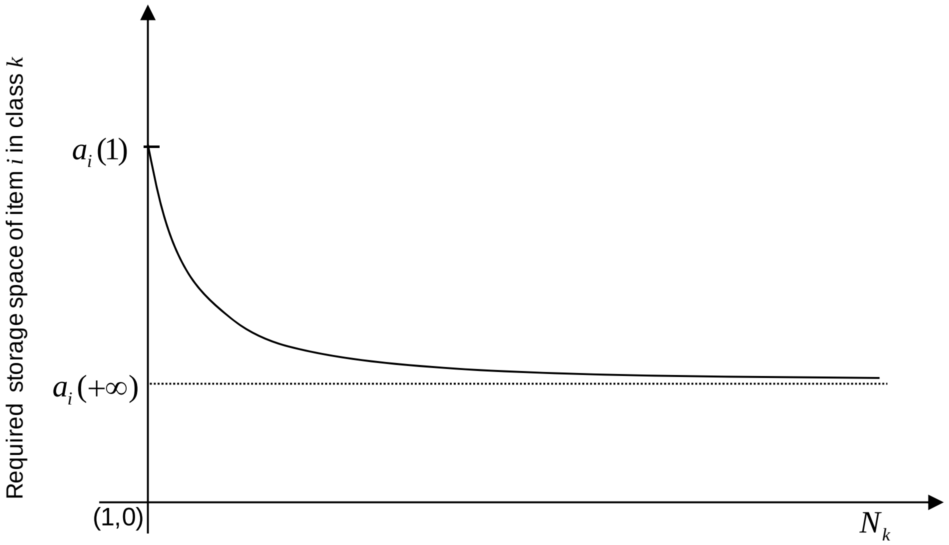


Figure 3. The required storage space of item *i* as a function of N_k

At $N_k = +\infty$, the lower bound of the required storage space of item *i*, $a_i(+\infty)$, is obtained as:

$$a_i(+\infty) = Q(i)/2 , \qquad (19)$$

which is consistent with Hausman et al. (1976).

At $N_k = 1$, dedicated storage is implemented. The upper bound of the required storage space of item *i*, can then be determined by the EOQ of item *i*:

$$a_i(1) = Q(i) \qquad (20)$$

For $1 < N_k < +\infty$, the required storage space of item *i* is between $a_i(+\infty)$ and $a_i(1)$. With an increase in N_k, the required storage space of item *i* decreases and the decrease rate is also decreasing to approach zero.

To formulate the curve in Figure 3, we can approximately represent the required storage space for item *i* as a function of the number of items as:

$$a_i(N_k) = a_i(+\infty) + (a_i(1) - a_i(+\infty))N_k^{-\varepsilon} = a_i(+\infty)(1 + N_k^{-\varepsilon}) , \qquad (21)$$

where ε is a fixed parameter and ε>0. This is a type of Cobb-Douglas function, with an elasticity ε. A 1% increase in N_k leads to an ε% decrease in $(a_i(N_k) - a_i(+\infty))$. The larger ε is, the larger the impact of a change of N_k is on the required space of item *i*. According to our simulation results, ε does not heavily depend on *s* and is around 0.3.

Substituting Eq.(7) and (19) into Eq. (21), we have

$$a_i(N_k) = (0.5KD(i))^{1/2}(1 + N_k^{-\varepsilon}) . \qquad (22)$$

Eq. (22) shows that $a_i(N_k)$ is not only determined by N_k, but also by other various factors such as the skewness s of the ABC curve, the replenishment policy and the costs of items.

Using Eq. (22), we can obtain the required storage space (in number of pallets) for class *k*:

$$j_k - j_{k-1} = \int_{i_{k-1}}^{i_k} a_i(N_k)di = \int_{i_{k-1}}^{i_k} (0.5KD(i))^{1/2}(1 + N_k^{-\varepsilon})di \ , \qquad k = 1,...,n \, , \qquad (23)$$

where j_0 =0.

To make our result comparable to that of (Hausman et al., 1976), we rescale *j* to *j*/ *L* by replacing *j* with *j***L*, and then obtain

$$(j_k - j_{k-1})L = \int_{i_{k-1}}^{i_k} (0.5KD(i))^{1/2}(1 + N_k^{-\varepsilon})di \ , \qquad k = 1,...,n \, . \qquad (24)$$

By substituting Eq. (8), Eq. (24) can then be reformulated as:

$$j_k = \frac{(s+1)(0.5K)^{1/2}}{(2Ks)^{1/2}} \sum_{l=1}^{k} (1 + N_l^{-\varepsilon}) \int_{i_{l-1}}^{i_l} D(i)^{1/2} di \ , \qquad k = 1,...,n \, . \qquad (25)$$

Substituting Eq. (2) into Eq. (25), the total required storage of classes 1,…, *k*-1 equals:

$$j_k = i_k^{(s+1)/2} + \sum_{l=1}^{k} N_l^{-\varepsilon} (i_l^{(s+1)/2} - i_{l-1}^{(s+1)/2}) \ , \qquad k = 1,...,n \, . \qquad (26)$$

Eq. (26) shows the relation between j_k and $i_1, \ldots, i_k$.

4.2 Relationship between R and i

R_k is determined by the total required storage space of all items in the previous *k* classes. As before,

$$R_k = \sqrt{j_k} \ , \qquad k = 1,...,n \, , \qquad (27)$$

and $j_0 = R_0 = 0$.

Substituting Eq. (26) and **Error! Reference source not found.**(18) into Eq. (27), the relationship between R_k and $i_1,...,i_k$ can be expressed as:

$$R_k = \sqrt{i_k^{(s+1)/2} + \sum_{l=1}^{k} (N(i_l - i_{l-1}))^{-\varepsilon} (i_l^{(s+1)/2} - i_{l-1}^{(s+1)/2})} \ , \qquad k = 1,...,n \, . \qquad (28)$$

4.3 Travel time Model

Considering the new relationship between R_k and i_k, we obtain the following model.

Model NM(*n*):

$$\text{Min } T_n = \sum_{k=1}^{n} \frac{2(R_k^3 - R_{k-1}^3)}{3(R_k^2 - R_{k-1}^2)} (i_k^s - i_{k-1}^s) \tag{29}$$

Subject to: Eq. (28) which gives the relationship between and R_k and i_k and

$$N(i_k - i_{k-1}) \geq 1 \quad k = 1,...,n. \tag{30}$$

Decision variables: either $R_k > 0$, $k = 1,...,n$ or i_k >0, $k = 1,...,n-1$.

Model NM differs from Model CM in three respects. Eq. (28) shows that the class boundary, R_k, is not only related to i_k and *s*, but also to the number of items in class *k* and all its previous classes 1,…,*k*-1. Next, because *N* is finite, Constraints (30) are required to ensure that at least one item is stored in each class. Finally, R_n is an unknown value in Model NM because R_n relates to R_k, $k = 1,...,n-1$ are given, but R_n =1 in Model CM.

Model NM is quite complex to solve. The methodology for solving Model CM used by Rosenblatt and Eynan (1989) cannot be applied here. An iterative relation as in Eq. (17) does not hold for Model NM. Next, the objective function (29) is nonlinear and we do not know whether it is a convex function of $R_k > 0$ (or j_k or i_k) $k = 1,...,n$ or not. Finally, Constraints (30) are nonlinear functions of R_k, $k = 1,...,n$ considering Eq. (28). If i_k >0, $k = 1,...,n-1$ are taken as the decision variables, i_n is known to be 1. Moreover, to avoid reformulating i_k as a function of $R_1,...,R_k$ using Eq. (28), without loss of generality, we hereafter select i_k >0, $k = 1,...,n-1$ as the decision variables.

5. Solution Methodology

To solve Model NM, we introduce an algorithm including two main steps. We first find all the local optimal solutions. Next, these local optimal solutions are then compared to obtain a global optimal solution that minimizes the travel time T_n.

To obtain a local optimal solution, we introduce Lagrange multiplier vectors $\lambda = (\lambda_1,...,\lambda_n) \geq 0$ and $u = (u_1,...,u_{n-1}) \geq 0$ for constraints (30) and i_k >0, $k = 1,...,n-1$, respectively. The Lagrangian function of Model NM can be defined by

$$L = T_n - \sum_{k=1}^{n} \lambda_k [N(i_k - i_{k-1}) - 1] + \sum_{k=1}^{n-1} u_k i_k, \tag{31}$$

According to Bertsekas (2003), the local minimal solutions of Model NM must satisfy the Karush-Kuhn-Tucker necessary conditions that can be described as follows.

$$\partial L / \partial i_k = 0, \qquad k = 1,...,n-1, \tag{32}$$

$$\lambda_k [N(i_k - i_{k-1}) - 1] = 0, \quad k = 1,...,n, \tag{33}$$

$$u_k i_k = 0 \quad k = 1,...,n-1. \tag{34}$$

To find a local minimum solution of Model NM, we have to solve simultaneous Eq. (32)-(34) that are nonlinear. Here, we use the Newton-Raphson method (Press et al.,

2007) to solve the equations, which converges in $O(n^2)$ where *n* is the number (classes) of variables. Therefore, a local minimum can be obtained.

The difficulty is now to find all local optimal solutions. To do so, a grid search (Hausman et al., 1976) is applied, which divides the feasible area of i_k >0, $k = 1,...,n-1$ into many cells and a starting point in each cell is selected to find a local optimal solution. We subsequently make the grid more dense and repeatedly compute the local optima in each cell. We stop when a more dense grid does not produce a better solution in a certain cell. The output at that stage is the "optimal" solution in the cell.

From all local optima we can approximately find the optimal solutions of Model NM. Moreover, by varying *n*, we can find the optimal solutions to Model NM for a different number of classes to obtain the optimal number of classes.

6. Experiments

In our numerical examples, the total number of items *N*=100 and ε =0.3 (approximate value, obtained by simulation). The results for the optimal one-way travel time T_n as a function of the number classes, *n*, are shown in Fig. 4-7 under different ABC curves with *s*=1 (20%/20%), 0.431(20%/50%), 0.222(20%/70%), and 0.065(20%/90%). An ABC curve with *x*%/*y*% indicates that *x*% of all items in the warehouse are responsible for explain *y*% of the total turnover in pallets and the corresponding *s* is determined by solving Eq. (1). The required storage space as a function of *n* for two extreme cases of *s*=1 (20%/20%) and 0.065(20%/90%) is shown in Fig. 8. The computation time of our algorithm much depends on the number of classes, which takes a second to solve a problem of 1- 3 classes, but may take more than one hour if $n \geq 50$.

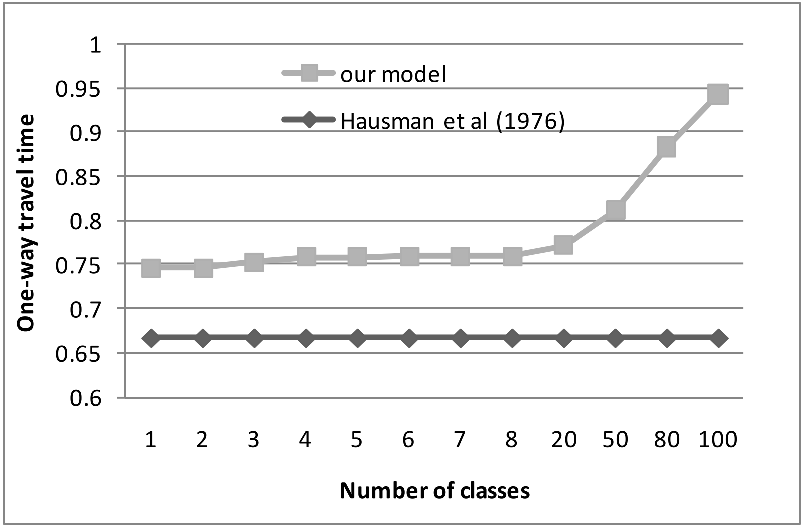


Figure 4. Travel time with the 20%/20% ABC curve

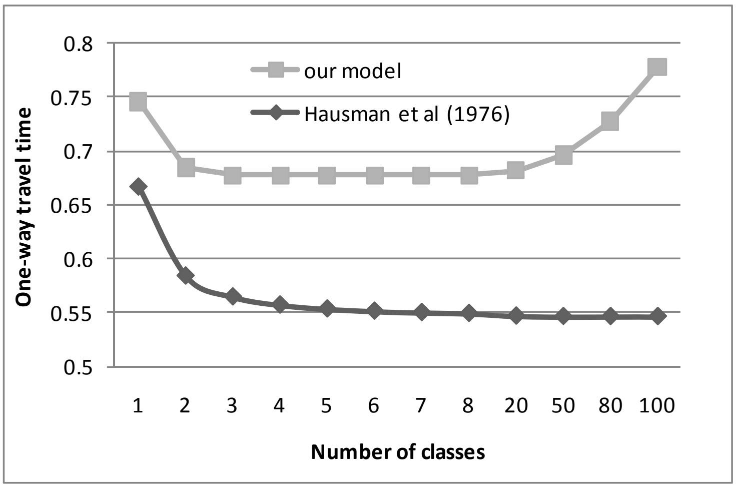


Figure 5. Travel time with the 20%/50% ABC curve

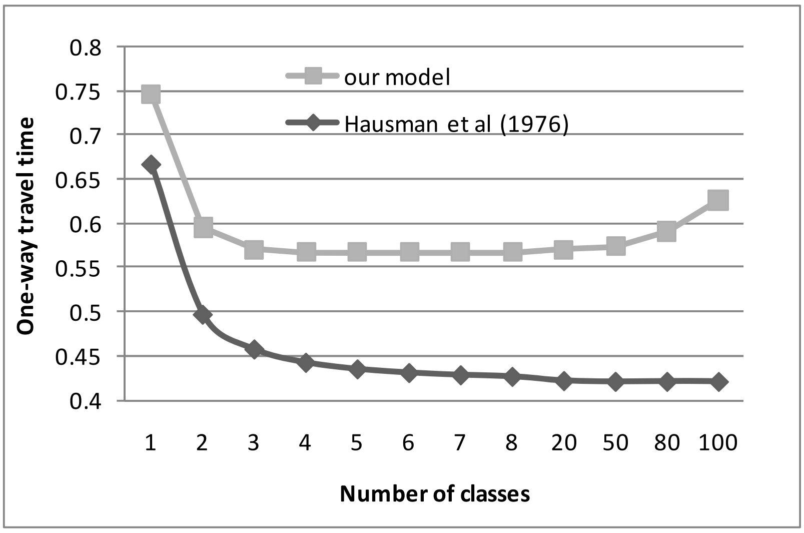


Figure 6. Travel time with the 20%/70% ABC curve

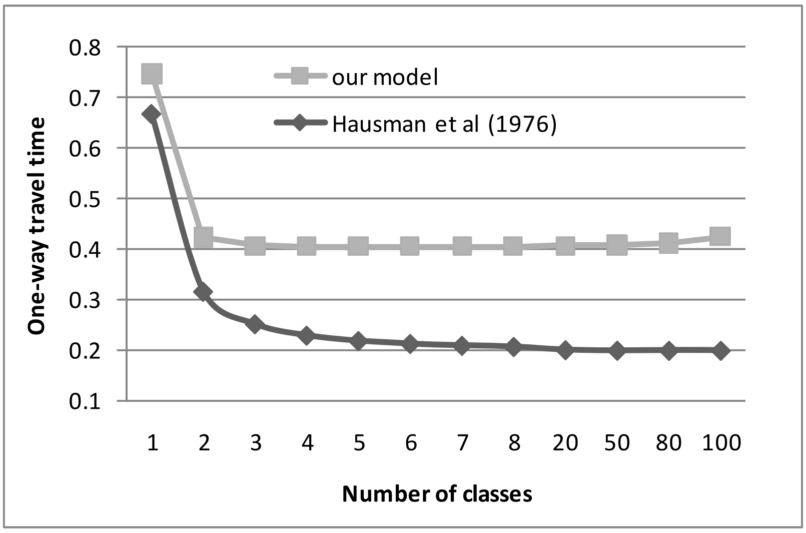
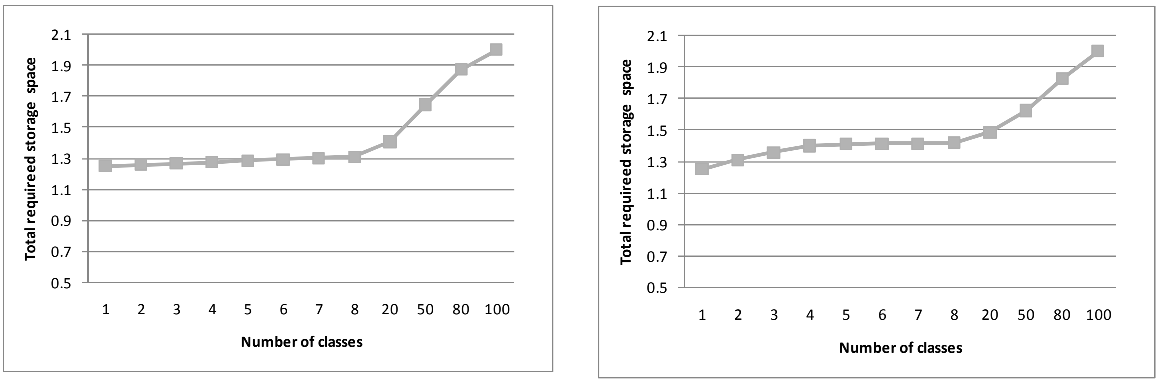


Figure 7. Travel time with the 20%/90% ABC curve



(a) *s*=1 (the 20%/20% ABC curve) **(b) *s*=0.065 (the 20%/90% ABC curve)**

Figure 8. The total required storage space changes by varying *n*

The results in Figs 4-8 lead to the following findings.

- With an increase in the number of classes, *n*, the one-way travel time, T_n increases, if the number of classes is beyond its optimal value. For example, when *n* increases from 5 to 100, T_n increases more than 10% for the cases of *s*=1 (20%/20%), 0.431 (20%/50%), and 0.222 (20%/70%). These results differ from those of Hausman et al., (1976) who find a shorter T_n for a larger *n*. This difference can be explained as follows. Hausman et al. (1976) assume that *N* and N_k are infinite, which implies the required storage space of every item equals its average inventory level. An increase in *n* makes it possible to locate high turnover items closer to the depot without increasing the total required storage space. However, as we assume *N* is finite an increase in n leads to a smaller number of items in individual classes, who need more total required storage space (see Fig. 8) and then lengthens the travel time T_n^*.

- The optimal number of classes, n^*, is small in our results. $n^* \leq 5$ for all our examples. Using only 3 classes can achieve a near-shortest T_n. This is very close to warehousing practice where ABC class-based storage is commonly used.
- The relative gaps between the travel times of our model and those of Hausman et al. (1976) increase with an increase in n in all examples. Even at n=1, the relative gaps are still quite large (>10% for all examples). As the number of items stored in a real warehouse is finite, it is therefore not recommended to use Eq. (6) to estimate T_n.
- The required storage space is an increasing function of n but the increase rates are different for different s values. Fig. 8 shows that the increase rate at a larger s is smaller when n is small. This is because, when s is larger, the items are more evenly divided over storage classes, which gives better space sharing.

7. Conclusion

This paper extends the work of Hausman et al. (1976) by considering a finite number of items to be stored in a warehousing system. Our results show that the optimal number of classes is commonly between 2 and 5. Three classes can give near-shortest travel times for the ABC curve with s-values between 0.431(20%/50%) and 0.065(20%/90%).

Hundreds of papers exist on class-based storage implicitly using Hausman et al.'s assumption of an infinite number of items. For further research, some of these papers can be revisited to address the consequence by assuming a finite number of items to be stored in the system. Moreover, facility costs caused from a different number of classes may be included in the analysis. The more classes a warehouse has, the more required storage space the warehouse needs, which also implies higher facility operational costs for this larger warehouse.

Acknowledgement

The research is supported by VENI grant (#016.075.154) financed by NWO (the Netherlands Organization for Scientific Research).

References

Adams, N.D., Firth, R., Brown, T.W. and Misenheimer, L.P., 1996. *Warehouse & Distribution Automation Handbook*, pp. McGraw-Hill).

Bertsekas, D.P., 2003. *Nonlinear Programming* (SECOND EDITION ed.), pp. 316-318 Athena Scientific,Belmont, Massachusetts).

De Koster, R., Le-Duc, T. and Roodbergen, K.J., 2007. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2), 481-501.

Eldemir, F., Graves, R.J. and Malmborg, C.J., 2004. New cycle time and space estimation models for automated storage and retrieval system conceptualization. *International Journal of Production Research*, 42(22), 4767-4783.

Eynan, A. and Rosenblatt, M.J., 1994. Establishing Zones in Single-Command Class-Based Rectangular AS/RS. *IIE Transactions*, 26(1), 38-46.

Graves, S.C., Hausman, W.H. and Schwarz, L.B., 1977. Storage-retrieval interleaving in automatic warehousing systems. *Management Science*, 23, 935-945.

Gu, J.X., Goetschalckx, M. and McGinnis, L.F., 2007. Research on warehouse operation: A comprehensive review. *European Journal of Operational Research*, 177(1), 1-21.

Hausman, W.H., Schwarz, L.B. and Graves, S.C., 1976. Optimal storage assignment in automatic warehousing systems. *Management Science*, 22(6), 629-638.

Heragu, S.S., 2006. *Facilities Design*, pp. (Lincoln, NE: iUniverse).

Kouvelis, P. and Papanicolaou, V., 1995. Expected Travel-Time and Optimal Boundary Formulas for a 2-Class-Based Automated Storage-Retrieval System. *International Journal of Production Research*, 33(10), 2889-2905.

Larson, T.N., March, H. and Kusiak, A., 1997. A heuristic approach to warehouse layout with class-based storage. *IIE Transactions*, 29(4), 337-348.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P., 2007. *Numerical Recipes: The Art of Scientific Computing* (3 ed.), pp. Cambridge University Press).

Rosenblatt, M.J. and Eynan, A., 1989. Deriving the optimal boundaries for class-based automatic storage/retrieval systems. *Management Science*, 35(12), 1519-1524.

Thonemann, U.W. and Brandeau, M.L., 1998. Optimal storage assignment policies for automated storage and retrieval systems with stochastic demands. *Management Science*, 44(1), 142-148.

Tompkins, J.A., White, J.A., Bozer, Y.A. and Tanchoco, J.M.A., 2003. *Facilities Planning*, pp. (New York: John Wiley and Sons).

Yu, Y. and De Koster, M.B.M., 2009. Optimal zone boundaries for two-class-based compact three-dimensional automated storage and retrieval systems. *IIE Transactions*, 41(3), 194-208.