Transportation Research Part E 146 (2021) 102207

Contents lists available at ScienceDirect

Transportation Research Part E

journal homepage: www.elsevier.com/locate/tre

The impact of integrated cluster-based storage allocation on parts-to-picker warehouse performance

Masoud Mirzaei[a,*], Nima Zaerpour[b], René de Koster[c]

[a] Department of Industrial Engineering and Innovation, Eindhoven University of Technology, Eindhoven, the Netherlands
[b] College of Business Administration, California State University San Marcos, San Marcos, United States
[c] Rotterdam School of Management, Erasmus University, Rotterdam, the Netherlands

A R T I C L E I N F O

Keywords:
Automated warehousing
Storage policy
Cluster-based storage
Correlated storage
Product affinity
Analytical and simulation modeling
Performance analysis

A B S T R A C T

Order picking is one of the most demanding activities in many warehouses in terms of capital and labor. In parts-to-picker systems, automated vehicles or cranes bring the parts to a human picker. The storage assignment policy, the assignment of products to the storage locations, influences order picking efficiency. Commonly used storage assignment policies, such as full turnover-based and class-based storage, only consider the frequency at which each product has been requested but ignore information on the frequency at which products are ordered jointly, known as product affinity. Warehouses can use product affinity to make informed decisions and assign multiple correlated products to the same inventory "*pod*" to reduce retrieval time. Existing affinity-based assignments sequentially cluster products with high affinity and assign the clusters to storage locations. We propose an integrated cluster allocation (ICA) policy to minimize the retrieval time of parts-to-picker systems based on both product *turnover* and *affinity* obtained from historical customer orders. We formulate a mathematical model that can solve small instances and develop a greedy construction heuristic for solving large instances. The ICA storage policy can reduce total retrieval time by up to 40% compared to full turnover-based storage and class-based policies. The model is validated using a real warehouse dataset and tested against uncertainties in customer demand and for different travel time models.

1. Introduction

Warehouses decouple supply from demand in supply chains. To increase the efficiency of warehouse operations, we focus on the order picking process. Order picking is the central activity that requires much labor in picker-to-parts systems and much capital in parts-to-picker systems (Goetschalckx and Ashayeri, 1989; De Koster et al., 2007). The storage assignment policy, which determines how products are assigned to storage locations, influences order picking efficiency. Storage assignment and order picking are even modeled as integrated problems (Silva et al., 2020). Several storage assignment policies that improve the order picking are used in practice. Below, we review the main policies.

The random storage policy is the most commonly used and has been widely studied in the literature (Hausman et al., 1976; Bozer and White, 1984; De Koster et al., 2008). A system applying such a policy randomly allocates products to available storage locations. The class-based (ABC) storage policy considers two product attributes to assign products to storage locations, namely *turnover speed*

* Corresponding author at: PO Box 513, Atlas 4.423, 5600 MB Eindhoven, the Netherlands.
E-mail address: m.mirzaei@tue.nl (M. Mirzaei).

https://doi.org/10.1016/j.tre.2020.102207
Received 24 July 2020; Received in revised form 23 October 2020; Accepted 20 December 2020
Available online 12 January 2021

and *storage space needed*. The former refers to how often a product is ordered in a certain period. The latter is the average amount of space that is required to store the inventory of the product. The Cube-per-Order Index (COI) ranks products based on these two attributes and determines how frequently a product is requested per unit of stock space required (Heskett, 1963). In the ABC policy, products are first classified into several turnover frequency classes (commonly two or three: A, B, and C) based on their COI. Then, the product class with the highest COI is assigned to a block of storage locations (a storage zone) that is most readily accessible. Within each zone, products of the same class are assigned randomly to the locations. In the full turnover-based (FTB) policy, products are sorted in descending order based on their COI and assigned to storage locations that are sorted in ascending order, based on the travel distance to the depot (Yang et al., 2017). As a result of assigning fast moving products to locations close to the depot, the ABC and FTB policies can reduce picking travel times to frequently requested products. For convenience, a list of acronyms used in this paper is available in Table 6 in Appendix A.

Compared to the random storage policy, the ABC and FTB policies can reduce picking travel time substantially by considering product turnover. However, they ignore the *affinity* between products in historical orders. The affinity of two products is defined as their "correlation" in customer orders, i.e., how frequently the two products are ordered together. Next to turnover speed and storage space needed, affinity is another important attribute of the products, which can be derived from order history and/or demand forecasts. If this attribute is ignored in the assignment, two products that are frequently requested together (e.g. peanut butter and jelly) might be assigned to storage locations far from each other, which can unnecessarily increase order picking time. By considering the affinity between the two products, it is possible to cluster them in several groups based on this correlation. These clusters then can be allocated to the storage locations to reduce order picking time. An application of demand-correlation in apparel industry is studied by Frazelle (2016). Order batch picking may also benefit from storing multiple products at the same storage location (Yang et al., 2020).

In robotic mobile fulfillment (RMF) systems (Roy et al., 2019; Lamballais Tessensohn et al., 2020), products are typically stored on mobile inventory pods, each holding several dozens of products. When a product is needed for a customer order, a robot transports the entire inventory pod, shown in Fig. 1(a), to a picking station. The system can benefit from clustering the products ordered together frequently on the same pod. Amazon Robotics and Swisslog's CarryPick are two examples of RMF systems. Clustered storage can also be found in some other parts-to-picker retrieval systems. In automated storage and retrieval (AS/R) systems a storage bin is retrieved by a crane and then transported to a pick station. Fig. 1(b) shows an AS/R system with aisle-captive cranes and products stored in bins. Multiple small products can be stored in a single bin, each in a separate compartment. Fig. 1(c) shows a storage bin with nine compartments, each containing a different product. To fulfill customer orders, the crane needs to visit the locations with bins that store the requested products. In both the RMF and AS/R system, the number of these location visits can be reduced by storing highly correlated products in the same bin or pod. If each storage pod or bin carries multiple stock keeping units (SKUs), these systems can fulfill orders by retrieving a smaller number of pods or bins, compared to single-SKU pods or bins. This can significantly increase the throughput capacity. The same holds for other types of AS/R systems, like shuttle-based storage and retrieval, SBS/R (also called autonomous vehicle storage and retrieval) systems, or carousel systems. SBS/R systems deploy shuttles to retrieve bins with products and carousels store items in bins on vertically rotating shelves that are displayed to the picker.

The dominant approach of cluster-based allocation in the literature, which decomposes the travel time minimization problem into a clustering problem and an allocation problem, consists of two sequential steps. In the first step, products are clustered based on their correlation. In the second step, these clustered products are assigned to locations close to each other. Note that optimizing both problems in the decomposition approach does not guarantee an overall optimal solution. We show this by using an illustrative example. Assume that four products A, B, C, and D are stored in a warehouse. A small set of customer orders is given: {AC, ABD, BC, AD, ABD, AC, BC, AB, BC, AD, ABD}, with resulting popularity of 8, 7, 5, and 5 for products A, B, C, and D, respectively. Each cluster can contain two products. Clustering them using the decomposition approach leads to the following result. The most popular product, A, has the highest correlation with D, with five joint requests. Therefore, A and D form the first cluster, and B and C form the second cluster. The first cluster with the total popularity of 13 is allocated to the first storage location at one distance unit from the depot, and the second cluster with the total popularity of 12 is allocated to the second storage location at two distance units. Picking all orders, for example, in an AS/R system sequentially, requires 26 distance units of travel, whereas the travel distance can be reduced to 25 units by swapping the locations of the clusters.

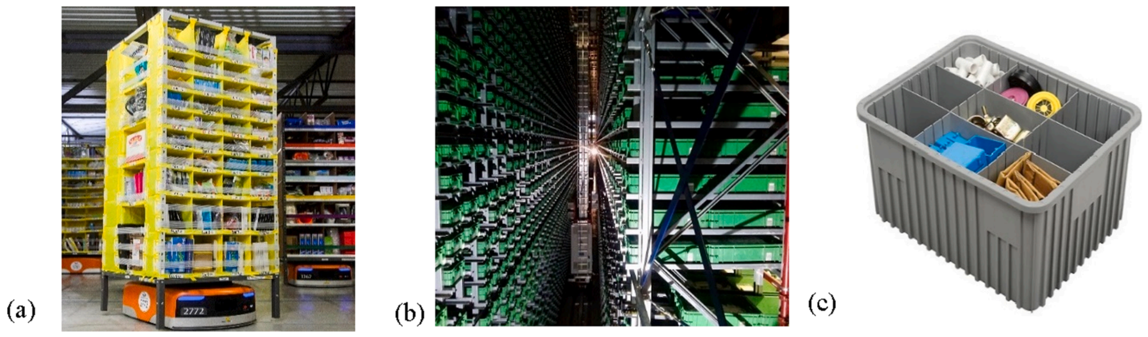

Fig. 1. (a) An Amazon robot carrying a storage pod (source: Business Wire, 2014), (b) an AS/R system, (c) a storage bin with sub-bins (source: Akro-Mils, 2019).

Decomposing the clustering and allocation problems is a common approach (Garfinkel, 2005; Chiang et al., 2014; Zhang, 2016) and picker-to-parts systems are the focus of the existing models (Bindi et al., 2007; Xiao and Zheng, 2012; Li et al., 2016). However, automated parts-to-picker warehousing systems are better suited for the growing ecommerce sector than the traditional picker-to-parts systems (Boysen et al., 2019). This paper proposes an integrated cluster allocation (ICA) storage assignment policy for parts-to-picker systems that minimizes the total order retrieval time by considering both product turnover and affinity concurrently. We examine storage systems where each storage location (e.g. storage bin) accommodates multiple products. We develop a model to optimally solve small instances and propose a construction heuristic for real-size problems. The model is validated using a real warehouse dataset and tested against disturbances in customer orders.

Note that the choice of storage policy also impacts the storage space requirements of warehouses. Random and ABC storage policies not only provide benefits such as easy implementation and replenishment, but they also provide high space efficiency (compared to e. g. FTB storage) as products can share the storage space in such systems. According to Hausman et al. (1976) and Rosenblatt and Eynan (1989), for an infinite number of products per storage class, the required storage space per product is equal to the average inventory level. Therefore, for a limited number of products, the required storage space varies between the quantity ordered from the supplier and the average inventory level (Yu et al., 2015). In the literature, the space-saving effect of the random and ABC policies compared to the FTB policy is either ignored (Heskett, 1963, 1964) or underestimated (Hausman et al., 1976; Eynan and Rosenblatt, 1994; Yu and de Koster, 2009). Space-sharing has a high impact on space requirements and consequently on the dimensions of the system and travel times. To provide a realistic and more accurate comparison of different systems, we include space-sharing in our numerical experiment. This is further explained in Section 4.1. The remainder of this paper is structured as follows. Section 2 reviews the literature on storage assignment policies based on product affinity and turnover. Section 3 introduces the mathematical model. Section 4 describes extensive numerical experiments to evaluate the performance of the proposed policy, and Section 5 concludes the paper.

2. Literature review

Analysis and evaluation of the impact of design parameters on performance of automated systems such as AS/R and SBS/R (also called AVS/R) systems has received much attention (Marchet et al., 2012, 2013; Borovinšek et al., 2016; Lerher et al., 2017; Ekren et al., 2018; Lerher, 2018; Küçükyaşar et al., 2020). Since we focus on the impact of an integrated cluster-based storage policy on the load retrieval time, this section first reviews literature on the impact on retrieval time of general storage assignment policies followed by cluster-based storage assignment policies.

2.1. General storage assignment

A storage assignment policy determines the allocation of products to storage locations and impacts the load retrieval time of the storage system (Roodbergen and Vis, 2009). Hausman et al. (1976) and Graves et al. (1977) compare the performance of an automated storage system for three storage policies, namely random, ABC, and FTB. The results show that turnover-based policies perform better for a unit load warehouse and has the lowest expected travel time. The ABC assignment also offers comparative benefits as it allows weaker assumptions regarding product turnover frequencies (they do not have to be constant). Ekren et al. (2015) study the best rack design when an ABC policy is applied to SBS/R systems to maximize utilization of lifts and shuttles and reduce cycle times. Yu et al. (2015) prove that the FTB policy is not optimal if the total required storage space does not equal the average inventory level of all products. In fact, given a finite number of products, a small number of storage classes already results in the minimum picking travel time. More storage classes require extra storage space. Weidinger and Boysen (2018) propose a scattered allocation that spreads product units in the warehouse to increase the likelihood of having some items located nearby. Lamballais Tessensohn et al. (2020) study the effect of spreading the inventory of a product among multiple pods on order throughput time in an RMF system.

2.2. Cluster-based storage

Products in customer orders may be correlated, and it may not be optimal to assign the premium locations to products with higher turnover. Product clustering in storage policies are applied in some parts-to-picker systems. Kim (1993) clusters correlated items in a mini-load AS/R system using a heuristic to minimize the material handling and inventory costs. Kress et al. (2017) partition the SKUs using mathematical models to minimize the total number of product groups that have to be accessed to pick customer orders. These models are particularly interesting for carousel systems and mobile racks, where one aims to minimize the number of partitions visited. We review the papers that study picker-to-parts systems below.

Some researchers (Frazelle, 1989; Sadiq et al., 1996; Ballou, 2004) have examined family-group based assignment policies, where certain products can be grouped and allocated to a subsection of the warehouse according to some shared properties. This strategy is common in retail warehouses, where the objective is to minimize shelf replenishment time. Amirhosseini and Sharp (1996) introduce several measures to find the correlation between products. An optimal assignment is possible by specifying all combinations pairs, which only works for warehouses with a very limited number of products. Frazelle (1989) proposes a heuristic for the stock location assignment problem that minimizes the order picking travel time by looking at the correlation between products. He applies a decomposition approach where in the first step, products are sequentially clustered, beginning with the most popular products and adding the highest correlated products until the capacity of the cluster is reached. In the second step, clusters with the highest total popularity are allocated to the closest available locations. Amirhosseini and Sharp (1996) propose another decomposition approach using a simultaneous clustering heuristic, in which two clusters with the highest correlation repeatedly merge until they reach the maximum allowed size. Zhang (2016) creates storage clusters using a *sum-seed* to calculate the correlation between all the remaining SKUs and those allocated to the cluster. He also introduces a *static-seed* that selects a SKU with the highest turnover frequency as the seed and assigns products with the highest correlation with the seed to the current cluster. Zhang et al. (2019) use the demand correlation pattern in a manual picker-to-parts warehouse to assign the products ordered together to storage locations and to simultaneously determine the pick route. To reduce the complexity of the mathematical model, they use an S-shape routing and solve it using simulated annealing. Accorsi et al. (2012) cluster correlated products using the nearest and farthest neighborhood algorithms. They then, assign clusters to locations according to the COI and other prioritization indices calculated for each cluster.

Sadiq et al. (1996) consider a correlated assignment in a dynamic environment and propose the dynamic stock location assignment algorithm that addresses product re-slotting when demand changes and product life cycles are short. They show that this algorithm performs better than a static COI rule. They use a two-step hierarchical clustering that improves the clusters. Sharp et al. (1998) propose a heuristic that improves existing hierarchical clustering algorithms. It is used to assign an assortment up to 700 products. Garfinkel (2005) studies correlated storage for zone picking, where products are assigned to the zones based on their correlation to minimize the number of zones visited for all orders. Jane and Laih (2005) define similarity measurement as coappearance of two items in the order set and use it to spread similar items over different zones. They maximize the utilization of a *synchronized* zone order picking system by formulating an integer program and solving it using a heuristic. Xiao and Zheng (2012) compute item correlation from the bill of materials and use it in a mathematical model to minimize zone visits. They use heuristics and a genetic algorithm to solve the model. Xiao and Zheng (2010) use the same correlation measure and present a mathematical model to minimize the travel distance of the bill of materials tours in the production warehouse. Jewkes et al. (2004) use a stochastic model for product assignment and picker allocation to minimize the cycle time of picking random orders from storage bins that store multiple products.

Rao (1971) and Hansen and Jaumard (1997) use a mixed-integer program to assign products to clusters to maximize the total affinity in the clusters. Such a model can be used in the first step of the decomposition approach since it does not allocate the products to storage locations. Liu (1999) examines these clustering techniques and proposes a primal-dual algorithm to solve the general clustering model as formulated by Rao (1971). Chiang et al. (2011) propose the data mining-based storage assignment approach that uses a fitness value that is a function of correlation, turnover, and distance as an association index. Chiang et al. (2014) introduce a new measure for correlation between products called weighed support count, which is maximized by applying two heuristics. First, the modified class-based heuristic maximizes the aggregated score within each storage zone, and then the association seed-based heuristic maximizes the aggregated score within each aisle. Bindi et al. (2009) take a data mining approach to define a similarity measure that is used in a clustering algorithm and assignment rules. In a case study, they show that a similarity-based strategy performs better than class-based and random strategies. Wang et al. (2020) formulate a storage location assignment model that minimizes the total travel distance for fulfilling customer orders. They solve it using a local search heuristic that swaps item pairs to improve the travel distance, according to the information in the order data. Li et al. (2016) use a product affinity-based technique in a dynamic storage assignment problem. A greedy genetic algorithm is used to solve the mathematical model that maximizes the total affinity between products of each zone and the total weighted popularity (a higher weight is given to zone *A* compared to zone *B* and *C*). Zhang et al. (2020) present

Table 1
Overview of papers using a cluster-based storage allocation policy.

Paper	Parts-to-picker	Cluster Allocation Approach	Objective	Solution
Wang et al. (2020)	No	Decomposition	Travel distance	Heuristic DBA[1]
Foroughi et al. (2020)	No	Decomposition	Travel distance/Aisles accessed	Optimal/Heuristic
Accorsi et al. (2012)	No	Decomposition	Travel distance/Aisles crossed	Simulation and what-if analysis
Garfinkel (2005)	No	Decomposition	Number of zone visits	Heuristic
Frazelle (1989)	No	Decomposition	Travel time	Heuristic
Jane and Laih (2005)	No	Decomposition	Zone picking utilization	Heuristic
Zhang (2016)	No	Decomposition	Travel distance	Heuristic, sum and static seed
Xiao and Zheng (2010)	No	Decomposition	Travel distance	Multi-stage heuristic
Xiao and Zheng (2012)	No	Decomposition	Number of zone visits	Heuristic, GA
Liu (1999)	No	Decomposition	Sum of similarity measure	Heuristic
Chiang et al. (2011)	No	Semi-integrated	Sum of fitness values	Heuristic, DMSA[2]
Chiang et al. (2014)	No	Decomposition	Weighed support count/travel distance	Heuristic, MCBH[3] and ASBH[4]
Li et al. (2016)	No	Semi-integrated	Sum of affinity and turnover/travel distance	GA
Amirhosseini and Sharp (1996)	No	Decomposition	Travel distance	Heuristic
Sharp et al. (1998)	No	Decomposition	Travel distance	Heuristic
Sadiq et al. (1996)	No	Decomposition	Travel time	Heuristic
Bindi et al. (2007)	No	Decomposition	Travel distance	Heuristic
Jewkes et al. (2004)	No	Decomposition	Order cycle time	Greedy heuristic
Kim (1993)	Yes	Decomposition	Handling/inventory costs	Heuristic
Kress et al. (2017)	Yes	Decomposition	Number of groups accessed	Heuristic, Branch and Bound
This paper	Yes	Integrated/Decomposition	Retrieval time	Optimal/SA[5] heuristic

[1] Data-based approach. [2] Data mining-based storage assignment. [3] Modified class-based heuristic. [4] Association seed-based heuristic. [5] Sequential alternating heuristic (refer to Section 3.2).

a mixed-integer program for a dynamic slotting problem to minimize the fulfilment time. They increase the likelihood of finding correlated items that are ordered in a period of few hours in a short pick route. Foroughi et al. (2020) minimize the number of aisles accessed and the total distance covered in mobile racks by modeling a dynamic program for picker routing and heuristic priority rule for product allocation based on joint orders. Table 1 shows an overview of studies on cluster-based assignment policy and highlights the research gap.

Most of the studies summarized in Table 1 apply a storage allocation in picker-to-parts systems using affinity between products. They decompose the problem by first clustering products based on some measure of demand affinity between products, followed by assigning the clusters to storage locations. These decomposed problems do not necessarily optimize the same objective function. Semi-integrated approaches in the literature (Chiang et al., 2011; Li et al., 2016) combine the information on product-demand affinity, product-demand turnover, and distance into a fitness value measure. This paper contributes by introducing an integrated cluster allocation (ICA) approach to parts-to-picker systems which clusters products and allocates them to storage locations simultaneously, using both product turnover frequency and affinity in historical demand information. Additionally, in a scenario-based analysis, we investigate for which order profiles an ICA policy is beneficial compared to common policies like a turnover frequency-based policy and a sequential approach. To solve large instances and validate the model, a SA heuristic is proposed. In the evaluation, the effect of space-sharing by products, and applying different travel time models are taken into account. In the next section, we propose the mathematical model for the integrated approach.

3. Problem description and mathematical formulation

To reduce order picking travel time, products can be clustered in the storage area based on the correlation observed in the order history. Products of each cluster are assigned to a location consisting of multiple sub-locations. We make the following assumptions:

- Each product is assigned to only one cluster.
- Each storage location (e.g. bin) consists of multiple sub-locations (e.g. sub-bins). Each sub-bin can store one product.
- The order history is sufficiently large to accurately capture product turnover and affinity.
- The retrieval machine (i.e., AS/R crane or a robot) transports an entire storage bin or pod to the depot.
- The total storage capacity is sufficient to accommodate all products.

We propose a mathematical model for the integrated cluster allocation policy that uses historical order set O to allocate product $i \in P$ to the cluster (a storage bin) at location $l \in L$, to minimize total retrieval time. We use the following notations.

Parameters:	
P	the set of products in the assortment,
O	the set of customer orders over a certain period of time,
L	the set of available storage locations in the system,
Z	the set of zones in the warehouse,
Π	the sorted storage location set L, based on ascending distance to the depot,
C_l	the number of sub-locations (sub-bins) available in the cluster (bin) at location $l \in L$,
I_i	the number of sub-bins needed to store the required inventory of product $i \in P$,
t_l	the one-way travel time from the depot to location $l \in L$,
ρ_{ij}	affinity between product $i, j \in P$,
r_{oi}	$= 1$if order $o \in O$ contains a request for product $i \in P$, $r_{oi} = 0$ otherwise,
τ_i	turnover frequency of product $i \in P$,
Q_i	the order quantity of product $i \in P$,
N_k	the number of items that share storage zone $k \in Z$,
$a_i(N_k)$	the average required space to store item $i \in P$ in class $k \in Z$ together with $N_k - 1$ other items,
ε	the space-sharing factor in the class-based policy.
Variables:	
x_{il}	$=1$ if product $i \in P$ is assigned to the cluster at location $l \in L, x_{il} = 0$ otherwise,
z_{ol}	$=1$ if picking order $o \in O$ requires a visit to location $l \in L, z_{ol} = 0$ otherwise.

The Integrated cluster allocation (ICA) model is formulated as follows:

$$min \sum_{o \in O} \sum_{l \in L} t_l z_{ol} \tag{1}$$

Subject to

$$\sum_{l \in L} x_{il} = 1, \forall i \in P, \tag{2}$$

$$\sum_{i \in P} I_i x_{il} \le C_l, \forall l \in L, \tag{3}$$

$$z_{ol} \geq r_{oi}x_{il}, \forall i \in P, \forall o \in O, \forall l \in L, \tag{4}$$

$$x_{il} \in \{0,1\}, \forall i \in P, \forall l \in L, \tag{5}$$

$$z_{ol} \in \{0,1\}, \forall o \in O, \forall l \in L. \tag{6}$$

The objective function (1) minimizes the total travel time to pick all customer orders, depending on the clusters and product allocation in the model. Therefore, products with high affinity are assigned to the same cluster, and products with higher turnover frequency are stored closer to the depot. Constraints (2) ensure that each product is assigned to exactly one storage location. Constraints (3) ensure that the capacity of each storage location is not exceeded. Constraints (4) guarantee necessary visits to storage locations of the requested products for all customer orders. Constraints (5), (6) define the binary variables.

3.1. Solution approaches

In terms of complexity, the model in Section 3 consists of two problems, namely clustering and assigning the products. Since the assignment problem is comparable to the bin packing problem, which is proven to be NP-hard (Garey and Johnson, 1979), it can be shown that this model is NP-hard too. The solution approach consists of using a general optimization solver and a greedy heuristic that can solve large instances. Solution approaches in the literature (see Table 1) generally decompose the problem into two sub-problems, clustering the correlated products and then assigning the clusters to the storage locations or the zones. These methods are developed for picker-to-parts warehouses and are not directly applicable to parts-to-picker warehouses where correlated products on the same storage bin or pod are moved together using automation technologies. In Section 3.2, we develop a sequential heuristic approach fit for parts-to-picker warehouses that uses both affinity and turnover frequency by alternating between allocation and clustering. This heuristic is used for solving real-size warehouse cases.

3.2. A sequential alternating (SA) heuristic

In this section, we develop a fast heuristic solution, inspired by existing picker-to-parts sequential methods (Garfinkel, 2005; Zhang, 2016), but adapted to parts-to-picker systems. To consider both turnover frequency and affinity of the products, the heuristic alternates between assigning products to locations and clustering products. Clusters may be defined according to several affinity measures (see Amirhosseini and Sharp, 1996). We use Expression (7) to obtain the affinity between products i and j.

$$\rho_{ij} = \frac{\sum_{o \in O} r_{oi} \times r_{oj}}{|O|}, i,j \in P, i \neq j, \tag{7}$$

where $r_{oi} = 1$ if order $o \in O$ contains a request for product $i \in P$, $r_{oi} = 0$ otherwise.

This heuristic is described in Algorithm 1. Note that empty bins are preassigned to locations, so we refer to these as locations. Let set Π be the sorted storage location set L, based on ascending distance to the depot. Each storage location consists of multiple sub-locations that can house a cluster of products. The algorithm assigns product i with the highest turnover frequency to an available sub-location closest to the depot. Then, if the remaining capacity of the location allows, products with the highest affinity with the assigned product, ρ_{ij}, are added to this cluster. The capacity of the location and the set of unassigned products are updated each time. In the case of more than one product with the highest correlation with i, product j with the highest τ_j is selected (Line 13). If product i is not correlated with any other products (Line 14), the product with the highest turnover frequency is assigned to the respective cluster. In this way, highly correlated items are clustered, whilst popular products are assigned to locations closer to the depot. This procedure is repeated until all the products are assigned to locations and clusters.

Algorithm 1. (*Pseudocode for a sequential alternating (SA) heuristic*)

```
1: input $L$, $t_l$, $C_l$, $I_i$, $P$, $\rho_{ij}$ and $O$.
2: Compute $\tau_i, \forall i \in P$ over $O$.
3: Compute $\rho_{ij}, \forall i,j \in P$ over $O$.
4: $\Lambda$ =: set of products $i \in P$ sorted in descending $\tau_i$.
5: $\Pi$ =: set of locations $l \in L$ sorted in ascending $t_l$.
6: while $\Lambda \neq \emptyset$ do
7:     Assign the first $i \in \Lambda$ to the first $l \in \Pi$, $x_{il} = 1$.
8:     $\Lambda = \Lambda - \{i\}$.
9:     $C_l = C_l - I_i$.
10:    while $C_l > 0$ do
11:        Select $j := \mathrm{Argmax}\{\rho_{ij} | j \in \Lambda, I_j < C_l\}$.
12:        if $j$ is not unique then
13:            assign the one with highest $\tau_j$ to location $l$, $x_{jl} = 1$.
14:        else if $\rho_{ij} = 0$ then
15:            assign the first $j \in \Lambda | I_j < C_l$ to location $l$, $x_{jl} = 1$.
```

(continued on next page)

(*continued*)

```
        else assign $j$ to location $l$, $x_{jl} = 1$.
        end if
        $\Lambda = \Lambda - \{j\}$.
        $C_l = C_l - I_j$.
    end while
    $\Pi = \Pi - \{l\}$.
end while
return all $x_{il}$
```

3.3. *Exact solution to the ICA model*

The model of the ICA policy is programmed in AIMMS and solved using Gurobi 7.5. The solution generated by Algorithm 1 is an initial solution to the model to speed up the procedure. Since the problem is NP-hard, large instances are not solvable to optimality. Using a machine running Windows 7 with 4 GB of RAM, we were able to solve instances comprising up to 300 orders and 500 products in two hours.

4. Numerical analysis

This section investigates the impact of parameters such as customer demand and cluster size on the performance of the ICA policy. We examine the application of the ICA policy in two types of systems: in a conventional warehouse using an AS/R system and in a modern warehouse using an RMF system, such as the ones by Amazon. We take a realistic approach by including the effect of space-sharing in the class-based storage system, as explained in Section 4.1. This leads to smaller storage space and shorter average travel times in such systems, compared to the FTB and ICA policies that do not use the storage space as flexibly. Section 4.2 describes the sampling procedure. Sections 4.3 and 4.4 discuss the results for the AS/R and RMF systems, respectively. Section 4.5 evaluates the performance of the ICA policy in presence of disturbances in demand pattern and under different travel time models. Section 4.6 uses a real warehouse dataset to validate the cluster allocation policy. Section 4.7 discusses the results and managerial implications.

4.1. *Space-sharing in the class-based policy*

We compare the ICA policy with the sequential alternating heuristic, the FTB, and the ABC class-based storage systems. Although FTB and ICA storage both have the advantage of more precise assignment and shorter travel time, ABC storage has an advantage of space-sharing. To incorporate the space-sharing effect in our experiment, we use the formula of Yu et al. (2015). They show that the required storage space for product i in storage zone $k \in Z$, where Z is the set of zones, with a given order quantity of Q_i sub-bins is estimated as:

$$a_i(N_k) = 0.5\left(1 + N_k^{-\varepsilon}\right)Q_i, \tag{8}$$

where $N_k, k \in Z$ is the number of items that share storage zone k, and $0 < \varepsilon \leq 1$ is the space-sharing factor. Safety stock is excluded in all policies. Therefore, the total required storage space needed for storing all products is $\left\lceil \sum_i a_i(N_k) \right\rceil$, expressed in number of sub-bins. Although ε depends on initial inventory, the Pareto demand curve, and other factors, Yu et al. (2015) show that it is fairly constant and that it can be estimated at between 0.17 and 0.25. We assume $\varepsilon = 0.20$ in our experiment. The storage space requirement of all items in the class-based policy is adjusted according to Eq. (8). We calculate the space requirement for the AS/R and RMF systems upfront according to the policy. Therefore, the dimensions and travel time of the system are known according to the assortment, inventory, and storage policy. For instance, in our base example, using a class-based policy consisting of two classes, the required space per product (in sub-bins) equals 69% and 67% of the economic order quantity for each item in zones A (120 SKUs) and B (180 SKUs), respectively. These values are adjusted in the experiments accordingly when the number of products or classes changes. Note that in the ICA and FTB policies space cannot be shared, so for each product i space for the whole lot size Q_i must be reserved.

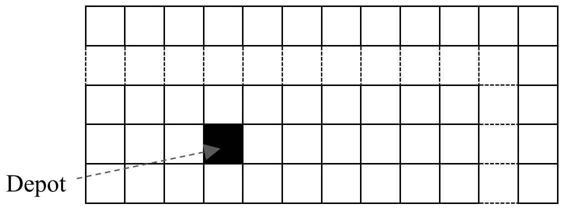

Fig. 2. A side view of the rack in an AS/R system showing the storage locations and an example travel path of the crane.

4.2. Sample generation

We consider two system configurations for AS/R and RMF systems. A base example and several scenarios are generated for the experiment.

AS/R Systems: We consider a single-deep mini-load AS/R system, like the one shown in Fig. 1(b), consisting of one aisle with racks on both sides. Fig. 2 shows a side view of the racks. The required one-way travel time in seconds to reach each storage location, according to a Chebyshev metric distance (i.e., the maximum of vertical and horizontal distance) is used to compute the travel time. The black square shows a requested bin which is retrieved by the crane traveling on the dashed arrow. The crane drives and lifts/lowers simultaneously. The horizontal speed of the S/R crane is 2 m/s, the vertical speed is 0.5 m/s, and each storage location (slot) is 1 m wide and 1 m high. Acceleration, deceleration, pick up and drop off times are ignored. The system dimensions are square in time, i.e., the travel times of the crane from the depot to the farthest horizontal and vertical storage locations are equal. Popular products take up two sub-bins, and less popular products take up one sub-bin.

RMF Systems: We consider a robotic mobile fulfilment system, where products are stored on pods grouped in blocks, and where the pods are transported using robots. Fig. 3 shows an example of the system layout in this experiment. It consists of six blocks of ten storage pods. The system is flexible and the number of blocks and the number of pods in each block can easily be adapted. A robot travels to one of these locations and takes a pod that contains one or more of the requested products to the depot. The robot returns the pod to its location after picking. The black square shows a requested pod. The dashed line shows the path of the robot to retrieve the requested load. To avoid deadlocks and reduce congestion, aisles are one-directional, except the front and back aisles. To retrieve a pod, the empty robot can travel underneath the pods. There are one robot and one pick station. This assumption does not limit the results as we focus on the total travel time. The average speed of the robot is 1.5 m/s, and the required time for a full turn is 2.5 s. Acceleration, deceleration and the time needed for lifting or storing a pod are ignored since they are equal across the polocies. Each pod is 1×1 m and contains one cluster of products. The inventory level for popular products is two slots, and for less popular products one slot. The total storage capacity is enough to accommodate all products.

Dataset and Scenarios: To evaluate the performance of the ICA policy in these two systems, we use a dataset with customer demands of a wholesale distributor of office stationery products, available from Warehouse Science (2019). This dataset consists of more than 50,000 orders and 16,000 products. We create a base example of 200 orders that capture the characteristics of this dataset, such as the average order size, turnover, and affinity. Table 2 shows the parameter values for the base example and different scenarios based on varying five different parameters: order size, assortment, number of sub-bins per bin, the skewness of the ABC curve, and the affinity between products. The defined ranges of these parameters are based on the original dataset. Algorithm 2 in Appendix B shows how the instances are generated. For each instance, we only vary one input parameter at a time. The order size has a discrete uniform distribution with the lowest value of 1 and the maximum value denoted in Table 2. Next, the number of SKUs in the assortment is varied. The number of sub-bins per bin, defining the cluster capacity, is also varied. We consider four distributions of the ABC curve: random, moderate skewness of 20/40, and higher skewness of 20/60, and 20/80. A 20/40 ABC curve means 20% of the products account for 40% of the demand. The affinity between products is measured according to Equation (7). We generate instances of zero, low, moderate, high and very high affinity by manipulating the chance of ordering a correlated product when a specific product is requested. Table 3 shows the frequency of affinity scores among all pairs of products for these generated instances. A zero affinity, which is generated by a set of single-line orders, is included to test an extreme case.

4.3. Results for the AS/R systems

We calculate the order picking travel time of the AS/R system for all scenarios. The execution time to solve the model of the ICA policy is limited to two hours per instance except for the base case, which was run two days to obtain a smaller optimality gap. The linear programming (LP) relaxation is used to obtain a lower bound on the solution quality of the ICA algorithm. The execution time of the SA heuristic is a fraction of a second. Table 4 shows the results for the scenarios defined in Table 2. The first column in each part of the table shows the parameter varied. The *SA, ABC, FTB, and ICA* columns show the one-way travel time in seconds for the four solutions, respectively. The *LP Gap column* shows the ICA solution gap with the linear program relaxation bound, which is obtained by relaxing constraints (6) and (7). The last three columns show the relative savings of the ICA solution, compared to each of the other

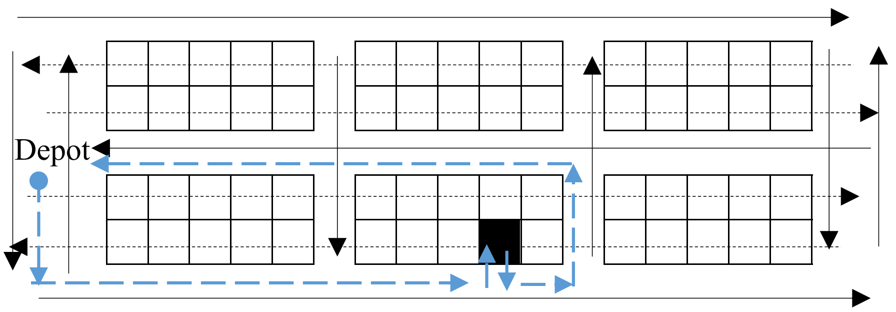

Fig. 3. A top view of the RMF system.

Table 2
Parameters related to base example and scenarios of the AS/R and RMF systems.

Parameters	Base examples	Range for scenarios
Maximum order size	3	1, 2, 3, 4
Assortment size (# SKUs)	300	100, 200, 300, 500
Number of sub-bins per bin	6 (AS/R), 10 (RMF)	1, 2, 4, 6, 8, 16
Skewness of ABC curve	20/40	Random (20/20), 20/40, 20/60, 20/80
Affinity	Moderate	zero, low, moderate, high, very high

Table 3
Frequency per affinity score for the datasets when varying affinity.

	Instance set				
ρ	Zero	Low	Moderate	High	Very high
0.013	0	0	0	0	2
0.010	0	0	0	2	6
0.007	0	0	3	12	34
0.003	0	268	262	238	995
0.000	44,850	44,582	44,585	44,598	43,813

Table 4
Comparison of travel time of the four solutions (SA, ABC, FTB, and ICA) in an AS/R system.

Scenarios	SA (s)	ABC (s)	FTB (s)	ICA (s) [*]	LP Gap %	Savings % ICA/SA	Savings % ICA/ABC	Savings % ICA/FTB
1.Max. order size								
1	434.5	355.2	434.5	**433.7**	0.0	0.2	−22.1	0.2
2	647.0	556.1	675.0	560.0	1.7	13.4	−0.7	17.0
3	824.0	787.2	977.7	685.5	6.7	16.8	12.9	29.9
4	1044.5	994.2	1193.5	856.3	15.9	18.0	13.9	28.3
2.Assortment size								
100	648.3	559.5	706.7	555.0	13.2	14.4	0.8	21.5
200	759.5	719.0	879.2	653.2	10.0	14.0	9.2	25.7
300	824.0	787.2	977.7	685.5	6.7	16.8	12.9	29.9
500	933.5	945.5	1119.2	725.7	5.4	22.3	23.2	35.2
3.No. of sub-bins								
(no cluster) 1	2248.0	1758.2	2248.0	**2248.0**	0.0	0.0	−27.9	0.0
2	1338.8	1301.5	1645.0	1212.7	10.9	9.4	6.8	26.3
4	877.8	890.7	1043.5	739.2	6.3	15.8	17.0	29.2
6	824.0	787.2	977.7	685.5	6.7	16.8	12.9	29.9
8	696.3	706.5	825.7	560.5	9.4	19.5	20.7	32.1
16	501.3	517.7	639.0	390.0	1.2	22.2	24.7	39.0
4.Affinity								
Zero	434.5	355.2	434.5	**433.7**	0.0	0.2	−22.1	0.2
Low	880.8	787.2	1044.2	692.2	11.6	21.4	12.1	33.7
Moderate	824.0	787.2	977.7	685.5	6.7	16.8	12.9	29.9
High	578.0	608.5	702.0	512.0	6.7	11.4	15.9	27.1
Very high	478.8	553.5	575.5	**441.0**	0.0	7.9	20.3	23.4
5.ABC curve								
(random) 20/20	1443.7	1282.2	1567.5	1160.7	10.6	19.6	9.5	26.0
20/40	1506.2	1353.2	1701.5	1265.2	9.4	16.0	6.5	25.6
20/60	1421.5	1366.2	1678.0	**1295.0**	0.0	8.9	5.2	22.8
20/80	1239.0	1138.2	1460.5	1211.7	12.5	2.2	−6.5	17.0

[*] The results for the ICA in bold print are optimal values.

three. The results for the base example are repeated in each part of the table to facilitate comparison. From Table 4, we can make the following observations.

Observation 1: The ICA and SA consistently perform equally or better than the FTB. This was expected since next to turnover frequency, the ICA and SA use affinity as a complementary criterion. In case of zero affinity (no clusters), they perform equally. However, ICA and SA do not always outperform the ABC. In the instances of zero affinity and a highly skewed ABC curve, the ABC is preferable to the ICA. In addition, the ABC performs better than the SA, except when affinity between products is high. This is mainly due to the space-sharing effect in the ABC, which reduces the storage space requirement and consequently decreases travel time.

Observation 2: In large order sizes, the ICA outperforms the other policies and the marginal performance improvement has a positive relation with the order size. In addition, the ICA policy outperforms both ABC and FTB policies even for small-sized orders (e.g. for orders with maximum two lines). This is because the ICA policy benefits from identifying affinity between products and assigns them to the same cluster to reduce travel time.

Observation 3: Savings of the ICA increase with a larger assortment. More products require a larger storage capacity which implies longer travel time for all policies. In a larger warehouse, the randomness in the ABC policy becomes a disadvantage, as ordered products now, can be assigned to the same turnover-based storage group far from each other and from the pick station.

Observation 4: A larger number of sub-bins (i.e., larger cluster sizes) per bin increases the benefits of the ICA policy. This is expected as a larger cluster size increases the chance of grouping highly correlated products. This observation supports the fact that clustering correlated products improves order picking performance.

Observation 5: Samples with a higher affinity lead to higher savings in the ICA. Higher affinity enables the ICA to construct strongly correlated product clusters and generate higher benefits. When the product affinity is very low, the ICA policy has no clear benefits. In this case ABC is the best policy, while FTB performs at the same level as ICA.

Observation 6: Savings of the ICA policy decrease for more skewed ABC curves. In such a situation, the ABC and the FTB policies, which primarily depend on turnover frequency of products, perform better and suffer less from neglecting affinity information.

Observation 7: The exact solution to the ICA policy outperforms the SA heuristic solution by13% on average. This shows that an integrated approach outperforms a sequential approach if historical data on product affinity and turnover frequency are used in the assignment.

4.4. Results for the RMF system

We follow the same procedure to obtain the order picking travel time of the RMF system for all scenarios. Table 5 presents the results. In general, we observe the same trends as in the results from AS/R systems. The results show that the ICA policy leads to higher savings when affinity, order size, assortment size, and cluster size increase. However, when the skewness of the ABC curve increases, we observe lower savings. These support the findings in the AS/R system.

4.5. Sensitivity of the ICA policy

This section evaluates the performance of the ICA storage policy when the customer demand pattern changes, and when different travel time models are used instead of the one-way single command model we used.

Sensitivity to Changes in Demand Pattern: First, we use the base example dataset to assign products to the storage locations using the ICA policy, for both the AS/R and RMF systems described in Section 4.2. Then, we replace several orders from the base example with new random orders generated following the characteristics of the base example such as order size, affinity and the ABC curve, using the method outlined in Algorithm 3 in Appendix C. We do this for 10% up to 100% change in the base example order set. Using Monte Carlo simulation, we generate 100 new order sets per scenario to calculate the savings in order picking travel time.

Fig. 4(a) shows the average savings achieved by applying the ICA policy in an AS/R system compared to FTB policy (top curve), and class-based policy (bottom curve). The dashed lines show the lower and upper bounds of 95% confidence intervals. Fig. 4(b) shows the

Table 5
Comparison of travel time of the four solutions (SA, ABC, FTB, and ICA) in an RMF system.

Scenarios	SA (s)	ABC (s)	FTB (s)	ICA (s) *	LP Gap %	Savings % ICA/SA	Savings % ICA/ABC	Savings % ICA/FTB
1.Max. order size								
1	3929.0	3200.5	3929.0	**3847.2**	0.0	2.1	−20.2	2.1
2	5443.9	4951.8	6241.2	**4633.0**	0.0	14.9	6.4	25.8
3	7228.5	6971.0	8573.3	5959.7	7.2	17.6	14.5	30.5
4	8987.9	8459.2	10579.8	7229.7	21.4	19.6	14.5	31.7
2.Assortment size								
100	5756.5	5122.6	6577.7	4956.0	14.5	13.9	3.3	24.7
200	7228.5	6971.0	8573.3	5959.7	7.2	17.6	14.5	30.5
300	7561.9	7985.2	9199.5	6038.2	6.2	20.1	24.4	34.4
500	8758.3	10246.8	10863.8	6557.3	3.7	25.1	36.0	39.6
3.No. of sub-bins								
4	10281.4	9679.9	11517.7	8648.8	11.2	15.9	10.7	24.9
8	7228.5	6971.0	8573.3	5959.7	7.2	17.6	14.5	30.5
16	5661.3	5512.0	6765.7	4593.7	8.4	18.9	16.7	32.1
4.Affinity								
Zero	3929.0	3200.5	3929.0	**3847.2**	0.0	2.1	−20.2	2.1
Low	7548.3	6769.6	8747.5	6056.5	11.9	19.8	10.5	30.8
Moderate	7228.5	6971.0	8573.3	5959.7	7.2	17.6	14.5	30.5
High	4881.5	5130.3	6281.2	4240.2	1.1	13.1	17.3	32.5
Very high	4350.8	4528.5	5931.3	**3867.3**	0.0	11.1	14.6	34.8
5.ABC curve								
(Random) 20/20	6961.3	6394.2	7934.5	5508.7	7.2	20.9	13.8	30.6
20/40	7228.5	6971.0	8573.3	5959.7	7.2	17.6	14.5	30.5
20/60	6857.6	6082.9	7727.5	5826.5	14.3	15.0	4.2	24.6
20/80	5961.5	5225.5	6546.7	5136.2	13.2	13.8	1.7	21.5

* The results for the ICA in bold print are optimal values.

average savings achieved in an RMF system. The horizontal axis represents the percentage change in the order set compared to the base case, and the vertical axis shows the savings percentage. These graphs show that the ICA policy performs well despite large changes in the demand pattern. More specifically, the ICA policy is still beneficial when the customer demand pattern is prone to random changes of up to 60% in the AS/R system and 70% in the RMF system.

Sensitivity to Travel Time Models: The model used in our numerical analysis considers one-way single command travel time, that is the travel time of a crane or a robot from the location of the requested load to the depot. Applying two-way or dual command travel time models, may affect the results of the policies. A two-way single command travel time considers the travel time of the crane from a dwell point (not necessarily the depot) to the location of the requested load and then, from there to the depot. In a dual command travel model, the crane performs consecutive storage and retrieval requests. Thus, the dual command travel time consists of traveling from the dwell point to the depot, from the depot to the location of the product that should be stored, travel time between the location of stored product and the requested product, and then from there to the depot. To evaluate the effect of different travel time models, we apply them to the base example for both the AS/R and RMF systems and compare the results. Fig. 5 shows the percentage relative savings of the ICA compared to the benchmark policies for one-way and two-way single command, and dual command travel time models. The results show that the ICA model consistently outperforms the other policies for all the travel time models. However, the benefits of the ICA decrease in a dual command travel model. This can be explained by the additional time for storage and travel between time. The results of two-way single command travel time are similar to those of one-way travel, as the crane or robot travels from a dwell point to the requested load in all the policies.

4.6. Validation using a real warehouse dataset

This section is dedicated to validating the ICA policy using the data of a real-life omni-channel warehouse distributing personal care products. This warehouse is expecting a steep growth in online retail and therefore requires a responsive and efficient policy for their automated storage system (an AS/R system). The retrieved data include the information of more than 2300 products and 28,000 orders. We use 70% of the order data, randomly selected, for training the model and the remaining portion for validation. We use the SA heuristic introduced in Section 3.2 to solve large instances. To evaluate the role of the ICA policy, we slightly modify the heuristic such that it assigns the storage bins to storage zones instead of dedicated storage locations. This means in Algorithm 1, the set of locations L with capacity C_l is replaced by a set of zones Z with capacity C_z. This modification allows the SA to benefit from space-sharing similarly to the class-based storage policy, and therefore, provides a better opportunity to evaluate the benefits of clustering correlated products.

The AS/R crane transports the storage bins containing the requested products to the pick station, which are then picked and packed in customer bins. Orders are released one at a time in the sequence they arrive, i.e., there are no order waves or batching. Each storage bin is currently used to store only one product. Bins are stored according to a class-based storage policy in two storage zones. We apply the ICA policy to this system by storing multiple correlated products in each bin and compare the results of the two policies.

The SA heuristic uses the training set to determine the clustering and allocation of the products. Then, we sample 500 order sets from the validation set each containing 300 random customer orders and calculate the retrieval times of the SA and ABC policies. This sampling is repeated several times such that a maximum order size is imposed on the samples each time. Fig. 6 shows the savings in the retrieval time that the warehouse can obtain by the SA solution compared to the ABC. The horizontal axis is the percentage of savings,

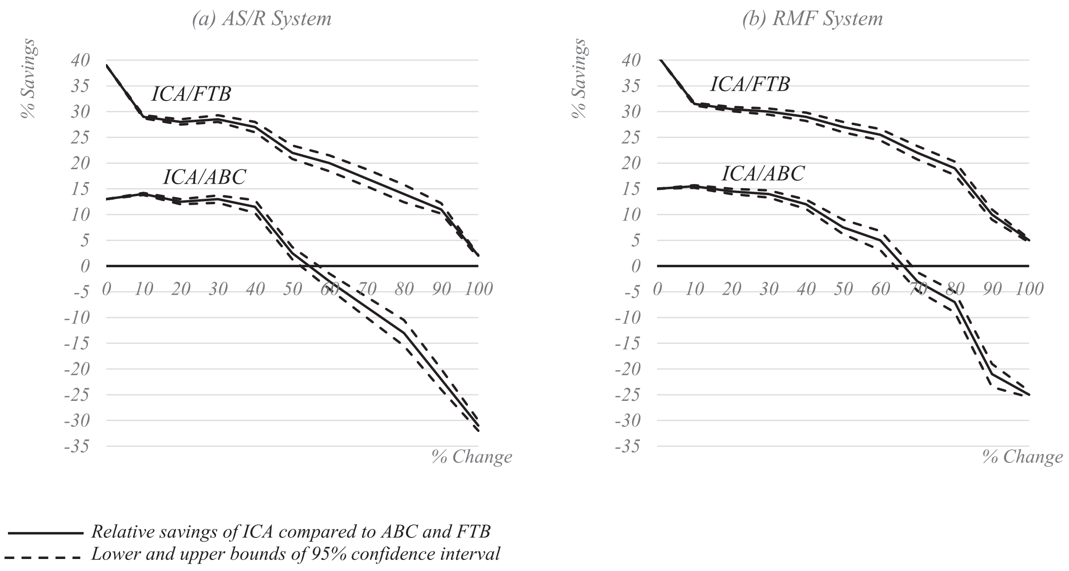

Fig. 4. Results of the Monte Carlo simulation of savings obtained using the ICA policy in (a) AS/R and (b) RMF systems, when the customer demand pattern changes.

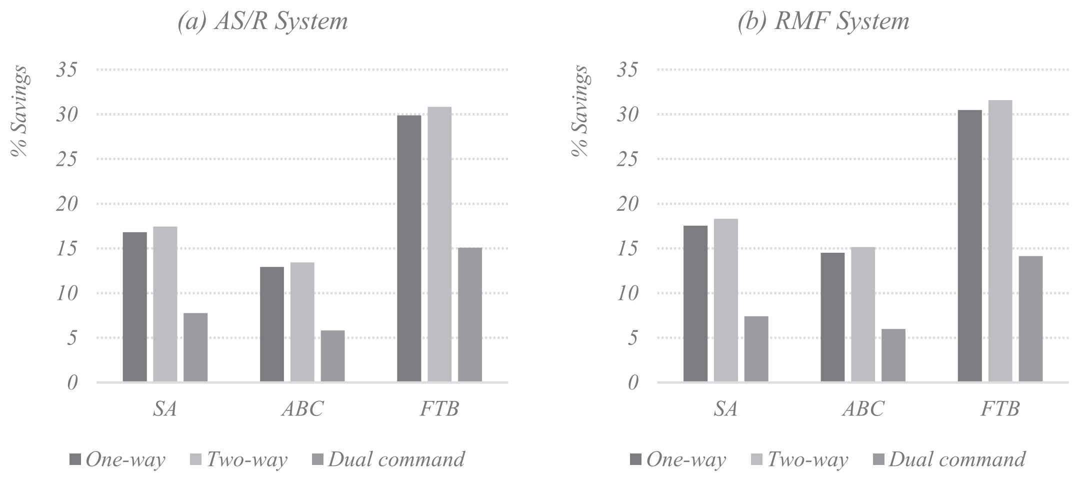

Fig. 5. Comparison of percentage savings obtained using the ICA policy for different travel time models in (a) AS/R and (b) RMF systems.

and the vertical axis is the maximum order size in the sampled order set. When increasing the maximum order size, the savings of the SA grow compared to the ABC policy. However, the SA's benefits converge for larger order sizes. The dotted and dashed lines show the average savings of the 10% sample order sets with the highest and lowest affinity, respectively. At a given maximum order size, the sample order sets with a higher affinity result in greater average savings than those with a lower affinity.

4.7. Managerial implications

The findings of this research show that using an integrated cluster-based allocation model based on historical customer demand may improve storage allocation and save order retrieval time. Clustering is particularly interesting for automated warehouses that deploy parts-to-picker systems. Fig. 7 shows a framework that helps managers of such warehouses to decide the best storage policy based on the skewness of the product turnover frequency curve and the relative average product affinity. Low (L_a) and high (H_a) affinity mean lower and higher than the moderate affinity in the base example, respectively. Under H_a/L_t condition, that is customer orders with high product affinity and low turnover frequency skewness, the ICA policy outperforms the ABC and FTB storage strategies (see top left corner of Fig. 7). Operations managers should consider a cluster-based policy when customers order multiple correlated products. Examples are fashion and apparel retail. Under L_a/L_t or H_a/H_t conditions, i.e. the bottom left and top right corners of the framework, there is no obvious choice of policies. Here, managers should further investigate the order profile and specifications of the storage system. For example, the ICA policy may still be favorable for large order sizes or when each storage unit can house a large number of products. Under a L_a/H_t condition, the bottom right corner of the framework, our results recommend a class-based storage policy because of its space-saving benefits. Here, managers should group products by turnover frequency and assign popular storage classes to locations close to pick stations. Benefits of clustering products is negligible.

Warehouse managers should analyze historical customer demand and decide prudently on storage policy as it has a significant impact on the throughput capacity and responsiveness of the warehouse. Our numerical analysis shows that selecting a suitable storage policy can decrease the retrieval time up to 40%. To use the cluster-based policy, the replenishment process should be adjusted accordingly, which might take additional time and effort. Implementing the ICA does not require technological adaptation in the

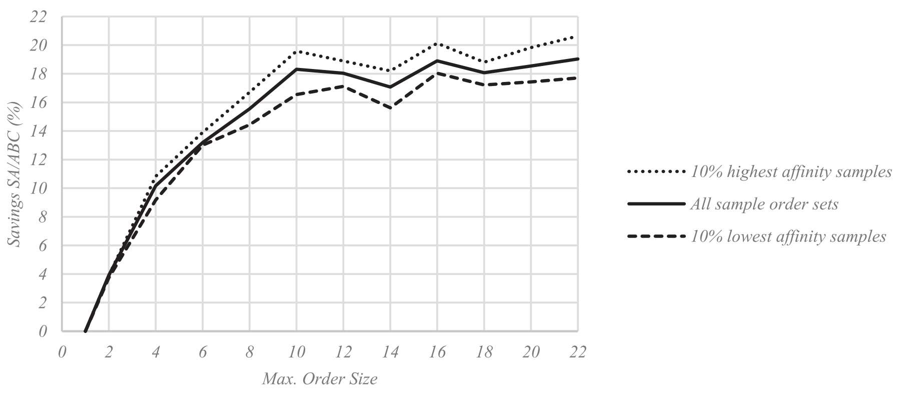

Fig. 6. Travel time savings of the SA solution compared to the ABC for different order sizes in a real system.

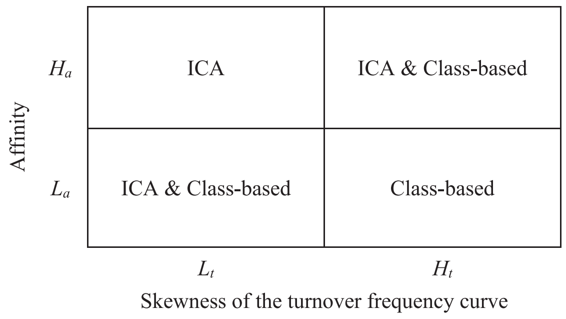

Fig. 7. A decision typology on storage allocation policy by considering the turnover frequency and affinity of products.

existing storage and retrieval systems capable of storing multiple items in each storage unit. This facilitates transition from the ABC to the ICA policy, where desired.

5. Conclusion and future research

This paper uses historical information on customer orders that contains product affinity and turnover frequency to develop the ICA policy. This policy can be applied in environments where multiple products are stored on a single storage shelf, such as robotic mobile fulfilment systems, and where retrieval systems retrieve inventory shelves or bins and transport them to pick stations. The model formulated for the ICA policy can be solved using a standard commercial solver for small instances. The SA heuristic can solve the model for the dataset of a real warehouse. We compare the order retrieval time of the ICA policy with the commonly used ABC and FTB policies, while taking into account the space-sharing effect of the class-based storage (which typically leads to a smaller storage space requirement than the ICA policy). For the instances we tested, the ICA policy generally leads to shorter retrieval times, even for low and moderate values of product affinity. Retrieval time can be reduced by up to 40% and depends on customer order characteristics, especially the affinity and the skewness of the product turnover frequency curve. We also show that the ICA storage policy is generally insensitive to changes in the composition of the orders (while preserving the conditional average affinity and skewness of the ABC curve). The ICA policy performs better in a single command travel time model than with dual command cycles. Validation of the cluster allocation policy for a real warehouse shows that it can save on average 18% in retrieval time compared to the ABC policy. Future research should look at a dynamic storage assignment, in combination with a changing assortment. In addition, more efficient solution methods may be used to solve the mathematical model of the ICA policy, e.g. based on metaheuristics, to find better product allocations that reduce the optimality gap and solve larger instances. Additional analysis may examine the effect of batching or sequencing orders, which can increase the affinity in the orders and, consequently, the effectiveness of the ICA policy.

CRediT authorship contribution statement

Masoud Mirzaei: Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Writing - original draft, Visualization. **Nima Zaerpour:** Conceptualization, Writing - review & editing, Supervision. **René de Koster:** Conceptualization, Methodology, Resources, Writing - review & editing, Supervision.

Appendix A

See Table 6.

Table 6
List of frequently used acronyms.

ICA	Integrated Cluster Allocation
COI	Cube-per-Order Index
FTB	Full turnover-based
RMF	Robotic mobile fulfillment
AS/R	Automated storage and retrieval
SBS/R	Shuttle-based storage and retrieval
AVS/R	Autonomous vehicle storage and retrieval
SKU	Stock keeping unit
GA	Genetic algorithm
DBA	Data-based approach
DMSA	Data mining-based storage assignment
MCBH	Modified class-based heuristic
ASBH	Association seed-based heuristic
SA	Sequential alternating

Appendix B

Algorithm 2. (*Pseudocode for generating the order sets for the base example and scenarios.*)

```
1: input $O$, maximum order size $\theta$, $P$, correlated product subset $P_i$', affinity level
   indicator* $\xi$ and Pareto curve $\Omega$.
2: while the number of generated orders < $|O|$ do
3:    Generate the number of lines $\eta_o$ from discrete uniform distribution $U$ [1, $\theta$].
4:    Generate SKU $i \in P$ according to $\Omega$.
5:    Number of generated SKUs for current order $N_O$ =1.
6:    while $N_O < \eta_o$ do
7:        Generate a random number $R$ from $U$ [0, 1].
8:        if $R < \xi$ then
9:            Generate a SKU from the subset $P_i$',
10:       else generate a SKU from P according to $\Omega$.
11:       $N_O$= $N_O$+1.
12:   end while
13: end while
14: return generated orders.
```

**Affinity level indicator takes* 0, 0.2, 0.6, 0.8, *and* 0.9 *for zero, low, moderate, high, and very high affinity levels, respectively.*

Appendix C

Algorithm 3. (*Pseudocode for generating a new order set by changing the base example by Δ%, while affinity level is controlled.*)

```
1: input $O$, maximum order size $\theta$, $P$, affinity level indicator $\xi$ and Pareto curve $\Omega$,
   base example $\overline{O}$, rate of change $\Delta$.
2: Generated order subset $O'$ with $|O'| = \Delta.\left|\overline{O}\right|$ according to Algorithm 2.
3: Replace randomly $\Delta.|\overline{O}|$ orders from the base example by $O'$.
4: Calculate the conditional average affinity* of $\overline{O}$, $\overline{\xi}_{\overline{O}}$ and $O'$, $\overline{\xi}_{O'}$.
5: Calculate $\delta = |\overline{\xi}_{\overline{O}} - \overline{\xi}_{O'}|/\overline{\xi}_{\overline{O}}$.
6: while $\delta$ > 5% do
7:    Divide $O$' into 4 chunks.
8:    Calculate the $\overline{\xi}_c$ for each chunk.
9:    Eliminate the chunk with the largest $\delta_c$.
10:    Generate a new subset $O_c$'.
11:    Add $O_c$' to $O$'.
12:    Recalculate $\delta$.
13: end while
14: return updated orders.
```

**The conditional average affinities are calculated for* $1/|O| < \rho_{ij} < 0.1|O|$, *where* ρ_{ij} *is derived from Formula* 7, *because frequency of affinity levels beyond these limits are either too small or too large.*

References

Accorsi, R., Manzini, R., Bortolini, M., 2012. A hierarchical procedure for storage allocation and assignment within an order-picking system. A case study. Int. J. Logist. Res. Appl. 15 (6), 351–364. https://doi.org/10.1080/13675567.2012.742877.

Akro-Mils Multi-Load Storage Totes. Available at: https://akro-mils.com/Products/Types/Plastic-Storage-Containers/Multi-Load-Tote (accessed: 29 November 2019).

Amazon Unveils its Eighth Generation Fulfillment Center, Business Wire (2014). Available at: https://www.businesswire.com/news/home/20141130005031/en/Amazon-Unveils-Eighth-Generation-Fulfillment-Center (accessed: 29 January 2019).

Amirhosseini, M.M., Sharp, G.P., 1996. Simultaneous analysis of products and orders in storage assignment. Am. Soc. Mech. Eng. 803–812.

Ballou, R.H., 2004. Business Logistics/Supply Chain Management: Planning, Organizing, and Controlling the Supply Chain, fifth ed. Pearson Prentice Hall.

Bindi, F., et al., 2009. Similarity-based storage allocation rules in an order picking system: an application to the food service industry. Int. J. Logistics Res. Appl. 12 (4), 233–247. https://doi.org/10.1080/13675560903075943.

Bindi, F., Pareschi, A., Regattieri, A., 2007. Similarity coefficients and clustering techniques for the correlated assignment problem in warehousing systems. 19th International Conference on Production Research.

Borovinšek, M., et al., 2016. Multi-objective optimisation model of shuttle-based storage and retrieval system. Transport 32 (2), 120–137. https://doi.org/10.3846/16484142.2016.1186732.

Boysen, N., de Koster, R., Weidinger, F., 2019. Warehousing in the e-commerce era: A survey. Eur. J. Oper. Res. 277 (2), 396–411. https://doi.org/10.1016/j.ejor.2018.08.023.

Bozer, Y.A., White, J.A., 1984. Travel-time models for automated storage/retrieval systems. IIE Trans. 16 (4), 329–338. https://doi.org/10.1080/07408178408975252.

Chiang, D.-M.-H., Lin, C.-P., Chen, M.-C., 2011. The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. Enterprise Inform. Syst. 5 (2), 219–234. https://doi.org/10.1080/17517575.2010.537784.

Chiang, D.-M.-H., Lin, C.P., Chen, M.C., 2014. Data mining based storage assignment heuristics for travel distance reduction. Expert Syst. 31 (1), 81–90. https://doi.org/10.1111/exsy.12006.

Ekren, B.Y., et al., 2018. A tool for time, variance and energy related performance estimations in a shuttle-based storage and retrieval system. Appl. Math. Model. 63, 109–127. https://doi.org/10.1016/j.apm.2018.06.037.

Ekren, B.Y., Sari, Z., Lerher, T., 2015. Warehouse Design under Class-Based Storage Policy of Shuttle-Based Storage and Retrieval System. IFAC-PapersOnLine. 48 (3), 1152–1154. https://doi.org/10.1016/j.ifacol.2015.06.239.

Eynan, A., Rosenblatt, M.J., 1994. Establishing zones in single-command class-based rectangular AS/RS. IIE Trans. 26 (1), 38–46. https://doi.org/10.1080/07408179408966583.

Foroughi, A., et al., 2020. High-density storage with mobile racks: Picker routing and product location. J. Operat. Res. Soc. 1–19. https://doi.org/10.1080/01605682.2019.1700180.

Frazelle, E.H., 1989. Stock location assignment and order picking productivity. Georgia Institute of Technology. Available at: https://smartech.gatech.edu/handle/1853/29364?show=full (accessed: 22 February 2019).

Frazelle, E.H., 2016. World-Class Warehousing and Material Handling, second ed. McGraw-Hill.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman.

Garfinkel, M., 2005. Minimizing Multi-zone Orders in the Correlated Storage Assignment Problem. Systems Engineering. Georgia Institute of Technology. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.133.7668&rep=rep1&type=pdf (accessed: 7 March 2019).

Goetschalckx, M., Ashayeri, J., 1989. Classification and Design of Order Picking. Logistics World 2 (2), 99–106. https://doi.org/10.1108/eb007469.

Graves, S.C., Hausman, W.H., Schwarz, L.B., 1977. Storage-retrieval interleaving in automatic warehousing systems. Manage. Sci. 23 (9), 935–945. https://doi.org/10.1287/mnsc.23.9.935.

Hansen, P., Jaumard, B., 1997. Cluster analysis and mathematical programming. J. Am. Stat. Assoc. 79 (1–3), 191–215. https://doi.org/10.1007/BF02614317.

Hausman, W.H., Schwarz, L.B., Graves, S.C., 1976. Optimal storage assignment in automatic warehousing systems. Manage. Sci. 22 (6), 629–638. https://doi.org/10.1287/mnsc.22.6.629.

Heskett, J.L., 1963. Cube-per-order index - a key to warehouse stock location. Transp. Distrib. Manage. 3, 27–31.

Heskett, J.L., 1964. Putting the cube-per-order index to work in warehouse layout. Transp. Distrib. Manage. 4 (8), 23–30.

Jane, C.C., Laih, Y.W., 2005. A clustering algorithm for item assignment in a synchronized zone order picking system. Eur. J. Oper. Res. 166 (2), 489–496. https://doi.org/10.1016/j.ejor.2004.01.042.

Jewkes, E., Lee, C., Vickson, R., 2004. Product location, allocation and server home base location for an order picking line with multiple servers. Comput. Oper. Res. 31 (4), 623–636. https://doi.org/10.1016/S0305-0548(03)00035-2.

Kim, K.H., 1993. A joint determination of storage locations and space requirements for correlated items in a mini-load automated storage-retrieval system. Int. J. Prod. Res. 31 (11), 2649–2659. https://doi.org/10.1080/00207549308956888.

De Koster, R.B.M., Le-Duc, T., Yu, Y., 2008. Optimal storage rack design for a 3-dimensional compact AS/RS. Int. J. Prod. Res. 46 (6), 1495–1514. https://doi.org/10.1080/00207540600957795.

De Koster, R., Le-Duc, T., Roodbergen, K.J., 2007. Design and control of warehouse order picking: A literature review. Eur. J. Oper. Res. 182, 481–501. https://doi.org/10.1016/j.ejor.2006.07.009.

Kress, D., Boysen, N., Pesch, E., 2017. Which items should be stored together? A basic partition problem to assign storage space in group-based storage systems. IISE Trans. 49 (1), 13–30. https://doi.org/10.1080/0740817X.2016.1213469.

Küçükyaşar, M., Ekren, Y.B., Lerher, T., 2020. Cost and performance comparison for tier-captive and tier-to-tier SBS/RS warehouse configurations. Int. Trans. Operat. Res. itor.12864. https://doi.org/10.1111/itor.12864.

Lamballais Tessensohn, T., Roy, D., De Koster, R.B.M., 2020. Inventory allocation in robotic mobile fulfillment systems'. IISE Trans. 52 (1), 1–17. https://doi.org/10.1080/24725854.2018.1560517.

Lerher, T., 2018. Aisle changing shuttle carriers in autonomous vehicle storage and retrieval systems. Int. J. Prod. Res. 56 (11), 3859–3879. https://doi.org/10.1080/00207543.2018.1467060.

Lerher, T., Borovinsek, M., Ficko, M., 2017. Parametric study of throughput performance in SBS/RS based on simulation. Int. J. Simul. Model. 16 (1), 96–107. https://doi.org/10.2507/IJSIMM16(1)8.372.

Li, J., Moghaddam, M., Nof, S.Y., 2016. Dynamic storage assignment with product affinity and ABC classification—a case study. Int. J. Adv. Manuf. Technol. 84 (9–12), 2179–2194. https://doi.org/10.1007/s00170-015-7806-7.

Liu, C.M., 1999. Clustering techniques for stock location and order-picking in a distribution center. Comput. Oper. Res. 26 (10–11), 989–1002. https://doi.org/10.1016/S0305-0548(99)00026-X.

Marchet, G., et al., 2012. Analytical model to estimate performances of autonomous vehicle storage and retrieval systems for product totes. Int. J. Prod. Res. 50 (24), 7134–7148. https://doi.org/10.1080/00207543.2011.639815.

Marchet, G., et al., 2013. Development of a framework for the design of autonomous vehicle storage and retrieval systems. Int. J. Prod. Res. 51 (14), 4365–4387. https://doi.org/10.1080/00207543.2013.778430.

Rao, M.R., 1971. Cluster analysis and mathematical programming'. J. Am. Stat. Assoc. 66 (335), 622. https://doi.org/10.2307/2283542.

Roodbergen, K.J., Vis, I.F.A., 2009. A survey of literature on automated storage and retrieval systems. Eur. J. Oper. Res. 194 (2), 343–362. https://doi.org/10.1016/j.ejor.2008.01.038.

Rosenblatt, M.J., Eynan, A., 1989. Deriving the optimal boundaries for class-based automatic storage/retrieval systems. Manage. Sci. 35 (12), 1519–1524. https://doi.org/10.1287/mnsc.35.12.1519.

Roy, D., et al., 2019. Robot-storage zone assignment strategies in mobile fulfillment systems. Transp. Res. Part E: Logist. Transp. Rev. 122 (2019), 119–142. https://doi.org/10.1016/j.tre.2018.11.005.

Sadiq, M., Landers, T.L., Taylor, G.D., 1996. An assignment algorithm for dynamic picking systems. IIE Trans. 28 (8), 607–616. https://doi.org/10.1080/15458830.1996.11770706.

Sharp, G., Amirhosseini, M.M., Schwarz, F., 1998. New Approaches and Results for Product Storage Assignment: Consideration of Demand Variability, Demand Correlation, Storage Compartment Size, and Degree of Order Completion. Progress in Material Handling Research.

Silva, A., et al., 2020. Integrating storage location and order picking problems in warehouse planning. Transp. Res. Part E: Logistics Transp. Rev. 140, 102003 https://doi.org/10.1016/j.tre.2020.102003.

Wang, M., Zhang, R.-Q., Fan, K., 2020. Improving order-picking operation through efficient storage location assignment: a new approach. Comput. Indust. Eng. 139 (April 2019), 106186. https://doi.org/10.1016/j.cie.2019.106186.

Warehouse Science, 2017. Available at: www.warehouse-science.com (accessed: 10 November 2019).

Weidinger, F., Boysen, N., 2018. Scattered storage: how to distribute stock keeping units all around a mixed-shelves warehouse. Transp. Sci. 52 (6), 1412–1427. https://doi.org/10.1287/trsc.2017.0779.

Xiao, J., Zheng, L., 2010. A correlated storage location assignment problem in a single-block-multi- aisles warehouse considering BOM information. Int. J. Prod. Res. 48 (5), 1321–1338. https://doi.org/10.1080/00207540802555736.

Xiao, J., Zheng, L., 2012. Correlated storage assignment to minimize zone visits for BOM picking. Int. J. Adv. Manuf. Technol. 61 (5–8), 797–807. https://doi.org/10.1007/s00170-011-3740-5.

Yang, P., et al., 2017. Designing the optimal multi-deep AS/RS storage rack under full turnover-based storage policy based on non-approximate speed model of S/R machine. Transp. Res. Part E: Logist. Transp. Rev. 104, 113–130. https://doi.org/10.1016/j.tre.2017.05.010.

Yang, P., Zhao, Z., Guo, H., 2020. Order batch picking optimization under different storage scenarios for e-commerce warehouses. Transp. Res. Part E: Logist. Transp. Rev. 136(June 2019), 101897. https://doi.org/10.1016/j.tre.2020.101897.

Yu, Y., de Koster, R.B.M., Guo, X., 2015. Class-Based storage with a finite number of items: using more classes is not always better. Prod. Operat. Manage. 24 (8), 1235–1247. https://doi.org/10.1111/poms.12334.

Yu, Y., de Koster, R., 2009. Optimal zone boundaries for two-class-based compact three-dimensional automated storage and retrieval systems. IIE Trans. 41 (3), 194–208. https://doi.org/10.1080/07408170802375778.

Zhang, J., Onal, S., Das, S., 2020. The dynamic stocking location problem – Dispersing inventory in fulfillment warehouses with explosive storage. International Journal of Production Economics. 224(February 2019), 107550. https://doi.org/10.1016/j.ijpe.2019.107550.

Zhang, R.-Q., Wang, M., Pan, X., 2019. New model of the storage location assignment problem considering demand correlation pattern. Comput. Indust. Eng. 129 (December 2018), 210–219. https://doi.org/10.1016/j.cie.2019.01.027.

Zhang, Y., 2016. Correlated storage assignment strategy to reduce travel distance in order picking. IFAC-PapersOnLine 49 (2), 30–35. https://doi.org/10.1016/j.ifacol.2016.03.006.