MASOUD MIRZAEI

Advanced Storage and Retrieval Policies in Automated Warehouses

ADVANCED STORAGE AND RETRIEVAL POLICIES IN AUTOMATED WAREHOUSES

Advanced Storage and Retrieval Policies in Automated Warehouses

Geavanceerde methoden voor in- en uitslag in automatische magazijnen

Thesis

to obtain the degree of Doctor from the
Erasmus University Rotterdam
by command of the
rector magnificus

Prof.dr. R.C.M.E. Engels

and in accordance with the decision of the Doctorate Board.

The public defense shall be held on
Thursday 16 April 2020 at 11:30 hours

by

MASOUD MIRZAEI

born in Arak, Iran

Erasmus University Rotterdam

Doctoral Committee

Doctoral dissertation supervisor:	Prof.dr. M.B.M. de Koster
Other members:	Prof.dr. R.A. Zuidwijk Prof.dr.ir. I.J.B.F. Adan Dr. M.E. Schmidt
Co-supervisor:	Dr. N. Zaerpour

Erasmus Research Institute of Management – ERIM
The joint research institute of the Rotterdam School of Management (RSM)
and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam
Internet: www.erim.eur.nl

ERIM Electronic Series Portal: repub.eur.nl/

ERIM PhD Series in Research in Management, 490
ERIM reference number: EPS-2020-490-LIS
ISBN 978-90-5892-582-4

Design: PanArt, www.panart.nl

This publication (cover and interior) is printed by Tuijtel on recycled paper, BalanceSilk®
The ink used is produced from renewable resources and alcohol free fountain solution.
Certifications for the paper and the printing production process: Recycle, EU Ecolabel, FSC®C007225
More info: www.tuijtel.com

Acknowledgement

Any important project involves ample professional and social support of several individuals, and a PhD thesis is no exception. First and foremost, I would like to express my gratitude to René de Koster who was always available for discussing new ideas and helped me grow as an academic. He is not only an expert in warehousing, but also a great companion for skiing. Thank you, René, for all the support during these years. I would like to thank Yugang Yu who initially helped me with the proposal and hosted me at University of Science and Technology of China in Hefei. I am thankful to Nima Zaerpour who agreed to become my co-supervisor halfway and greatly helped me during early morning skype calls. I am grateful for the constructive feedback and comments from my inner committee Marie Schmidt, Ivo Adan and Rob Zuidwijk.

During the whole trajectory, and before it, my family has always been there for me. I thank Narges, who agreed to leave many things behind in Iran and move to the Netherlands with me. She was a tremendous support for me and our two beautiful daughters, Nazgol and Lily Rose. I cannot be grateful enough to my parents, who have always encouraged me to follow through on my dreams.

I am also thankful to all the professional and friendly colleagues at the department. The lunch and coffee breaks were pleasant and inspirational. My office mates, Tim, Alberto and Johann were always open, not only to have fun activities, but also to intense research discussions.

Masoud Mirzaei

Eindhoven, 2020

Contents

Chapter 1

Introduction

Warehouses play a critical role in supply chains. They are responsible for storing products and distributing them to customers. The performance of a warehouse depends on the storage and retrieval systems and their control methods. With the advent of new automated and robotic technologies, new storage and retrieval methods have emerged, which can help the warehouse to become more efficient and responsive. This thesis aims to develop new storage and retrieval policies and methods that benefit from such automated technologies. Section 1.1 reviews important trends that influence warehouse operations. Section 1.2 discusses different choices in storage assignment. Section 1.3 explains different types of automated storage systems that are studied in this thesis. Section 1.4 gives an overview of research questions, corresponding methodologies and contribution of different chapters.

1.1. Current Trends in Warehousing

Over the last decades, market competition has put pressure on companies to reduce their customer order delivery time. In retail, single-day delivery has become a standard service. The so-called "Retail Apocalypse" in the U.S. (Washingtonpost.com, 2019) shows that customers are no longer satisfied with the limited assortment and services in brick-and-mortar stores and demand shifts to online shopping. In the Netherlands, 11% of the shops are closed since 2010 due to the shift towards online shopping (CBS, 2019). According to Brynjolfsson *et al.* (2011), the sales of retailers such as Amazon may no longer follow the Pareto principle, but rather exhibit a "long tail" curve. The long-tail theory, popularized by Anderson (2004, 2008), refers to the fact that customers demand a wider variety of products and tend to buy more niche products rather than the popular ones. Additionally, land shortage stimulates development of compact storage systems that increase space efficiency. However, this is often at the expense of lower throughput capacity.

These developments, on one hand, result in distribution centers that grow in size, which leads to long order picking travel times. On the other hand, however, they should offer short delivery times. Therefore, distribution centers require operations that allow reduced throughput times. Since the labor cost is increasing while the technology is becoming more affordable than before, using new technologies and robotics may be a viable option. According to Tompkins *et al.* (2010), more than 50% of the order picking time in manual warehouses is spent on traveling to the inventory locations and retrieving requested quantities. Developing advanced storage and retrieval policies can reduce order retrieval time. The next section gives an overview of storage assignment policies used in practice and studied in the literature.

1.2. Choices in Storage Assignment

Products can be assigned to storage locations in various ways. A storage assignment policy decides to which location a product should be assigned and in what quantity. It thereby determines the distance of a product to the order drop-off point and hence impacts picking time. Extensive literature has studied the role of storage assignment policies in warehouses. Here, we review policies commonly used in practice. In a random storage policy, products are stored randomly in available storage locations in the warehouse. For convenience, the closest available location may be used which results in random storage after a longer period of use (Malmborg, 1998). The random policy is easy to implement and is widely used in practice. It requires a smaller storage space compared to other policies because each storage location can be used by the inventory of any product (Malmborg, 1996). However, it does not result in a fast retrieval process since customer demand is not random and often follows a pattern. Many papers study the random assignment, and it is often used as a benchmark to other policies (see De Koster *et al.*, 2007; Roodbergen and Vis, 2009; Onal *et al.*, 2017).

Turnover-based policies rely on historical data of customer demand to identify frequently requested products. Products may be ranked based on several criteria such as units picked, lines picked, or the cube-per-order index (COI, see Heskett, 1963), which relates product turnover to the number of loads stored in the system.

The products are then assigned to a dedicated storage zone close to order delivery points, based on their rank. Products may be grouped first in storage zones, and then, stored randomly in storage locations within each storage zone. The turnover-based assignment using such zones is called a class-based policy (also known as ABC class-based storage, referring to classes A, B, and C, although more than three classes may be used). If the number of zones equals the number of products, the storage policy is called full turnover-based storage. The decision on the number of zones is a design choice for the warehouse. Yu *et al.* (2015) suggest that few zones, usually two or three, result in the minimum picking travel time. Compared to random assignment, turnover-based assignments result in shorter order picking retrieval time for popular products.

Another assignment choice is to disperse a product over the storage locations, i.e. splitting the inventory of a product and spreading it over the storage system. Specifically, when orders contain more than one product, a dispersed approach can help to find the requested products at closer proximity compared to when each product is assigned to only one location (Weidinger and Boysen, 2018). However, this may require a higher replenishment effort when a received product must be spread over multiple locations each time. Another dispersion approach is to replenish a product to only one location, but different from the current inventory locations. This approach results in less dispersion compared to the other dispersion method but requires less replenishment effort. The main difference between random and dispersed assignments is that the inventory of one product is assigned to the required number of storage locations in random storage while it can be assigned to any desired number of storage locations in dispersed storage. Literature studying the impact of dispersed assignment policies is not yet abundant.

Correlated assignment is yet another different storage policy. It uses the information of product correlations in historical customer demand data on the frequency of joint requests for products in multiple line orders. This relative historical frequency is called product correlation. A correlated storage policy assigns highly correlated products in close proximity in order to reduce the total retrieval time in order picking. The correlated assignment is relatively new in the literature and is also known as similarity-based assignment (Bindi *et al.*, 2009), cluster-based assignment (Jane and Laih, 2005) and affinity-based assignment (Li *et al.*, 2016).

Table 1 provides an overview of different storage assignment policies. The second column of the table shows the main decision factor for each policy. The third column shows some typical examples that study systems using such policies and their impact on retrieval time performance.

Table 1. Storage assignment policies.

Policy	**Decision Factor**	**Papers**
Random	Random	De Koster *et al.* (2008), Fukunari and Malmborg (2008)
Class-based	Product Turnover Frequency	Yu *et al.* (2015), Yu and de Koster (2009)
Dispersed	Random / Product Correlation	Weidinger and Boysen (2018), Onal *et al.*, (2017, 2018)
Correlated	Product Correlation	Garfinkel (2005), Chiang *et al.* (2011, 2014), Li *et al.* (2016)

1.3. Automated Storage Systems

Recently, new warehouse automation and robotic systems have been introduced to store and handle individual products and product loads. This thesis focuses on three types of automated systems, puzzle-based storage (PBS), automated storage and retrieval (AS/R), and autonomous robotic mobile fulfillment (RMF) systems.

A puzzle-based storage (PBS) system is a very compact storage system that stores loads on shuttles on the storage locations. Figure 1.1(a) shows a PBS system used in an automated parking garage where each car is stored on a mobile shuttle. The driver leaves the car at the entrance, and a shuttle will move the car inside the system and store it on an empty location. One or more lifts transport the cars in vertical direction, between the storage tiers. This type of system uses the available space very efficiently, as no transport aisles are needed for moving cars. It is used in areas where space is expensive, like city centers and airports. Figure 1.1(b) shows a mini-load AS/R system where loads are stored in bins which are stored in high-rise shelves. AS/R systems are compact systems where cranes can move within narrow aisles to access the bins at different levels and bring them to pick stations. At a pick station, the requested quantity of a product is picked, after which a crane returns the bin to a storage location. Figure 1.1(c) shows an RMF system where products are stored on multi-level pods. Available space on each pod is divided into compartments that allow storing multiple products on each pod. Autonomous robots pick up the entire pod from the storage area and move it to a pick station, where customer orders are picked. After picking, robots return the pod to the storage area. Robots can travel underneath the pods when empty. RMF systems save labor costs compared to manual order picking systems and are easy to expand by adding more pods and robots to increase the storage capacity and throughput capacity, respectively. They have been adopted by big players in ecommerce such as Amazon and Alibaba.

(a) (b) (c)

Figure 1.1. (a) An automated compact parking lot (source: Japan Parking System Manufacturers Association Incorporated, 2019) (b) a mini-load AS/R system (source: Ferretto Group, 2019) (c) Autonomous robots carrying storage pods (source: Reuters.com, 2019).

1.4. Research Questions and Outline of the Dissertation

Warehouses and distribution centers are increasingly adopting automated systems. Employing an efficient storage and retrieval policy is important to achieve short order throughput times and a high system throughput capacity. This thesis, therefore, focuses on developing advanced storage and retrieval policies that particularly support such automated systems. In the following sections, the outline of the dissertation, research questions and methodology of each chapter are discussed.

Chapter 2. Modeling Load Retrievals in Puzzle-based Storage Systems

Space is at a premium in many locations, such as densely populated areas. Compact storage systems are designed to achieve high space utilization. Puzzle-based storage (PBS) systems are very compact storage systems, without transport aisles. However, a major drawback of these systems is a long retrieval time due to lack of transportation space. Chapter 2 studies retrieval time models in PBS systems with maximum space utilization. The system consists of loads stored on shuttles that are placed right next to each other, i.e. no access aisles for a robot or picker are available. There is only one open location available in the system that allows the reshuffling of the storage locations. This configuration resembles the well-known 15-tile puzzle. Retrieval in such systems has been modeled for a single load by Gue and Kim (2007), but in this chapter, the problem of retrieving multiple loads is addressed. The following research question is answered.

What is the optimal retrieval method in PBS systems (i.e. minimizing the number of required moves) to retrieve multiple requested loads, using one open location?

A finite algorithm is developed for the optimal joint retrieval of two loads by joining them at an intermediary location first and moving them together afterward. Closed-form expressions are derived for joint retrieval of two adjacent loads (see Theorem 1). This chapter proves this is the optimal path. For the case of two requested loads, the position of the optimal intermediary locations is determined. For multiple loads, close-to-optimal joining locations are determined. Based on this, an efficient heuristic is developed for the retrieval of multiple loads simultaneously. The results show that large savings can be achieved using multiple-load retrievals compared to sequential single-load retrievals in these systems.

Chapter 3. The Impact of Integrated Cluster-based Storage Assignment in Automated Warehouses

Historical demand data can provide rich information on the customer demand profile. Various storage assignment policies, such as class-based and full turnover-based storage take advantage of historical order information. However, they mainly look at the turnover frequency of demand for a certain product and use it to identify popular products. Few studies look at the information on the correlation between products in customer demand. Correlated assignment models in the literature (such as those proposed by Garfinkel, 2005, Xiao and Zheng 2010, 2012 and Chiang *et al.*, 2014) mainly take a sequential approach to find the clusters of correlated products and then assign them to storage areas/zones. These methods are suboptimal, as they first maximize the correlation of products in the clusters and then assign the clusters to storage areas. Additionally, because clusters of products are assigned to storage zones or aisles, these models are only applicable to manual order picking where a picker runs a picking tour of several lines. In robotic warehouses, a robot (or shuttle) visits a storage location and picks up the entire storage pod/bin which carries some of the requested products. Here, grouping correlated products is beneficial only if products in a group are stored on the same storage pod, not in the same storage zone. This chapter answers the following research question.

How does integrated clustering and storage assignment of correlated products affect the order picking performance in automated warehouses?

An integer linear program is developed that models the optimal integrated clustering and storage assignment of the products to minimize the total retrieval time. The model is solved with a general optimization solver and tested for multiple levels of correlation, turnover frequency and order size. The performance of the model is evaluated for both mini-load AS/R and for RMF systems (see Section 1.3), where each cluster of products is assigned to a storage bin or storage pod consisting of multiple compartments sufficient to house the number of products in the cluster. A comparison of the proposed integrated model with the sequential correlated assignment and turnover-based assignment shows that integrated assignment can yield considerable benefits when the correlation of the products is high, and the product turnover frequency curve is not highly skewed.

Chapter 4. Correlated Dispersed Storage Assignment in Robotic Warehouses

This chapter builds on chapter 3, by not only looking at correlated storage assignment but also combining it with product dispersion. This combination of storage policies is applied to RMF systems (see Section 1.3). Chapter 4 investigates the effect of this combined policy on the expected retrieval time and compares it with random, class-based, correlated (but not dispersed), and dispersed (but not correlated) assignment policies. The following research questions are studied.

What is the effect of product dispersion and storage clustering on the expected order picking retrieval time in RMF systems? How do product correlation and product turnover frequency contribute to the performance of the policies?

First, a mixed-integer linear program is developed for optimal product to cluster and cluster to zone allocation to minimize the expected retrieval time to a closest pick station. Note that, if orders contain a large number of lines, downstream order consolidation may be needed before the order can be shipped. Such possible consolidation time is not included in the analysis. The retrieval time expressions are developed for different zone configurations and positions of pick stations. Solvers such as Gurobi 9.0 are able to solve small instances of the model. An efficient heuristic method is proposed to enable solving real size instances of the problem. Particularly, a thorough analysis of the impact of turnover frequency and correlation of the products on the performance of different storage policies is conducted using a dataset of the warehouse of a wholesaler in personal care products. The analytical results show that, for this warehouse, the correlated dispersed assignment leads to a significantly shorter expected retrieval time compared to the benchmarks. Furthermore, the correlation in customer demand plays a major role in the performance of the models while the turnover frequency showed a minor influence in the cases we tested.

Chapter 5. Summary and Conclusion

Chapter 5 gives an overview of the results of previous chapters. This summary highlights the contribution of this dissertation by revisiting the main research questions and findings. The limitations of the conducted research are also discussed. An outlook of further research on storage and retrieval policies applicable to automated warehouses is also presented.

Research Statement

All the chapters of this dissertation are written by the author. The author is responsible for the research questions, methodology and analytical results of each chapter. The models and results have been validated using simulation, benchmarks, and numerical analysis. The promoters had a great impact on the quality of the chapters by providing continuous critical feedback during my Ph.D. program. Feedbacks from the doctoral committee helped to improve the quality of the dissertation. Chapter 2 is published as Mirzaei *et al.* (2017) which benefitted from the constructive comments from reviewers. Chapter 3 is submitted to a journal for peer review and Chapter 4 will be submitted soon.

Chapter 2

Modeling Load Retrievals in Puzzle-based Storage Systems

2.1. Introduction

Warehouses are important nodes in the supply chain as they allow to match supply with customer demand and to achieve economies of scale in transport. Warehouses are labor-intensive and consume much space. Bartholdi and Hackman (2016) state that the fundamental idea of warehouse management lies in two resources: space and labor. While labor is usually available in urban areas, land is expensive. Space efficient storage systems offer a solution to this problem.

A conventional storage system consists of racks and aisles. Aisles are used for transporting goods to and from the storage racks. They take up space which could alternatively be used efficiently for storing loads. If space is not used efficiently, larger distances may have to be traveled to transport loads, requiring more resources. According to Tompkins et al. (1996), Roodbergen and De Koster (2001) and De Koster et al. (2007), non-value adding travel forms the majority of an order picker's time in such conventional storage systems. In the 60s, Automated Storage/ Retrieval (AS/R) systems were introduced. These systems can store a large number of unit loads on a limited footprint. These systems have received much attention from researchers focusing on, e.g., travel time models and system size optimization (see e.g. Bozer and White, 1984 and Lee, 1997). In order to use the space even more efficiently, very high density storage (or puzzle-based) systems were introduced by storing the loads multi-deep (De Koster et al., 2008).

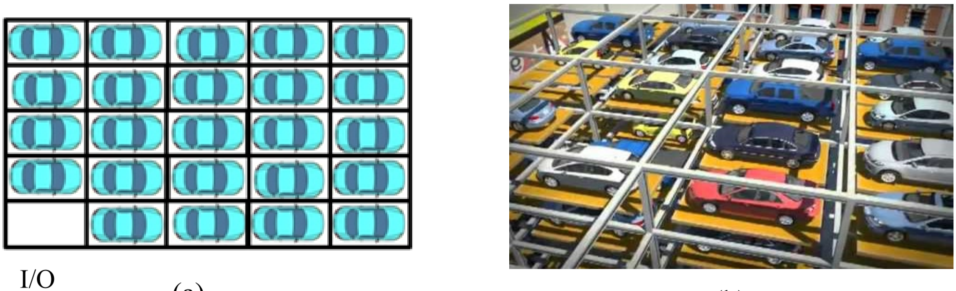
Recently, so called "puzzle-based" storage systems have been introduced. The term Puzzle-based Storage (PBS) system comes from Gue and Kim (2007). PBS systems are very compact storage systems which are fully automated. Unit-loads are stored dense, without even a single aisle, yet each unit load can be retrieved independently. Applications of PBS systems can be found in warehouses and distribution centers (DCs), automated car parking systems, and container terminals (Zaerpour, Yu and De Koster 2015).

2.1.1. Description of the PBS System

The main components of a PBS system are: (1) shuttles that can move in horizontal x- and y- directions, carrying the unit loads, (2) a depot (I/O point), (3) a lift for vertical transportation in case of a multilevel system, and (4) one or more empty locations which provide sufficient maneuvering space for shuttles to move. Such an open location is also called an "escort" because of its role of escorting the load to the destination. To bring a requested unit load to the I/O point, other shuttles have to move to first bring an escort next to the requested load. Then, the load is escorted to the I/O point by the escort.

A PBS system with one escort is comparable to the well-known 15-tile puzzle game. This game consists of 15 numbered tiles which are randomly distributed in a 4×4 square with one missing tile. The mission is to sort numbers by shuffling the tiles. In the same fashion, N^2-1 unit loads can be stored at each level in an $N \times N$ PBS system. This results in very high space usage efficiency.

Figure 1(a) shows the top view of a typical PBS system. Cars in the picture represent loads stored in the system and the white cell represents the escort. The escort is initially located at the lower left corner, next to the I/O point. Figure 1(b) shows a PBS car parking system. Each unit load (a car) is stored on its own shuttle which can move in both horizontal directions. When an order for retrieving a load is released, the escort moves towards the requested load. This means all shuttles on the path have to move in the opposite direction. Once the escort reaches a position next to the load (down or left), depending on the load's location, the shuttle which contains the load will move to the empty location. Then the escort will end up at the right or top of the requested load. It again needs to move to either down or left of the requested load to provide space for it to move. This repeats until the load arrives at the I/O point. In this way, any load can be accessed individually with no more than one empty space unit in the storage area. Although PBS systems are extremely space efficient solution, they are not fast. Therefore it is of the utmost importance to furnish this solution with faster methods. The multiple load retrieval method proposed in this chapter addresses this drawback and makes PBS systems more responsive.



(a) (b)

Figure 1. (a) A top view of a PBS, (b) A 3D view of a PBS car parking system (source Eweco, 2013).

2.1.2. Literature Review

Literature on unit-load compact storage systems is not abundant. In practice, most compact storage still have at least one aisle. A crane, or S/R machine, operates in the aisle and retrieves unit loads using a satellite connected to the crane. Sari et al. (2005) study a flow rack compact storage system where the pallets are stored and retrieved at different rack sides by two S/R machines responsible for storage and retrieval respectively. De Koster et al. (2008) and Yu and De Koster (2009, 2012) study a compact crane-based storage system with built-in multi-deep circular conveyors. The system is fully automated, and every pallet stored is accessible individually by rotating conveyors. Zaerpour et al. (2015) derive the optimal storage allocation for a crane-based compact storage system, operating in a cross-dock when all destinations of incoming loads are known.

Carousel systems constitute another category of compact automated storage and retrieval systems in which accessing an item requires moving other items. These systems consist of a number of linked drawers carrying small and medium-sized products that rotate in a closed loop. Litvak (2006) proposes optimal picking of large orders based on the shortest rotation time and she studies the number of items collected before a turn. Hwang et al. (1999) study standard and double carousel systems and analytically measure the effect of double shuttles on throughput.

Studies on PBS systems are new. Gue and Kim (2007) appear to be the first researchers to study PBS systems. They study a PBS system where unit loads are retrieved one by one using single or multiple empty locations. They compare storage density and retrieval time of puzzle-based systems with traditional low-density aisle-based warehouses. While traditional warehouses usually perform better than puzzle systems in terms of retrieval time, they have lower space efficiency. Kota, Taylor and Gue (2015) analytically derive the single-load retrieval time expression when multiple escorts are randomly placed within the system. They extend the expression to a system with two escorts and formulate an integer program for the general case with multiple escorts. Alfieri et al. (2012) propose heuristics for using a limited set of shared shuttles to transport unit loads in puzzle-based systems. They consider multiple I/O points, partition the storage area, and then assign shuttles to partitions based on expected workload. Shuttles move parallel where possible. Gue et al. (2013) propose a decentralized control for a deadlock-free puzzle system named GridStore. Loads arrive at one side of the system, can move individually within the system, and leave at the opposite side of the system. Each unit load communicates with neighboring locations to decide its route. Zaerpour, Yu and De Koster (2015) study the optimal configuration of a multi-level PBS system (they call them live-cube systems) by assuming sufficient empty locations exist at each level to create virtual aisles and multiple loads can move simultaneously. When a virtual aisle has been created, determining the retrieval time is similar to a traditional, aisle-based, warehouse. The minimum number of empty locations to create a virtual aisle at a given storage level equals the maximum of the rows and columns of the system. Yu et al. (2016) propose a method to find the optimal retrieval path for a requested load, with multiple open locations and with so-called block load movement. All these studies assume the loads all have the same size. Flake and Baum (2002) and Hearn and Demaine (2005) study the rush hour problem in a PBS car parking system, with the objective to store as many cars of different sizes in a compact storage.

While previous studies have focused on single load retrieval, in practice, information of multiple retrieval requests is usually available. Hence, multiple loads may be retrieved simultaneously, improving the performance of PBS systems significantly. In this chapter, we study multiple load retrieval in a PBS system. We answer questions like how and in which sequence loads should be retrieved in order to minimize total retrieval time. This question has not yet been addressed in literature. We develop an optimal method for this problem based on joint load retrieval. The results show that by using joint retrieval, the total retrieval time can be reduced significantly compared to individual retrieval. We first present a retrieval method for two loads that finds the minimum number of retrieval moves. Then, we extend this optimal method for jointly retrieving three loads and afterward, generalize it to retrieve multiple loads using approximate analysis. Table 1 summarizes the literature on PBS systems and highlights the contribution of this chapter. The second column shows whether an optimal solution is provided or a heuristic. Column three defines the number of open locations assumed in the system. Column four shows whether there is a single move at a time or multiple loads can move simultaneously. The last column defines the number of loads that the system can retrieve together.

Table 1. Comparing the papers on PBS systems.

Paper	**Optimal/ Heuristic**	**Number of escorts**	**Simultaneous load moves**	**Single / Multiple Load**
Gue and Kim (2007)	Optimal	One, many	No	Single
Kota et al. (2015)	Optimal & Heuristic	Many (randomly)	No	Single
Alfieri et al. (2012)	Heuristic	Many	Yes	Single
Zaerpour et al. (2015)	Heuristic	Many	Yes	Single
Gue et al. (2014)	Heuristic	Many	Yes	Single
Yu. et al. (2016)	Optimal	One, many	Yes	Single
This chapter	Optimal & Heuristic	One	No	Multiple

The remainder of the chapter is organized as follows. Section 2.2 describes an optimal retrieval method for two arbitrary loads in the system. Section 2.3 extends the dual-load retrieval method to three or more loads. Section 2.4 compares the results with single- load retrieval. In the last section, conclusions are drawn.

2.2. An optimal Dual-load Retrieval Method

In case two loads need retrieval, it is possible to reduce travel time, as compared with individual retrieval, by retrieving them jointly. We propose a dual-load retrieval method for this and demonstrate optimality by enumeration. Three methods are distinguished for retrieving two loads: (1) moving loads individually towards the I/O point using the algorithm of Gue and Kim (2007), (2) moving loads A and B by alternating between them, requiring the escort to move back and forth between the loads, and (3) bringing both loads to a given joint location and then moving them together. Obviously, the method (2) is not optimal, due to unnecessary extra moves of the escort traveling between the loads. We prove method (3) leads to an optimal solution, for a joint position where the loads are adjacent.

We make the following assumptions for the system:

(1) All loads are stored on shuttles, which can move in both horizontal directions. This assumption is valid for particular types of PBS systems.

(2) The storage system has N rows and N columns. This can be extended to non-square systems.

(3) The I/O point is located at the lower left corner.

(4) There is only one escort, which is initially located at position (1, 1), next to the I/O point. Usually, escort will be found here, after each retrieval.

(5) Only one load moves at a time, even when multiple loads need to move in the same direction.

(6) We distinguish only retrievals on a single storage level. For multiple levels, a lift fulfills the vertical transportations.

We first define several concepts to ease the exposition.

Definition 1 (Joining location for two loads in the PBS grid): A location where the two requested loads become adjacent for the first time in their retrieval path. The joining location is defined as the location of one of these two adjacent loads, namely the one which is the closest to the I/O point.

Definition 2 (Dual load move): Moving two loads consecutively on the same retrieval path with no other loads between them.

Definition 3 (Optimal joining location): A joining location for two loads, which leads to the minimum total number of retrieval moves.

Now we can formulate the following lemma.

Lemma 1: Dual-load retrieval from an optimal joining location, always performs better than or equal to two single-load retrievals, in terms of the total number of moves.

Proof: Setting the joining location at (1, 1), immediately transforms the dual-load retrieval problem into two single load retrievals. Indeed, a better joining location saves moves.

■

As Lemma 1 shows, retrieving two loads using an optimal joining location always results in a total number of moves less than or equal to the number of moves of two individual single load retrievals. Gue and Kim (2007) propose an optimal method for single load retrieval, where each load first moves by several so-called 3-moves, followed by so-called 5-moves when the load reaches the side of the system. In the 3-moves, the empty location moves from a location behind (above) the load one step down and one step left to reach a location below (in front of) the load. Now, it makes space available for the load. to approach the I/O point with one more step. It takes the empty location 4 moves to move around a load at the bottom or left side of the system.

Therefore, in the optimal dual-load retrieval method, the loads are first brought together at an optimal joining location, using optimal single load moves, and are then moved toward the I/O point jointly by optimal dual-load moves. Figure 2 gives a flow diagram of the dual-load retrieval method. This procedure can be explained in four steps. Algorithm 1 illustrates the steps in this method. Optimality of the joining location is ensured by enumerating all possibilities with a complexity of $O(N^2)$. Optimality of the adjacent location of the joining location, to which the second load will be brought, is ensured again by enumeration in step 2. Optimality of the single load moves is ensured by the method of Gue and Kim (2007). Moving the loads jointly in an optimal fashion in step 3 is explained in theorem 1. The two requested loads are (i_1, j_1) and (i_2, j_2). The load closest to the I/O point is labeled as the first load and the other load is labeled as the second load. In the case of equal distances, they can be labeled randomly.

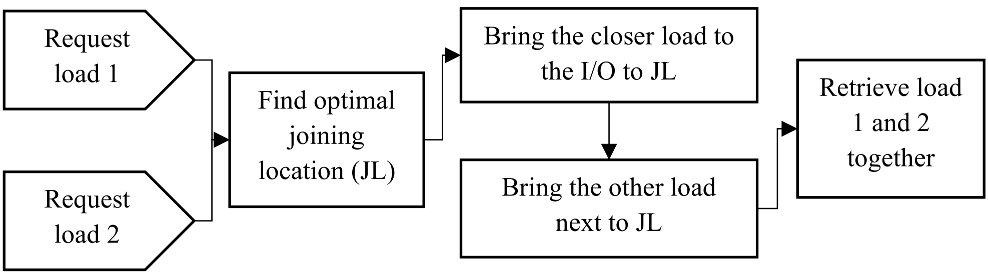


Figure 2. Flow diagram of dual-load retrieval.

Algorithm 1: Optimal dual-load retrieval method

Step 1

for $i = 1: max\ \{i_1, i_2\}$

    for $j = 1: max\ \{j_1, j_2\}$

        let (i,j) be the joining location.

        calculate the minimum number of moves, $M_1(i,j)$, needed to bring the first load to (i,j).

Step 2

        for $k = 1: 4$

calculate the minimum number of moves, $M_0(k)$, needed to bring the second load to one of the 4 adjacent locations of (i,j).
end for
pick the location $k := Argmin\{M_0(k) | \ k = 1,2,3,4\}$.
$M_2(i,j) := M_0(k)$.
Step 3
calculate the minimum number of moves, $M_3(i,j)$, needed to bring the loads jointly to the I/O point.
$M(i,j) := M_1(i,j) + M_2(i,j) + M_3(i,j)$.
end for
end for
Step 4
pick the solution with optimal joining location (i,j) which minimizes $M(i,j)$.

Algorithm 1 determines the optimum joining location, by enumerating all possible locations and comparing the results. In a system of size $N \times N$ there are N^2 possible joining locations. But, in practice, certain areas can be excluded from enumeration, depending on the position of the loads. We show in lemma 2 the number of locations that need to be enumerated is actually less than N^2. This accelerates the process of finding the joining location. Figure 4(a) shows two requested loads in the system. A joining location is marked by a 'plus' sign. The dashed lines show the boundary of locations to be enumerated.

Lemma 2: The Manhattan distance to the I/O point of an optimal joining location, is less than or equal to the Manhattan distance of the requested loads (i_1,j_1) and (i_2,j_2) to the I/O point.

Proof: a) Assume a location $L = (i_0, j_0)$ with a Manhattan distance larger than that of at least one of the loads, is nominated as the optimal joining location. See figure 3. This means $i_0 + j_0 > i_1 + j_1$ or $i_0 + j_0 > i_2 + j_2$. Without loss of generality, we assume $i_0 + j_0 > i_1 + j_1$ and $i_0 + j_0 < i_2 + j_2$. We show that joining location (i_1, j_1) outperforms L in the required number of moves to retrieve the loads. The number of moves to bring the two loads to L equals to the number of moves needed to bring the load (i_2, j_2) to L plus the number of moves to further move it next to (i_1, j_1). However, this, at most, equals to the minimum number of moves to bring (i_2, j_2) directly to (i_1, j_1). Furthermore, we know that, by definition, (i_1, j_1) is closer to the I/O point than L. Therefore, less joint moves is required from there to the I/O point. Thus, L cannot be located farther than any of the two loads.

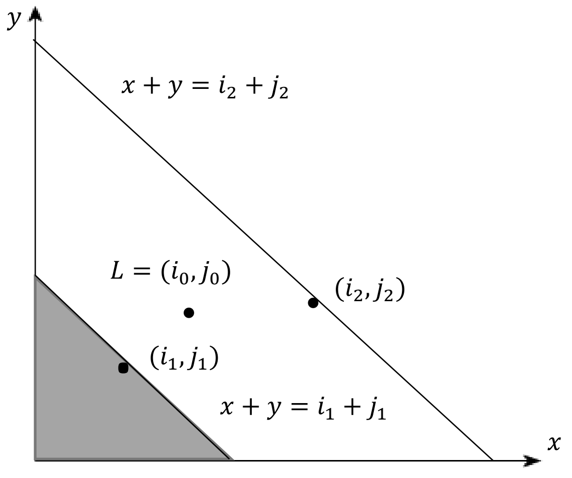


Figure 3. The search area for the optimal joining location.

■

In a number of situations, the joining location can be predefined, and no enumeration is needed. Table 2 provides a list of such conditions. In condition 1, when both requested loads are located at the left side of the system (i.e. i_1, i_2=1), the escort goes directly to the farther load and brings the closer load to $(1, j_1\text{-}1)$ on its path. This is the joining location for the loads. The same procedure can be applied for condition 2. In condition 3, where both loads are on the diagonal, again the optimal path for the escort is to directly go to the farther load and bring it next to the closer one. This involves moving the closer load to either $(i_1, j_1\text{-}1)$ or $(i_1\text{-}1, j_1)$. In the worst case scenario of condition 4, one load is located anywhere at the left side of the system and the other one is anywhere at the bottom side. Then the joining location is (1,1) which basically means two individual retrievals.

Table 2. Conditions that lead to a predefined optimal joining location.

Nr.	**Condition**	**Predefined optimal joining location**
1	i_1, i_2=1 and j_1< j_2	$(1, j_1\text{-}1)$
2	j_1, j_2=1 and i_1< i_2	$(i_1\text{-}1,1)$
3	i_1= j_1 and i_2= j_2	$(i_1, j_1\text{-}1)$ and $(i_1\text{-}1, j_1)$
4	$(1,j_1)$ and $(i_2,1)$	(1,1)

In the first step of algorithm 1, we need to know the number of steps to bring the first load to the joining location via the shortest path for each possible joining location. Figure 4(b) shows this transfer. This can be done by the single-load retrieval method of Gue and Kim (2007); the only difference is the I/O point as the destination has been replaced by the joining location.

The second step brings the second load next to the first one. It selects the best locations adjacent to the joining location such that the total number of moves is minimized. It is determined by enumeration and comparing the results for each adjacent location. Figure 4(c) shows how the second load joins the first one. The enumerated locations are marked by ‘×’. Note that bringing the second load to some of these adjacent locations might alter the location of the first load. In Figure 4(c) for example, moving the second load to the adjacent location on the left side of the first load, moves the first load one step down. We ignore this because such candidate joining locations lead to higher total retrieval moves (extra moves to make them adjacent again) and are not chosen as the optimal joining location. At the end of this phase, as shown in Figure 4(d), the loads are adjacent, in a horizontal position. However, depending on the position of the loads, vertical optimal joining configurations are possible.

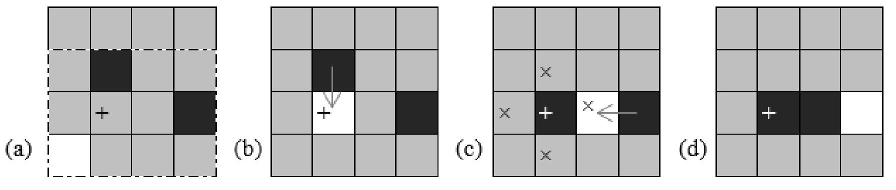


Figure 4. Joining procedure of two loads in dual-load retrieval: (a) joining location is selected, (b) first load moves there, (c) second load moves to its adjacent, (d) loads are ready to be retrieved together.

The third step calculates the number of moves needed to bring the loads jointly to the I/O point. In lemma 3 we prove the optimal way to move two adjacent loads is via so-called "dual-load" moves. Regardless of a horizontal or vertical position of the loads at the joining location, two types of dual-load moves are available: 5-moves and 7-moves. In the following, we explain them in detail. The smallest series of steps that is needed to perform a joint move is via 5-moves. As shown in Figure 4(a), the escort takes three steps to reach the proper position that makes space for the loads to move closer to the I/O point. Then it takes two more steps to move both loads ahead. A series of 5-moves are performed until no more move of this type is possible, i.e. the loads reach one side of the grid. After that, 7-moves are performed as shown in Figure 5(b). Here, the escort takes 5 steps to reach the proper position, and then two more steps are required to move the loads. This is repeated until the loads are retrieved. In the fourth step of the algorithm, the solution is picked with the total minimum number of moves.

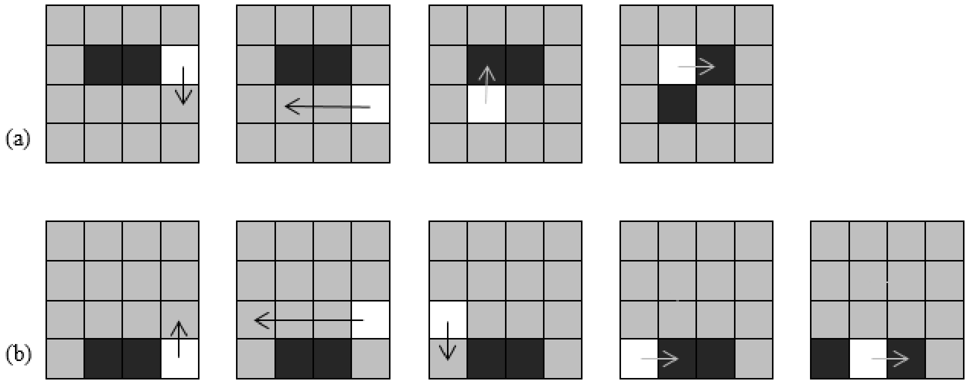


Figure 5. Demonstration of moves in dual-load retrieval: (a) 5-step, (b) 7-step.

Lemma 3: Moving two adjacent loads to the I/O point is optimal when there is no load between them during the retrieval steps.

Proof: We prove this lemma by contradiction. Suppose that one or more items are located between the two loads, we then show the number of steps can be reduced if there is no intermediary item. Assume there is one item between the loads, this means their rectilinear distance is two. As demonstrated in Figure 6(a), at least nine steps are required to move both loads one space unit. By simply eliminating the in-between item, as shown in Figure 6(b), the number of steps reduces to seven. This single item between loads results in two extra escort steps merely to bypass this item. The same argument holds for the case where the loads are vertically aligned. In a similar fashion, having k >1 items between the loads will result in $2k$ extra escort steps to reposition the loads one space unit. This means having no load between the loads leads to the minimum number of steps.

■

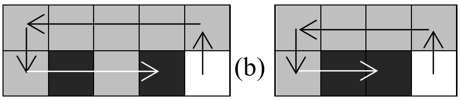
(a) (b)


Figure 6. Moving loads with different amount of interspace: (a) one interspace, (b) no interspace.

Theorem 1 formulates the number of moves needed in an optimal method of retrieving two adjacent loads. Before that, we make the following observations.

Observation 1: In order to determine the number of steps required to retrieve loads from given positions, it is sufficient to track the number of moves made by the escort. This is because each load move corresponds to an escort move.

Observation 2: Due to symmetry of the system, the number of moves required to retrieve two load at locations (i, j) and (j, i) are equal.

Theorem 1: The minimum number of moves to retrieve two horizontally aligned adjacent loads in a puzzle system is:

$$\begin{array}{ll} 7i + 3j - 9 & i > j \\ 10i - 9 & j = i \\ 7j + 3i - 13 & i < j \end{array}$$

where (i, j) is the location of the load closer to the I/O point and the escort is behind them.

Proof: According to Lemma 3, two adjacent loads should travel by dual-load moves. In this strategy, we keep track of the route of the first load and the second load follows it. The first load can move either leftward or downward, using 5-moves and 7-moves. Figure 7 shows an example of a typical route in this approach. An observed property of the dual-load moves is that the route changes direction every time after two 5-moves. In the case of $j > i$ there are *i* pairs of 5-moves necessary for a total of $5(2i) = 10i$ moves, after which the position of the first load should be $(1, j - i - 2)$. Next, $j - i - 2$ 7-moves are required to retrieve the first load for $7(j - i - 2)$ moves. In the end, an additional one move is performed to retrieve the second load, thanks to extra empty space obtained by retrieving the first load. Therefore, in total $10i + 7(j - i - 2) + 1 = 7j + 3i - 13$ escort moves are necessary. In case *j*= *i*, and *i* is an odd number, the first load can reach the I/O point with i-1 pairs of 5-moves. An additional move is necessary to retrieve the second load for a total of $5 \times 2(i - 1) + 1 = 10i - 9$ moves. If *i* is an even number, 2i-3 5-moves is needed, and then an extra 7-move and an additional single move are needed to retrieve both loads. In total $5(2i - 3) + 7 + 3 = 10i - 9$ moves are needed which shows the results are the same for both even and odd i. the case for $j < i$ follows in a similar fashion.

■

As a corollary to this theorem, according to the symmetry property stated in Observation 2, the same approach can be used for the case of vertical alignment of the loads. The formulation is as follows:

$$\begin{array}{ll} 7j + 3i - 9 & j > i \\ 7i + 3j - 13 & j < i \\ 10i - 9 & j = i \end{array}$$

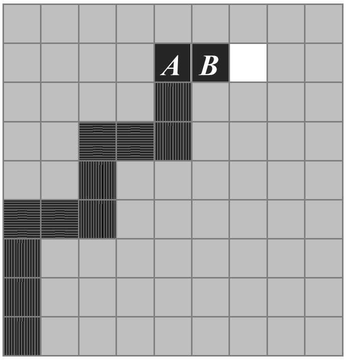


Figure 7. A typical dual retrieval route for two loads *A* and *B*.

In the dual-load retrieval method of algorithm 1, the optimal joining location is not unique. For instance, when the adjacent loads are located at the diagonal, two optimal joining locations exist: one space unit to the left and one space unit to the south.

Algorithm 1 helps to find the optimal solution. However, since the number of joining locations that are evaluated is $O(N^2)$, and the number of steps to jointly retrieve the loads is $O(N)$, the algorithm is $O(N^3)$, where N is the size of the system. Therefore, in addition to this optimal method, we propose a heuristic, that yields a near-optimal solution in a considerably shorter time. This heuristic can be easily adapted to retrieve more than 2 loads as will be explained in section 2.3. Algorithm 2 shows the steps required for two loads (i_1, j_1) and (i_2, j_2). The location of the load closest to the I/O point is denoted by (i_c, j_c) and the location of the farther load is denoted by (i_f, j_f). The subroutine introduced in this algorithm finds J and L as the joining location and the location of the adjacent load respectively, based on location of the loads as the input. For $(i, j) \neq (i_c, j_c)$, either $(i_f - i) = 0$ or $(j_f - j) = 0$.

Therefore, the formula in line 5 results in $(i+1, j)$ or $(i, j+1)$, if $i_f > i_c$ or $j_f > j_c$ respectively.

Algorithm 2: Heuristic method for two loads

1: let $(i,j) = (min\{i_1, i_2\}, min\{j_1, j_2\})$ be the joining location
2: **Subroutine** $\boldsymbol{JL\,(J, L, (i_1, j_1), (i_2, j_2))}$
3: **if** $(i,j) \neq (i_c, j_c)$ **then**
4: $J := (i,j)$
5: $L := (i + \frac{i_f - i}{(i_f - i) + (j_f - j)}\ ,\ j + \frac{j_f - j}{(i_f - i) + (j_f - j)})$
6: **else** $L := (i_c, j_c)$
7: **if** $i > j$ **then** $J := (i-1, j)$
8: **else** $J := (i, j-1)$
9: **end if**
10: **end if**
11: **End subroutine JL**
12: move the load located at (i_c, j_c) to J
13: bring the other load to L
14: move the loads jointly to the I/O point

To compare the performance of the heuristic with the optimal method, numerical results for 20 random instances of dual-load requests are presented in Table 3. To generate random requests, for any given *N*, two unique locations with random coordinates $(i\ ,j)$ are picked. Then, the loads stored at these locations are retrieved using both optimal and heuristic methods to compare the results. This is repeated 20 times for each system size. The optimal and approximate number of moves for dual-load retrieval are presented in column Avg. Opt. and Avg. Aprx., for systems of different sizes, averaged over 20 instances. The average gap between the number of optimal and heuristic methods is presented in column Avg. Gap, together with the minimum and maximum gap. The last column shows the average computation time for the optimal method. The computation time for the heuristics is negligible. The heuristic method appears to perform near-optimal. In fact, it appears the heuristic performs optimally in more than half of the instances.

Table 3. Comparison between optimal dual-load retrieval and the heuristic method.

System Size (N)	Avrg. Opt.	Avrg. Aprx.	Avrg. Gap	Compt. time (s)
5	21.4	22.1	0.7, (0, 2)	0.18
7	38.0	38.4	0.4, (0, 2)	0.61
10	63.4	63.8	0.4, (0, 2)	1.64
20	133.8	135.9	2.1, (0, 10)	18.62
50	346.2	358.6	12.4, (0, 46)	273.10

2.3. Multiple Load Retrieval Method

In this section, we first consider retrieving three loads in the system and then generalize it to more than three loads. In the case of three requested loads, each load can be retrieved individually or together with one or two other loads. Lemma 4 proves that joining loads is required to obtain the minimum number of moves, similar to lemma 1.

Lemma 4: Retrieving three loads jointly from an optimal 3-load joining location, performs better or equal to retrieving one or all of them individually.

Proof: Setting the joining location at (1, 1), immediately transforms the joint retrieval of three loads into three single-load retrievals, or a single-load retrieval and a dual-load retrieval. A better choice of joining locations saves moves.

■

To join and retrieve three loads, different combinations and sequences for the loads A, B and C exist. For example (AB, ABC) means the loads A and B join first at an intermediate joining location, and then they join C at a final joining location. Similarly, the other alternatives are (AC, ACB) and (BC, BCA). These are the main alternatives for joining the loads, but there are other alternatives that are sub-cases of these main alternatives. For instance, individual retrievals is a case when the joining location is set at (1, 1), or joining all loads at one location is a case when the intermediate and the final joining locations are at the same point. Therefore, we only evaluate the three main combinations in Algorithm 3.

One way to obtain the optimal solution to the problem is enumerating all move sequences to all possible joining location. Therefore the number of moves would be $O(m!\,N^m)$where *m* is the number of loads to be retrieved. As this number grows very rapidly with *m* and *N* we propose a heuristic method for three loads or more. Suppose that the third load (i_3, j_3) is requested in addition to the other two loads. Algorithm 3 shows the steps required to retrieve them together. $(i_a,\ j_a)$ and $(i_b,\ j_b)$ are the locations of the first two loads in the combination *k*. $(i_l,\ j_l)$ is the location of the last load in the combination *k*.

Algorithm 3: Heuristic method for three loads

let $(r, q) = (min\{i_1, i_2, i_3\}, min\{j_1, j_2, j_3\})$ be the joining location
for $k = 1{:}3$
    Subroutine $\boldsymbol{JL}\ (\boldsymbol{J}, \boldsymbol{L}, (\boldsymbol{i_a}, \boldsymbol{j_a}), (\boldsymbol{i_b}, \boldsymbol{j_b}))$
    calculate the number of moves needed to bring the first two loads in the combination to J and L for $M_1(k)$ moves.
    calculate the number of moves needed to bring these two loads jointly to the (r, q) for $M_2(k)$.
    $L_2 \coloneqq (r + \frac{i_l - r}{(i_l - r) + (j_l - q)}\ ,\ j + \frac{j_l - q}{(i_l - r) + (j_l - q)})$.
    calculate the number of moves needed to bring the third load (i_l, j_l), to L_2 for $M_3(k)$.

```
    M(k) := M_1(k) + M_2(k) + M_3(k).
end for
pick the combination k := Argmin{M(k) | k = 1,2,3}.
move the loads jointly from (r,q) to the I/O point.
```

In the heuristic method for three loads, usually, an intermediate joining location is established for combining two loads, in order to minimize the individual moves and maximize dual-load moves. Theorem 2 shows the number of moves required to retrieve the loads after they become adjacent. For more loads, the algorithm can be extended using the same approach.

Theorem 2: The minimum number of moves to retrieve three adjacent loads in a puzzle system is:

$$\begin{array}{ll} 9i + 5j - 11 & i > j \\ 14i - 13 & i = j, j = 3,6,9,\ldots \\ 14i - 6 & i = j - 1, j \neq 4,7,10,\ldots \\ 14i - 1 & i = j - 2, j = 4,7,10,\ldots \\ 14i + 8 & i = j - 3, j = 4,7,10,\ldots \\ 9j + 5i - 17 & \text{O.W.} \end{array}$$

where (i, j) is the location of the load closest to the I/O point and loads are horizontally aligned, having the escort behind them.

Proof: The proof is similar to theorem 1. Again, the moves of the first load are tracked, and the other two loads follow it. The only difference is that the loads move using 7-moves and 9-moves due to an extra load.

■

According to the symmetric property stated in Observation 2, the same approach applies to the case of the vertically aligned loads. The formulation is as follows:

$9j + 5i - 11$	$j > i$
$14j - 13$	$j = i, i = 3,6,9, \dots$
$14j - 6$	$j = i - 1, i \neq 4,7,10, \dots$
$14j - 1$	$j = i - 2, i = 4,7,10, \dots$
$14j + 8$	$j = i - 3, i = 4,7,10, \dots$
$9i + 5j - 17$	O.W.

2.4. Numerical Results

To evaluate the performance of the multiple load retrieval method presented in this chapter, we here compare the total number of moves required to retrieve the loads with single load retrieval method. All calculations are done in MATLAB.

For any given system size, two unique locations are randomly generated. These random locations represent requested loads. For the case of three-load retrieval, three unique locations are randomly generated. The numbers of steps required for retrieval of these loads by different methods are calculated. This is repeated for 100 instances.

Table 4 compares the dual-load retrieval method to individual retrieval and shows savings the dual-load retrieval method can obtain. AvgSL and AvgDL are the averages of the total number of moves in single-load and dual-load retrieval, respectively. The maximum savings are obtained when the loads are positioned at (1, *N*) and (1, *N*-1). The average saving is calculated as (AvgSL- AvgDL)/ AvgSL × 100%. According to Table 4, for large values of *N*, the maximum savings are about 33% and the average savings are about 17% of the number of moves needed for individual retrieval. Note that the savings for small systems are higher than for large systems. This is caused by the effect of the second empty spot that appears next to the I/O point, after the first load has been retrieved, and which makes the distance of the second load to the I/O point one unit shorter. This effect disappears for systems larger than 10.

Table 4. Savings for the optimal dual-load retrieval compared to individual retrieval.

N	**AvgSL**	**AvgDL**	**Max savings (%)**	**Avrg. savings (%)**
5	31.3	24.5	41	20
7	45.6	36.4	38	20
10	82.5	65.8	35	19
15	129.2	105.9	35	19
20	157.4	129.5	34	18
50	437.1	367.3	33	17
100	911.5	755.6	33	17

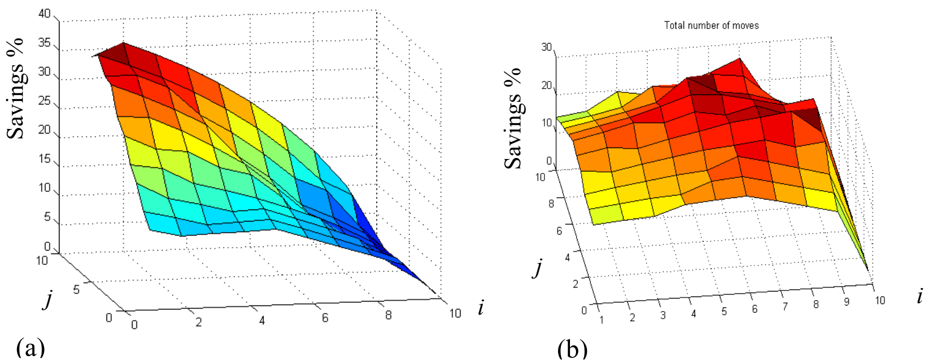


Figure 8. Savings achieved for a given first load located at (a) (1,10), and (b) (5,5), and the second load is at (i, j).

Based on the experiment and Figure 8, we can make the following observation.

Observation 3: when one load is located at $(1,j)$ for j=1..N and the other load at $(i,1)$ for i=1..N, savings in dual-load retrieval are not significant (see Figure 8(a)). Apparently, too many single-load moves are required before the loads become adjacent. On the other hand, when the loads are located at $(1,j)$ and $(1,j\text{-}1)$ for j=1..N or $(i,1)$ and $(i\text{-}1,1)$ for i=1..N, savings are substantial. This is due to the fact that they are already adjacent, and they are located at the very end of the grid.

Table 5 shows the saving that can be obtained by the three-load retrieval method, as compared to individual retrieval. AvgTL is the averages of the total number of moves in three-load retrieval. The maximum saving is calculated when the loads are located at (1, N) and (1, N-1). The average savings are shown in the last column.

Table 5. Savings for three-load retrieval heuristics.

N	**AvgSL**	**AvgTL**	**Max savings (%)**	**Avrg. savings (%)**
5	41.6	29.1	59	30
7	75.3	54.2	52	28
10	113.4	83.5	49	26
15	179.5	134.6	47	25
20	254.8	193.9	46	24
50	886.9	682.3	45	23
100	1683.1	1296.4	45	23

Savings achieved by three-load retrieval shows significantly better performance than single-load. Based on the results, the performance is on average 6% higher than the dual-load retrieval. The values converge as the size of the system grows.

2.4.1. A Parking Lot Case Study

In the numerical study, we considered random load requests. However, in a real system, where a pool of retrieval requests is available, there is an opportunity to group the loads that are close together and gain higher savings. To demonstrate such benefits and to illustrate the effect of joint retrieval in practice, we apply the multiple load retrieval method to a medium-sized car parking system. We consider a 10×10 and a 20×20 single-tier puzzle-based car parking system (total capacity of 99 and 399 cars respectively), where the size of each shuttle is 2.5×4.8 m (see multi-story car parks, 2016). The shuttle speed is 52 m/min in x-direction and is 100 m/min in y-direction. Given these specifications of the system, each move in both directions takes 2.88 seconds. Therefore, although the system's shape is rectangular (25×48 m and 50×96 respectively), it is square in terms of travel time. The system follows the dual-load retrieval method when possible. We assume always 3 people are waiting to retrieve their car with a flexible first-come first-served policy, which means for every retrieval, we take the first car request and pair it with the closer one of the other two car requests. The parking lot operates 24/7 and we perform a Monte Carlo simulation for 1000 random car requests. Table 6 shows the time it takes to retrieve cars individually and in pairs. The total retrieval time for 1000 cars using the dual-load retrieval method is 23.59 hours for the 10×10 105.44 hours for the 20×20 case. On the other hand, the total retrieval time using the individual retrieval method is 30.80 and 132.78 hours respectively. Thus, we can obtain at least 20% improvement in total retrieval time using the dual load retrieval method. This basically means more than 7 hours less retrieval time on a daily basis. Note that these savings are even higher than the average saving for the random case in Table 4, as we first examine the location of the three retrieval request and then pair the closer cars.

Table 6. Savings for a real car parking system.

Number of Bays	Individual retrieval time (h)	Dual-load retrieval (h)	Average saving (%)
99	30.80	23.59	23
399	132.78	105.44	20

2.5. Conclusions

Puzzle-based storage systems are fully automated, unit-load, high-density storage systems that pair a small footprint with high efficiency in retrieval. In this chapter, we first proposed an optimal dual-load retrieval method that, compared to single-load retrieval, saves on average 17 % in retrieval time by bringing the loads first together to an optimal joining location. In addition, a heuristic method is proposed for retrieving loads in pairs that finds a near-optimum solution much faster. This heuristic is then extended to retrieve three and more loads. For three-load retrieval, the results show that on average a 23% saving can be achieved compared to single-load retrieval. Puzzle-based storage systems are still quite rare in practice. However, as the technology becomes less expensive, space becomes scarcer, and as we move into a 24/7 economy, these systems provide a great opportunity to provide high fulfillment performance. Our algorithms and insights can help to realize such high performance, by properly grouping requests and retrieving them jointly.

This chapter makes some assumptions which may be relaxed in future research. First, we study retrieval of loads on a single storage tier. In multi-tier systems, our results will apply per tier. Second, we assume a single escort. Extension of exact results to systems with multiple escorts is not straightforward, but heuristics results may be possible. Third, the proposed algorithms can be embedded in a simulator to obtain the cycle time savings of different system configurations. Last, we assume loads move only one step at a time, and only one load can move at a time. This assumption is valid, depending on the type of mechanical retrieval system, but it may be possible to extend results to systems with simultaneous load movements.

Chapter 3

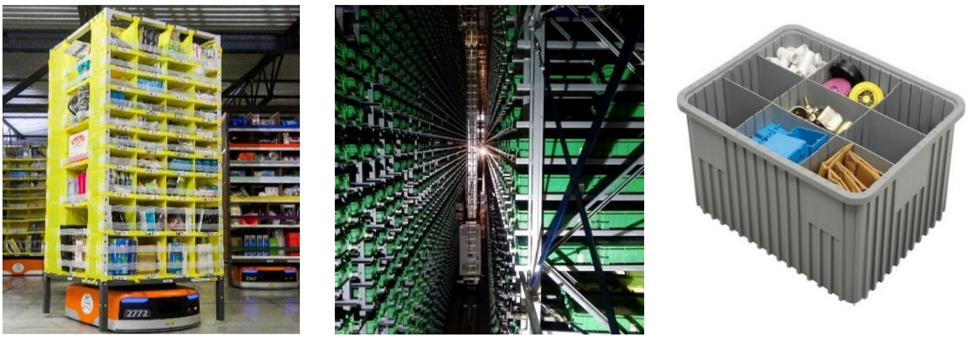
The Impact of Integrated Cluster-based Storage Assignment in Automated Warehouses

3.1. Introduction

Warehouses decouple supply from demand in supply chains. In order to make the warehouse operation more efficient, focus on the order picking process is a prime candidate. Order picking is the most labor-intensive process in a manual warehouse and the most capital intensive in an automated warehouse (Goetschalckx and Ashayeri, 1989; De Koster *et al.*, 2007) and it may take up to 60% of the labor activities (Coyle *et al.*, 1996). In manual warehouses, travel time is typically responsible for almost half of the order picking time (Tompkins *et al.*, 2010). The storage assignment policy, which determines how the products are assigned to storage locations, influences the order picking efficiency. Several storage assignment policies are used in practice to improve the order picking. Below, we review the main policies.

The random storage policy is the most commonly used policy and has been widely studied in the literature (Hausman *et al.*, 1976; Bozer and White, 1984; De Koster *et al.*, 2008). In such a system, products are randomly allocated to the available storage locations. The class-based (ABC) storage policy considers two product attributes to assign products to storage locations, namely *turnover speed* and *storage space needed*. The turnover speed of a product refers to how often it is ordered in a certain period of time and the storage space needed refers to the average amount of space that is required to store the inventory of the product. The COI (Cube-per-Order Index) is a measure that is used to rank the products based on these two attributes. The COI determines how frequently a product is requested per unit of stock space required (Heskett, 1963). In ABC storage, products are first classified into several turnover frequency classes (commonly two or three) based on their COI. Then, the product class with the highest COI is assigned to a block of storage locations (a storage zone) which is most readily accessible. Within each zone, products of the same class are assigned randomly to the locations of the storage zone. In the full turnover-based (FTB) policy, products are sorted in descending order based on COI and assigned to storage locations that are sorted in ascending order based on the travel distance to the depot.

Compared to random storage, ABC and FTB policies can reduce picking travel time substantially by considering the product turnover. However, they ignore the *affinity* between products in the orders. The affinity of two products is defined as the "correlation" of two products in customer orders, i.e. how frequent two products are ordered together. Next to turnover and storage space needed, affinity is another important attribute of the products, which is derived from order history and/or demand forecast. 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. This might unnecessarily increase the order picking time. By measuring the affinity between every two products, it is possible to cluster them in a number of groups based on this correlation. These clusters then can be allocated to the storage locations to reduce the order picking time.



(a) (b) (c)

Figure 1-(a) An Amazon Robotics robot carrying a pod, (b) An AS/R system, (c) A storage bin with nine sub-bins

In robotic mobile fulfillment (RMF) systems, products are typically stored on mobile inventory pods, each holding several dozens of products. When a product is needed for a customer order, the entire inventory pod is transported to a picking station by a robot (Lamballais *et al.*, 2019). An inventory pod carried by a robot is shown in Figure 1(a). The system can benefit if the products ordered together frequently are clustered on the same pod. Another example can be found in AS/R (Automated Storage and Retrieval) systems, where multiple small products may be stored in the same bin, each in a sub-bin. Figure 1(b) shows an AS/R system with aisle-captive cranes and products stored in bins. Figure 1(c) shows a storage bin with nine sub-bins, each containing a different product. Picking products of an order requires visits of the crane to the bins that store the requested products. The number of visits to the bins can be reduced by storing highly correlated products in the same bin.

The dominant approach 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, the products in a cluster (i.e. a pod or a multi-product bin) 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 is A, which has the highest correlation with D, namely 5 times requested jointly. 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 1 distance unit from the depot and the second cluster with the total popularity of 12 is allocated to the second storage location at 2 distance units. Picking all orders, for example, in an AS/R system sequentially, requires 26 distance units travel, whereas by swapping the locations of the clusters, the travel distance is reduced to 25 units.

In this chapter, we propose a new Integrated Cluster Assignment (ICA) storage assignment policy that minimizes the total order retrieval time by considering both product turnover and affinity concurrently. In particular, we consider storage systems where each storage location (e.g. storage bin) accommodates multiple products.

Note that randomness is not always harmful. Next to the benefits such as easy implementation and replenishment, random and ABC storage policies provide higher space efficiency. This is possible due to the fact that 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 an 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. In order to provide a realistic and more accurate comparison of different systems, we include the space sharing in our numerical experiment. This is further explained in section 3.4.1. The remainder of this chapter is structured as follows. Section 3.2 reviews the literature on storage assignment policies based on product affinity and turnover. In section 3.3, we introduce the mathematical model.

Section 3.4 describes extensive numerical experiments in order to evaluate the performance of the proposed policy. In section 3.5, the results and managerial implications are discussed.

3.2. Literature Review

In this section, we first review the literature on general storage assignment policies and then focus on cluster-based storage assignment.

3.2.1. General Storage Assignment

A storage assignment policy determines the allocation of products to storage locations. It determines the responsiveness of the warehouse and consequently the supply chain (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. While the FTB assignment has the lowest expected travel time, the ABC assignment offers comparative benefits too, as it allows weaker assumptions regarding the product turnover frequencies (they do not have to be constant). 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 provides the minimum picking travel time. Since it balances the need for extra storage space that more storage classes require, with the travel time saving due to better storage slotting of the faster moving products. Weidinger and Boysen (2018) propose a scattered allocation that spreads the units of the products in the warehouse to increase the chance for the picker to have the next item in the pick run close by. Lamballais *et al.* (2019) study the effect of spreading the inventory of a product among multiple pods on order throughput time in an RMF system.

3.2.2. Cluster-based Storage Assignment

Products in customer orders may be correlated and it may not be optimal to assign the premium locations to products with higher turnover. Some researchers (Frazelle, 1989; Sadiq *et al.*, 1996; Ballou, 2004) have looked into 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 load the roll cage used in the store replenishment such that the shelf replenishment time within the stores is minimized. Amirhosseini and Sharp (1996) introduce several measures to find the correlation between products. An optimal assignment is possible by enumerating all assignment combinations pairs, which only works for warehouses with a very small number of products. Frazelle (1989) proposes a heuristic for the stock location assignment problem (SLAP) that minimizes the order picking travel time by looking at the correlation between products. He applies a decomposition approach. 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 merge repeatedly until they reach the maximum allowed size. Zhang (2016) creates storage clusters using a so-called '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 an SKU with the biggest turnover frequency as the seed and assigns products with the highest correlation with the seed to the current cluster.

Sadiq *et al.* (1996) consider a correlated assignment in a dynamic environment and propose the dynamic Stock Location Assignment Algorithm (SLAA) that addresses product re-slotting when demand changes and product life cycles are short. They show that SLAA 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 of up to 700 products. Garfinkel (2005) studies correlated storage for zone picking, where products are assigned to the zones based on their correlation, in order to minimize the number of zones visited for all orders. Jane and Laih (2005) define similarity measurement as co-appearance 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 (BOM) 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 BOM tours in the production warehouse.

Rao (1971) and Hansen and Jaumard (1997) use a mixed-integer program to assign products to cluster so that the total affinity in the clusters is maximized. 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) looks at these different 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 mining-based storage assignment approach (DMSA) 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, which is then maximized by applying two heuristics. First, the modified class-based heuristic (MCBH) maximizes the aggregated score within each storage zone, and then the association seed-based heuristic (ASBH) 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. They show in a case study that a similarity-based strategy performs better than class-based and random strategies. Li *et al.* (2016) use a product affinity-based technique in a dynamic storage assignment problem (DSAP). A greedy genetic algorithm (GA) 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*). Table 1 shows an overview of the previous studies on cluster-based assignment policy and highlights the research gap.

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

Paper	**Approach**	**Solution**	**Objective**
Garfinkel (2005)	Decomposition	Heuristic	Number of zone visits
Frazelle (1989)	Decomposition	Heuristic, sequential clustering	Travel time
Jane and Laih (2005)	Decomposition	Heuristic	Zone picking utilization
Zhang (2016)	Decomposition	Heuristic, sum/static seed	Travel distance
Xiao and Zheng (2010)	Decomposition	Multi-stage heuristic	Travel distance
Xiao and Zheng (2012)	Decomposition	Heuristic, GA	Zone visits

Liu (1999)	Decomposition	Heuristic	Sum of similarity measure
Chiang *et al.* (2011)	(partly) Integrated	Heuristic, DMSA	Sum of fitness values
Chiang *et al.* (2014)	Decomposition	Heuristic, MCBH, and ASBH	Weighed support score/travel distance
Li *et al.* (2016)	(partly) Integrated	Heuristic, DSAP-GA	Sum of affinity and turnover/travel distance
Amirhosseini and Sharp (1996)	Decomposition	Heuristic, simultaneous clustering	Travel distance
Sharp *et al.* (1998)	Decomposition	Heuristic	Travel distance
Sadiq *et al.* (1996)	Decomposition	Heuristic, SLAA	Travel time
Bindi *et al.* (2007)	Decomposition	Heuristic	Travel distance
This chapter	Integrated	ICA / heuristic	Total travel time/ number of bin visits

The studies summarized in Table 1, improve the storage assignment by using affinity between products. The contribution of this chapter is to introduce an integrated approach that considers both product turnover and affinity simultaneously to the storage assignment problem. Additionally, we show for which order and product characteristics an ICA policy is beneficial compared to turnover frequency-based policies as well as compared to decomposition approaches. Partly integrated approaches in the literature combine the information regarding the affinity, turnover, and distance into a simplified fitness value. These three sets of information have the same weight in the optimization model. In contrast, this chapter concurrently uses this information in an integrated model, which allocates the products to storage locations while assigning them to the clusters at those locations. In addition, we consider the effect of space sharing to evaluate the impact of storage policies taking into account the capacity constraints. In the next section, a mathematical model is proposed for the integrated approach.

3.3. Problem Description and Mathematical Formulation

To reduce the 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 bin consisting of multiple sub-bins. 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) accommodating multiple products
- The order history is sufficiently large to accurately capture product turnover and affinity.
- Products are picked by order.
- The retrieval machine (i.e. AS/R crane or a robot) brings an entire storage bin or pod to the depot.

We propose a mathematical model for the integrated cluster-assignment policy that uses historical order set O to allocates product $i \in P$ to the cluster (a storage bin) at location $l \in L$, in order to minimize the total retrieval time. The following notation is used.

Parameters:

P The set of available storage locations in the system.

I is the set of products in the assortment,

O is the set of given orders over a certain period of time,

C_p The number of sub-locations (sub-bins) available in the cluster (bin) at location $p \in P$.

μ_i The number of sub-bins needed to store the required inventory of product $i \in I$.

τ_p The one-way travel time from the I/O point to location $p \in P$.

Variables:

x_{ip} =1 if product $i \in I$ is assigned to the cluster at location $p \in P$, x_{ip} =0 otherwise.

$y_{\sigma p}$ =1 if picking order $\sigma \in O$ requires a visit to location $p \in P$, $y_{\sigma p} = 0$ otherwise.

The proposed ICA mathematical model is as follows:

$$min \sum_{\sigma \in O} \sum_{p \in P} \tau_p y_{\sigma p} \tag{2}$$

Subject to

$$\sum_{p \in P} x_{ip} = 1\,, \qquad \forall i \in I \tag{3}$$

$$\sum_{i \in I} \mu_i x_{ip} \le C_p, \qquad \forall p \in P \tag{4}$$

$$y_{\sigma p} \ge q_{i\sigma} xip, \qquad \forall i \in I, \forall \sigma \in O, \forall p \in P \tag{5}$$

$$x_{ip} = 0{,}1, \qquad \forall i \in I, \forall p \in P \tag{6}$$

$$y_{\sigma p} = 0{,}1, \qquad \forall \sigma \in O, \forall p \in P \tag{7}$$

The objective function (2) minimizes the total travel time to pick all customer orders. This travel time depends on the clustering and allocation of the products in the model. The objective forces products with high affinity to be assigned to the same cluster and products with higher turnover frequency will be stored closer to the depot. The ICA model uses the information of customer demand in the data history, $q_{i\sigma}$, which basically provides necessary information on affinity and turnover frequency to cluster and allocate the products. Constraint (3) ensures that each product is assigned to exactly one storage location. Constraint (4) ensures that the capacity of each storage location is not exceeded. Constraint (5) guarantees necessary visits to storage locations of the requested products for all customer orders. Constraints (6), (7) define the binary variables. The storage assignment strategy resulting from the ICA model is called the "ICA policy" in this chapter.

3.3.1. Solution Approaches

In terms of complexity, this model essentially consists of two problems, namely clustering and assignment of 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 the ICA model is NP-hard too. The solution approach is a combination of a greedy heuristic and a general optimization solver. In order to speed up the solution process and reduce the optimality gap, we use an effective heuristic to generate an initial solution that is used by the solver. 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 to the zones. These methods are developed for manual warehouses and are not directly applicable to an automated warehouse. We adopt this reasoning and develop a sequential heuristic approach fit for part-to-picker warehouses that alternates between assignment and clustering, in Section 3.3.2. We use this heuristic as a benchmark to the performance of the ICA model.

3.3.2. A Sequential Alternating (SA) Heuristic

In this section, we develop a fast heuristic solution, inspired by existing picker-to-part sequential methods (Garfinkel, 2005; Zhang, 2016), but adopted to part-to-picker systems. In order to consider both turnover frequency and affinity of the products, the heuristic alternates between assigning products to locations and clustering products together. Clusters may defined according to several affinity measure (see Amirhosseini and Sharp, 1996). We define the affinity between products i and j as follows.

$$\rho_{ij} = \frac{\sum_{\sigma \in \mathcal{O}} q_{i\sigma} \times q_{i\sigma}}{|\mathcal{O}|}, \qquad i, j \in I, i < j, \tag{1}$$

where $q_{i\sigma} = 1$ if order $\sigma \in \mathcal{O}$ contains a request for product $i \in I$, $q_{i\sigma} = 0$ otherwise.

This heuristic is described in Algorithm 1. Note that empty bins are preassigned to locations, so we refer to the, as locations. Let the storage location set L be sorted based on increasing distance to the depot. Each storage location consists of multiple sub-locations that can house a cluster of products. The algorithm assigns the product i with the highest turnover frequency to an available sub-location at the location closest to the depot. Then, while 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_i is selected (Line 13). If a product is not correlated with any other product (Line 14), a product with the highest turnover frequency is assigned to the respective cluster. In this way, highly correlated items are clustered while 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** P, τ_p, C_p, μ_i, I and $\mathcal{O}$
2: **for** each product $i \in I$
3: Compute the turnover frequency F_i over $\mathcal{O}$.
4: **end for**
5: Λ=: set of products $i \in I$ sorted in descending order of F_i.
6: Π=: set of locations $p \in P$ sorted in ascending order of τ_p.
7: **for** each pair of products $i, j \in I$
8: Compute the affinity ρ_{ij}.

```
end for
while Λ ≠ ∅ do
    Assign the first i ∈ Λ to the first p ∈ Π, x_ip = 1.
    Λ = Λ − {j}.
    C_p = C_p − μ_i .
    while C_p > 0 do
            Select j ≔ Argmax{ρ_ij | j ∈ Λ, μ_j < C_p}.
            if j is not unique then
              assign the product with highest F_j to location p, x_jp = 1.
            elseif ρ_ij = 0 then
              Assign the first j ∈ Λ | μ_j < C_p to location p, x_jp = 1.
            else Assign j to location p, x_jp = 1.
            endif
            Λ = Λ − {j}.
            C_p = C_p − μ_j .
    end while
    Π = Π − {p}.
end while
return all x_ip.
```

The ICA model is programmed in AIMMS and solved using Gurobi 7.5. The solution generated by Algorithm 1 is used as an initial solution 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 4GB of RAM, we were able to solve instances consisting of up to 300 orders and 500 products in a time window of two hours.

3.4. Numerical Experiments

In this section, we investigate the impact of parameters such as customer demand and cluster size on the performance of the ICA policy. Two types of systems are studied: we first look at the application of the ICA policy in a conventional warehouse where AS/R system is used. Then, we look at the application of the ICA policy in a modern warehouse run by the RMF system, such as Amazon's. We take a realistic approach by including the effect of space sharing in the class-based storage system. This leads to smaller storage space and shorter average travel times in such systems, compared to FTB and ICA assignment which do not use the storage space as flexibly.

3.4.1. Space Sharing in the Class-based System

We compare the ICA policy with the full turnover-based (FTB), ABC class-based storage systems, and the sequential alternating heuristic. Although FTB, the sequential heuristic, and ICA storage all have the advantage of more precise assignment and shorter travel time, ABC storage has an advantage of space sharing. In order 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 the zones, with a given order quantity of Q_i sub-bins is estimated as:

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

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 $\lceil \sum_i a_i(N_k) \rceil$, expressed in the number of sub-bins. Although ε depends on initial inventory, the Pareto demand curve, and other factors, it is shown by Yu *et al.* (2015) that it is fairly constant and that it can be estimated 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 Equation (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 for the ICA or FTB policies, space cannot be shared, so for each product i space for the whole lot size Q_i must be reserved.

3.4.2. Sample Generation

We consider two system configurations for AS/R and RMF systems. A based example and a number of scenarios are generated for the experiment.

AS/R system: We consider a single-deep mini-load AS/R system, similar to the one demonstrated in Figure 1(b), consisting of one aisle with racks on both sides. Figure 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 distance metric, 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 and the vertical speed is 0.5 m/s and each storage location (slot) is 1 meter wide and 1 meter high. The system dimensions are square in time, i.e. the travel times of the crane from the I/O point to the farthest horizontal and vertical storage locations are equal. Popular products take up 2 sub-bins and less popular products take up 1 sub-bin. We assume the total storage capacity is enough to accommodate all products.

I/O

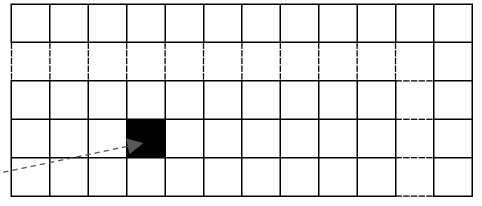


Figure 2- (a) side view of the rack in an AS/R system showing the storage locations and an example travel path of the crane.

RMF system: We consider a robotic mobile fulfillment system, where products are stored on pods, grouped in blocks and where the pods are transported using robots. Figure 3 shows an example of the system layout in this experiment. It consists of six blocks of 10 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 brings a pod that contains one or more of the requested products to the I/O point. 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 is not limiting the results as we look at the total travel time. The average speed of the robot is 1.5 m/s and the required time for a full turn is 2.5s. The time needed for lifting or storing a pod is ignored since it is equal across different systems. Acceleration and deceleration are also ignored. Each pod is 1×1 meter and contains one cluster of products. The inventory level for popular products is 2 slots, and for less popular products 1 slot. The total storage capacity is enough to accommodate all products.

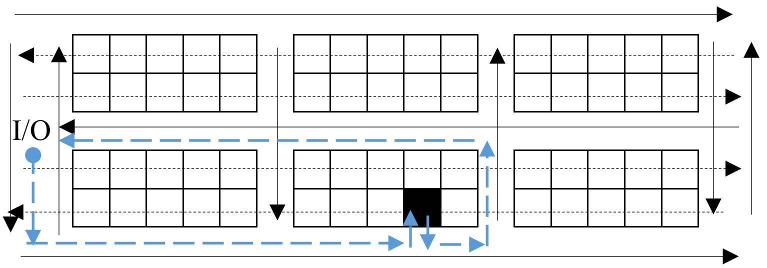


Figure 3- A top view of the RMF system

Data set and scenarios: In order to evaluate the performance of the ICA policy in these two systems, a customer demand data set of a wholesale distributor of office products available at *Warehouse Science* (2018), is used. We create a base example of 200 orders that captures the characteristics of this data set 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, capacity of the bins, the skewness of the ABC curve and the affinity between products. Algorithm 2 in the appendix 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 stock-keeping units (SKUs) in the assortment is varied. The number of sub-bins per bin that defines 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 (1). We generate instances of zero, low, moderate and 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.

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

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	44850	44582	44585	44598	43813

3.4.3. Results of the AS/R Systems

The order picking travel time of the AS/R system is obtained for all scenarios. The execution time to solve the ICA model is limited to two hours per instance except for the base model which is 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 different scenarios defined in Table 2.

Table 4-Travel time of different policies (SA, ABC, FTB and ICA) in an AS/R system in comparison.

Scenarios	SA (s)	ABC (s)	FTB (s)	ICA (s) *	LP Gap %	Savings % ICA/SA	Savings % ICA/ABC	Savings % ICA/FTB
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	-0.5	18.4
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	16.0	13.9	28.3
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	9.1	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
Nr. 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
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	576.5	413.5	**329.5**	0.0	7.9	42.8	20.3
ABC curve								
20/20	1443.7	1282.2	1567.5	1160.7	10.6	19.6	9.5	25.9
20/40	1506.2	1353.2	1701.5	1265.2	9.4	16.0	6.8	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 in bold print for ICA are optimal values.*

The first column in each part of the table shows the parameter varied. The *SA, ABC* and *FTB* columns show the one way travel time in seconds, for the sequential heuristic, ABC and FTB storage policies, respectively. The solution of the ICA model is given in column *ICA*. The solution gap with the LP relaxation bound, which is obtained by relaxing constraint (6) and (7), is given in column *LP Gap*. The savings achieved by applying the ICA policy, compared to the sequential, ABC and FTB policies are given in the last three columns. The results for the base example are repeated in each part of the table to facilitate the comparison. From Table 4, we can make the following observations.

Observation 1: the ICA and SA policies consistently perform equal or better than the FTB policy. This is expected since the ICA and SA policies use affinity as a complementary criterion next to turnover frequency. In the case of zero affinity (no clusters), these policies perform equally. On the other hand, the ICA and SA policies do not always outperform the ABC storage. In the instances of zero affinity and a highly skewed ABC curve, the ABC policy is preferred over the ICA policy. In addition, the ABC policy is preferred over the SA policy, except when the affinity between products is high. This is to a large extent due to the space sharing effect in the ABC storage which reduces the storage space requirement in the warehouse and consequently the travel time.

Observation 2: the relative savings are quite sensitive to the order size. The ICA policy outperforms both ABC and FTB policies even for small-sized orders (e.g. for orders with maximum 2 lines). This is due to the fact that ICA policy benefits from identifying affinity between products in orders and assigns them to the same cluster, to reduce travel time.

Observation 3: with an increasing assortment, the savings increase. Although considering more products implies more travel time for all systems, in larger assortments, randomness in the ABC policy becomes a disadvantage and higher marginal benefits are observed in the ICA policy.

Observation 4: larger number of sub-bins i.e. larger cluster sizes, increases the benefits of the ICA policy. This observation supports the fact that clustering correlated products improves the performance of order picking.

Observation 5: the results show that the samples with higher affinity are candidate for higher savings when ICA policy is used. In the conditions of no affinity, there is clearly no benefit of the policy. When the products are strongly correlated we observe higher benefits.

Observation 6: the savings reduce for more skewed ABC curves. Turnover-based storage systems such as ABC, have better performance opportunities and suffer less from lack of information regarding affinity.

Observation 7: the ICA policy outperforms the SA policy by 13% on average. This confirms that when including the affinity between products in the assignment model, using an integrated approach brings more benefits than a sequential approach.

3.4.4. Results of the RMF System

To obtain the order picking travel time of the RMF system for all scenarios, the same execution procedure is used as in section 3.4.3. The results are presented in Table 5. In general, the same trends are observed as the results from AS/R systems. The key points of these results are that the achievable savings by applying the ICA policy increase when affinity, order size, assortment size, and cluster size increases. The achievable savings decreases when the skewness of the ABC curve increases. This supports the findings in the AS/R system.

Table 5- Travel time of different policies (SA, ABC, FTB and ICA) in an RMF system in comparison

Scenarios	SA (s)	ABC (s)	FTB (s)	ICA (s) *	LP Gap %	Savings % ICA/SA	Savings % ICA/ABC	Savings % ICA/FTB
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	17.5	6.4	25.8
3	7228.5	6971.0	8573.3	5959.7	7.2	25.2	14.5	30.5
4	8987.9	8459.2	10579.8	7229.7	21.4	24.3	14.5	31.7
Assortment size								
100	5756.5	5122.6	6577.7	4956.0	14.5	16.2	3.3	24.7
200	7228.5	6971.0	8573.3	5959.7	7.2	25.2	14.5	30.5
300	7561.9	7985.2	9199.5	6038.2	6.2	25.2	24.4	34.4
500	8758.3	10246.8	10863.8	6557.3	3.7	33.6	36.0	39.6
Nr. of sub-bins								
4	10281.4	9679.9	11517.7	8648.8	11.2	16.2	10.7	24.9
10	7228.5	6971.0	8573.3	5959.7	7.2	25.2	14.5	30.5
16	5661.3	5512.0	6765.7	4593.7	8.4	23.2	16.7	32.1
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	24.6	10.5	30.8
Moderate	7228.5	6971.0	8573.3	5959.7	7.2	25.2	14.5	30.5
High	4881.5	5130.3	6281.2	4240.2	1.1	15.1	16.2	32.5
Very high	4350.8	4528.5	5931.3	**3867.3**	0.0	12.5	14.6	34.8
ABC curve								
20/20	6961.3	6394.2	7934.5	5508.7	7.2	26.9	13.8	30.6
20/40	7228.5	6971.0	8573.3	5959.7	7.2	25.2	14.5	30.5
20/60	6857.6	6082.9	7727.5	5826.5	14.3	24.6	4.2	24.6
20/80	5961.5	5225.5	6546.7	5136.2	13.2	16.2	1.7	21.5

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

3.4.5. Sensitivity of the ICA Policy to Changes in Demand Pattern

This section evaluates the performance of the ICA storage policy when the customer demand pattern changes. First, the base example dataset is used to assign products to the storage locations using the ICA policy, for both the AS/R and RMF systems described in section 3.4.2. Then, a number of orders from the base example is replaced 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 the appendix. We do this for 10% up to 100% change in the base example orderset. Using Monte Carlo simulation, 100 new order sets are generated per scenario, in order to calculate the savings in order picking travel time.

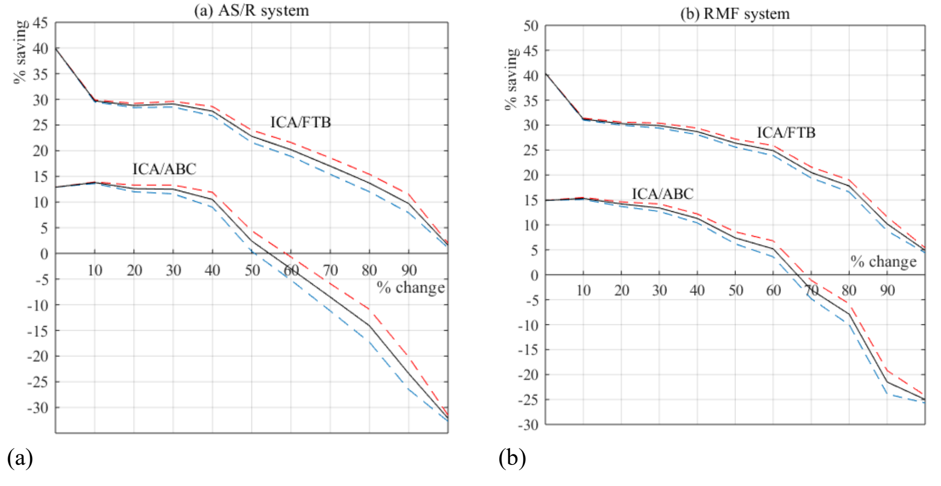


Figure 4- Results of the Monte Carlo simulation of savings obtained using the ICA policy in (a) AS/R system and (b) RMF system, when the demand is disturbed

Figure 4 shows the change of the average saving in the simulation. For each scenario, 95% confidence intervals are calculated. Figure 4(a) shows the average saving 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 95% confidence intervals. Figure 4(b) shows the average saving achieved in a RMF system. The horizontal axis represents the percentage change in the order set compared to the base case and the vertical axis shows the percentage of savings. Both graphs show that the ICA policy remains beneficial despite large changes in the demand pattern. More specifically, in the AS/R system one can benefit from the proposed policy when the customer demand pattern is prone to random changes up to 55% and in the RMF system up to 65%.

3.5. Conclusion

This chapter uses concurrent information on product affinity and turnover frequency to develop an integrated cluster assignment (ICA) algorithm. The ICA method may be applied in environments where multiple products can be stored on a single storage shelf, such as robotic mobile fulfillment systems, where retrieval systems retrieve inventory shelves or totes and bring them to pick stations. The model is NP-hard but can be solved approximately, using a greedy construction heuristic and a standard commercial solver. We compare the system retrieval time for the ICA policy with those for class-based and full turnover-based policies, taking into account the space-sharing effect of class-based storage (which leads to a smaller required storage area than the ICA policy). We also compare the ICA policy with a sequential alternating (SA) heuristic and conclude that the ICA outperforms the SA approach by up to 26%. For the instances we tested, the ICA policy generally leads to shorter retrieval times, even for low and moderate values of product affinity. The travel time reduction may be as large as 40% and depends on customer order characteristics, in particular, the affinity and the skewness of the product turnover frequency curve. We also show that the ICA storage policy is fairly insensitive to changes in the composition of the orders (while preserving the conditional average affinity and skewness of the ABC curve).

Affinity		*L*	*H*
	H	ICA	Class-based / ICA
	L	ICA / Class-based	Class-based

Skewness of the turnover frequency curve

Figure 5- A decision typology on storage assignment policy by considering popularity and affinity of the products

Figure 5 shows a framework that helps to select the best storage policy, based on the skewness of the turnover frequency curve and the conditional average product affinity. A low affinity *L*, means lower than the moderate affinity in the base example and a high affinity *H*, means higher than the moderate affinity in the base example. We particularly find that for orders with high product affinity, the ICA policy consistently outperforms the ABC and FTB storage strategies. This is shown in the top left corner of the framework. In this case, one can slightly benefit even by implementing a sequential approach. However, the benefits of ICA are higher than

SA. The bottom left corner of the framework shows that the ICA policy is favorable in case of orders with low skewness of the turnover frequency curve and a low product affinity. However, when the affinity is very low, i.e. for small order sizes, class-based storage is preferred which benefits from the space-saving effect. For a highly skewed product turnover frequency curve and low affinity level, a class-based storage policy is recommended. This is shown in the right bottom corner. However, if the affinity is very high, a cluster-based assignment can be used, as shown in the top right corner.

Future research should look at a more dynamic storage assignment, in combination with a changing assortment. In addition, more efficient solution methods may be used to solve the ICA model, e.g. based on metaheuristics, to find better product allocations that help to reduce the optimality gap and solve larger instances.

Appendix

Algorithm 2: Pseudocode for generating the order sets for the base example and scenarios

---

1: **input** Order set $\mathcal{O}$, maximum order size θ, product set I, correlated product subset I_i', affinity level indicator* ξ and Pareto curve Ω.
2: **while** the number of generated orders < $|\mathcal{O}|$ **do**
3: Generate the number of lines η_o from discrete uniform distribution U[1, θ].
4: Generate SKU $i \in I$ according to Ω.
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 I_i'.
10: **else** generate a SKU from I according to Ω.
11: $N_O = N_O + 1$.
12: **end while**
13: **end while**
14: **return** generated orders

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

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

---

input Order set $\mathcal{O}$, maximum order size θ, product set I, affinity level indicator ξ and Pareto curve Ω, base example $\bar{\mathcal{O}}$, rate of change Δ.
Generated order subset $\mathcal{O}'$ with $|\mathcal{O}'| = \Delta.|\bar{\mathcal{O}}|$ using **Algorithm 2.**
Replace randomly $\Delta.|\bar{\mathcal{O}}|$ orders from the base example by $\mathcal{O}'$.
Calculate the conditional average affinity* of $\bar{\mathcal{O}}$, $\bar{\xi}_{\bar{\mathcal{O}}}$ and $\mathcal{O}'$, $\bar{\xi}_{\mathcal{O}'}$.
Calculate $\delta = |\bar{\xi}_{\bar{\mathcal{O}}} - \bar{\xi}_{\mathcal{O}'}| / \bar{\xi}_{\bar{\mathcal{O}}}$.
while δ > 5% **do**
    Divide $\mathcal{O}'$ into 4 chunks.
    Calculate the $\bar{\xi}_c$ for each chunk.
    Eliminate the chunk with the largest δ_c
    Generated a new subset $\mathcal{O}_c'$
    Add $\mathcal{O}_c'$ to $\mathcal{O}'$
    Recalculate δ.
end while
return updated orders

---

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

Chapter 4

Correlated Dispersed Storage Assignment in Robotic Warehouses

4.1. Introduction

Many warehouses, particularly in ecommerce retail, compete for short order throughput times. Order picking time is a critical component of order throughput time. It depends on several factors, in particular on the storage assignment policy, which defines where the products are stored in the warehouse. Random storage is a simple and straightforward policy. However, a product turnover frequency-based storage policy, where products with higher turnover frequency are stored closer to the depot, typically leads to shorter throughput times. Storage policies such as class-based storage and full turnover-based storage use the product ranking by turnover frequency. With class-based storage, products are grouped in a number of classes and locations are grouped in the same number of storage zones. Products with higher turnover speed are grouped in a class and stored in a zone closer to the depot. Within a zone, storage is random. The gain from implementing such turnover frequency-based storage policies is particularly high when orders consist of few products and when product turnover speeds are known and relatively constant. For larger orders picked from large assortments (or when small orders are picked in batch), it may pay off to also focus on correlated storage assignments. Items appearing jointly in such orders can be stored close together and, particularly when items are dispersed over the storage area, a product may be retrieved in close proximity to another product that has to be picked for the same order. Several studies suggest that dispersing product units in the warehouse improves the order picking travel time (Onal *et al.*, 2017, 2018; Weidinger and Boysen, 2018). When replenishing a product, the incoming batch is divided into smaller quantities which are spread over the storage area. Spreading products typically requires a higher replenishment effort, but it can decrease the average proximity to the next pick location or to the pick stations.

The impact of storage assignment policies, based on the historic turnover frequency, correlation of products based on past customer demand and product dispersion, on order picking time has been studied in the literature, but usually in isolation. This chapter aims to combine these storage assignment decisions and see what their joint effect is on the total order retrieval time. Advanced automation technologies, such as autonomous shuttles and robots, provide more opportunities to take advantage of combined storage policies. Such systems are relatively flexible, as the throughput capacity can be adjusted by adding or withdrawing robots. Often, also the storage space is flexible and can be expanded by adding more racks. In a robotic mobile fulfillment (RMF) system (Lamballais *et al.*, 2017), like Amazon Robotics™, or Quicktron™ by Alibaba, the autonomous robots transport mobile storage racks, called storage 'pods', to the work stations. Figure 1(a) shows robots waiting at the aisle entrance to pick orders. Each storage rack (a "cluster") contains a group of products, which can be clustered based on historical correlation. The inventory of each product can be split into smaller quantities and assigned to multiple clusters to increase the dispersion in the system. The pod clusters are then assigned to storage locations and zones, taking into account turnover frequency. The problem is formulated as a mixed-integer program that determines the optimal assignment to minimize the expected order picking time. The problem is NP-hard, making it impossible to solve real-life instances (with large numbers of racks, locations, and products) efficiently. We, therefore, propose a simple construction and improvement heuristic to solve large problem instances efficiently.

A real dataset of a warehouse storing personal care products is used to evaluate the performance of the model. Numerical results show that significant improvement in order picking retrieval time is achieved compared to commonly used policies in practice. The remaining of the chapter is structured as follows. In Section 4.2, relevant research on storage policies and warehouse robotics is reviewed. Section 4.3 introduces the robotic mobile fulfillment system studied in this chapter. Section 4.4 presents the mathematical model for the correlated dispersed storage policy. Section 4.5 discusses the solution approach, and the numerical analysis is presented in Section 4.6. Section 4.7 draws conclusions and discusses further research.

4.2. Literature review

This section reviews literature on storage assignment policies in warehouses and their application in robotic systems. A vast amount of literature has been dedicated to the design and analysis of different storage assignment policies. We therefore limit ourselves to data intensive storage policies. De Koster *et al.* (2007) gives a comprehensive review of literature investigating design choices, including storage assignment, in manual order picking warehouses. The review paper of Roodbergen and Vis (2009) investigates topics such as storage assignment and travel time estimation in automated storage and retrieval (AS/R) systems.

Research on storage assignment policies widely studies the class-based assignment that uses the turnover frequency of the products to rank them, group them into different classes and to assign these classes to storage locations (Hausman *et al.*, 1976; Graves *et al.*, 1977; Zaerpour, Yu and R. B. M. de Koster, 2017; Zou *et al.*, 2018). Several studies discuss the optimal number of classes in the class-based storage assignment (Van den Berg and Gademann, 2000; Petersen *et al.*, 2004; Yu *et al.*, 2015). A full turnover-based assignment uses rules such as the cube-per-order index (COI) introduced by Heskett (1963, 1964), to fully rank the products based on their turnover frequency per unit of stock space required (e.g. per pallet stored). Products at a higher rank are assigned closer to the depot. While some papers (Malmborg and Bhaskaran, 1987, 1990) prove the optimality of COI-based storage, Yu *et al.* (2015) prove for finite number of products, a class-based storage with tiny number of classes is already optimal. Nearly all papers that study storage assignment rules, also study random (or variants, such as closest open location) assignment, often as a benchmark to compare with other policies (Petersen *et al.*, 2004; Onal *et al.*, 2017, 2018; Weidinger and Boysen, 2018).

Correlated storage assignment uses the information of demand correlation between products in addition to information on product turnover frequency. The correlation, or 'affinity', between products, can be calculated in different ways based on the joint frequency by which the products occur in a single order (see Amirhosseini and Sharp, 1996). One common approach in modeling correlated assignment is to sequentially group the correlated products in a number of clusters, and assign the clusters to storage locations to minimize the picking travel time (Frazelle, 1989; Amirhosseini and Sharp, 1996; Sharp *et al.*, 1998; Zhang, 2016). The sequential approach cannot guarantee the optimality of the correlated assignment because it decomposes the problem into a clustering and assignment problem. A similar approach is to assign correlated products to the same picking zone to minimize the number of zone visits in order picking (Garfinkel, 2005; Xiao and Zheng, 2012). Conversely, Jane and Laih (2005) assign correlated products to different zones, which increases the workers' utilization. A different approach to correlated assignment is with semi-integrated models that define a so-called similarity or fitness measure as a function of relevant measures such as correlation, turnover, and distance, to assign the products to the optimal storage location (Chiang *et al.*, 2014; Li *et al.*, 2016). The performance of these semi-integrated models depends on the relevance of the defined fitness values because they maximize the sum of the fitness value. Mirzaei *et al.*, (2020) propose an integrated model that optimally clusters and assigns products to storage locations to minimize the total picking travel time, based on historical demand.

Dispersed assignment, splitting the inventory of each product and assigning them to several locations, also referred to as scattered or explosive storage, may help to improve the order picking process. Weidinger and Boysen, (2018) show that evenly spreading product inventory over a manual warehouse, reduces the expected travel time of order picking by reducing the expected distance to location of the next requested product. When a product needs replenishment, they minimize the maximum distance between each pick station and the closest inventory location of the product. Lamballais Tessensohn *et al.* (2019) show that the pod-inventory reorder level of products in robotic mobile fulfillment systems (see Section 4.3.1) has a significant impact on the throughput time, for the cases of dispersed and non-dispersed products. They recommend replenishing a pod before the inventory level becomes zero. (Onal *et al.*, 2017) split the inventory of a product into several storage lots and randomly assign them to as many unique storage locations as possible. They show a higher explosion ratio up to 80%, that is the total ratio of the number of storage lots to the inventory per product, leads to shorter mean fulfillment time. A drawback of the limited existing dispersed assignment methods is that they do not consider the information available on the customer demand to optimally spread the inventory over the storage area. Using the joint-order correlation between products can help to disperse products in such a way that products which are more likely to be requested together are assigned in close proximity to reduce the travel time between sequential picks of order lines.

Choices of appropriate storage assignment in automated systems has been extensively investigated since the introduction of such systems (Hausman *et al.*, 1976; Zaerpour *et al.*, 2015; Mirzaei *et al.*, 2017; Zaerpour, Yu and R. de Koster, 2017; Zou *et al.*, 2018; Lamballais Tessensohn *et al.*, 2019). Azadeh *et al.* (2019) provide an overview of recent developments in automated warehouse systems. Storage assignment in robotic and automated warehouses may differ from storage assignment in manual warehouses because of technological and operational differences. For instance, in these automated systems, products are stored on movable storage units such as bins or pods, which can contain multiple products. This means a robot transports the whole storage unit when a product on it is requested. The inventory and combination of products on each storage unit has a high impact on the order picking performance. This chapter introduces a storage assignment method in robotic warehouses that optimally spreads the inventory of products over multiple storage units while clustering highly correlated products on the same storage unit, to minimize the retrieval time, based on historical customer demand.

4.3. System Description

Section 4.3.1 describes the robotic mobile fulfillment system and introduces the assumptions. Section 4.3.2 explains four choices for storage assignment that are considered in this chapter to investigate the operational performance of the order picking process in robotic systems.

4.3.1. Robotic Mobile Fulfillment Systems

In part-to-picker robotic mobile fulfillment (RMF) systems, products are stored on mobile pods that are transported by automated vehicles. Each pod contains multiple stock-keeping units (SKU), each in a compartment at different height levels, accessible for a manual picker. The stored pods are grouped in storage blocks. Each storage block consists of multiple floor locations that can each store a pod. Figure 1(b) shows six storage blocks, each consisting of twelve floor locations. The grey cells represent the floor locations occupied by pods. The white cells represent open locations that can accommodate pods. The black cell shows a pod that is requested to fulfill a customer order. In case, the order consists of more products, more pods may be required. The pod positions are accessible from the horizontal aisles, but not from (vertical) cross-aisles. All the aisles are single-directional, except the front and end cross-aisles, (the most left and right cross-aisles), to save aisle space and avoid congestion. An empty robot can travel underneath the pods. The allowed driving directions for the robots within the aisles and underneath the pods are shown on the left and top of Figure 1(b). Two pick stations are shown on the left and right sides. The dashed line shows a path that a robot can take from the left pick station to retrieve a requested pod. At the pick station, the requested products are picked from the pod. The robot then returns the pod to an open location in the storage area.

(a)

(b)

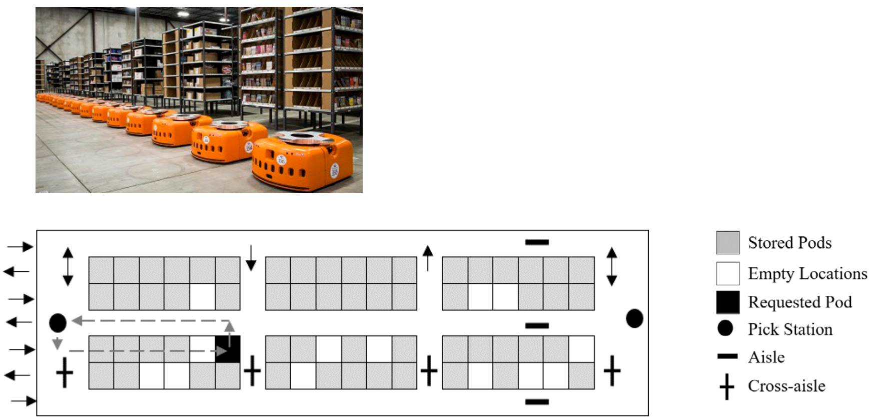


Figure 1. (a) Robots idling next to storage pods (Source: insidelogistics.ca), and (b) a top view of the storage locations in a typical RMF system.

The storage locations may be grouped in storage zones based on their distance to the pick stations. Figure 2 shows a division in two zones *1* and *2* for the system of Figure 1(b). Zone 1 is dedicated to the pods with higher turnover frequency and

Zone 2 to the remaining pods. The turnover frequency of a pod is defined as the collective turnover speed of the products it carries. Within each zone, the pods are randomly assigned to storage locations.

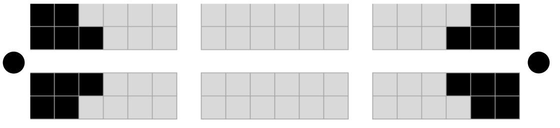


Figure 2. A top view of the storage locations divided into two zones, zone 1 is shown in black, and zone 2 is shown in gray.

4.3.2. Inventory Allocation Choices for RMF Systems

Several inventory allocation policies can be applied in RMF systems. Each pod can carry one or multiple products. Three allocation decisions should be made: 1) allocation of products to pods, 2) allocation of pods to zones, and 3) dispersion of a product over multiple pods. The quantity of a product stored in the picking storage area can be dispersed over multiple pods. A product is called 'dispersed' when it is spread over more than minimum required number of pods to house its inventory. Table 1 shows an overview of possible inventory allocation choices in different storage policies. The simplest and most commonly used approach is the random (RND) allocation of products to pods and pods to zones. In this policy, products are not dispersed and the entire pick inventory is stored on the minimum required number of pods. A turnover-based (or ABC) assignment policy allocates pods to zones using the pod turnover frequency. However, the allocation of products to pods is random and products are not dispersed. On the other hand, a correlated storage assignment (CRL) allocates products to pods based on the correlation between products, but the pod to zone allocation is random. Products are dispersed over multiple pods such that products with a higher correlation share the same pod. The correlated dispersed assignment (CDA) policy, purposefully allocates products to pods, pods to zones and disperses products over pods, using both product correlation and product turnover speed. This policy is explained in detail in Section 4.4.

Table 1. Four inventory allocation choices for RMF.

Policy	Product to pod	Pod to zone	Product Dispersion
Random (RND)	Random	Random	No
Turnover-based (ABC)	Random	Turnover Frequency	No
Correlated (CRL)	Product Correlation	Random	Yes
Correlated dispersed (CDA)	Product Correlation	Turnover Frequency	Yes

4.4. Correlated Dispersed Assignment Model Description

Section 4.4.1 presents a mathematical model is to minimize total expected retrieval time to the closest pick station, by deploying a CDA policy. In Section 4.4.2, symmetry breaking constraints are introduced to increase the efficiency of the model. In Section 4.4.3, we develop expressions for the expected retrieval time for various zone and pick station configurations.

4.4.1. Mathematical Model of CDA

In this section, we present a mathematical model for the correlated dispersed assignment (CDA) policy that clusters correlated products by assigning them to the same storage pod. The model aims to minimize the expected retrieval time of the customer orders. We make the following assumptions: 1) the total storage capacity is sufficiently large to accommodate the required number of pods; 2) one order is picked at a time at a pick station and all products of that order sitting on the retrieved pod are picked at once. If an order consists of many order lines, it can be broken down over multiple pick stations, depending on the proximity of the inventory to other pick stations. In this case, orders need a downstream consolidation before dispatching, which is not considered in this chapter; 3) the total inventory of a product is sufficient to pick all orders during the replenishment horizon; 4) the inventory of a product can be dispersed over multiple pods, but its inventory on each such pod is sufficiently large to pick each line; 5) each pod is assigned to a dedicated location and will return to that location after each retrieval. This assumption is relaxed in Section 4.2. The notation used in the model is as follows:

Parameters:

- P the set of available storage pods in the system.
- I the set of items in the assortment.
- $\mathcal{O}$ the set of given customer orders over a certain period.
- q_{io} =1 if product $i \in I$ is requested in order $o \in \mathcal{O}$, $q_{io} = 0$ otherwise.
- π_i the volume of product $i \in I$, in terms of fraction of the pod capacity.
- μ_i the inventory of product $i \in I$ in terms of the number of units received during the replenishment horizon.
- l_i the minimum allowed quantity of item $i \in I$ to be assigned to a pod.
- u_i the maximum allowed quantity of item $i \in I$ to be assigned to a pod.
- τ_p the retrieval time of pod $p \in P$ to the closest pick station.
- C_p the fraction of storage capacity available on the pod $p \in P$.

Variables:

- a_{ip} =1 if item $i \in I$ assigned to pod $p \in P$, $a_{ip} = 0$ otherwise.
- v_{ip} the quantity of item $i \in I$ assigned to pod $p \in P$.
- x_{iop} =1 if pod $p \in P$ is retrieved to pick item $i \in I$ in order $o \in \mathcal{O}$, $x_{iop} = 0$ otherwise.
- y_{op} =1 if pod $p \in P$ is retrieved to pick one or more lines in order $o \in \mathcal{O}$, $y_{op} = 0$ otherwise.

The mathematical model of the CDA policy is now formulated as follows.

$$Minimize \sum_{p\in P} \tau_p \sum_{\sigma\in O} y_{\sigma p} \tag{1}$$

Subject to

$$\sum_{p\in P} x_{i\sigma p} \geq 1, \forall i \in I, \forall \sigma \in O: q_{i\sigma} > 0 \tag{2}$$

$$a_{ip} \geq x_{i\sigma p}, \forall i \in I, \forall \sigma \in O, \forall p \in P \tag{3}$$

$$My_{\sigma p} \geq \sum_{i\in I} x_{i\sigma p}, \forall \sigma \in O, \forall p \in P \tag{4}$$

$$\sum_{p\in P} v_{ip} = \mu_i, \forall i \in I \tag{5}$$

$$\sum_{i\in I} v_{ip}.\pi_i \leq C_p, \forall p \in P \tag{6}$$

$$v_{ip} \geq l_i a_{ip}, \forall i \in I, \forall p \in P \tag{7}$$

$$v_{ip} \leq u_i a_{ip}, \forall i \in I, \forall p \in P \tag{8}$$

$$\sum_{\sigma\in O} x_{i\sigma p} \leq v_{ip}, \forall i \in I, \forall p \in P \tag{9}$$

$$y_{\sigma p} \in \{0,1\}, \forall \sigma \in O, \forall p \in P$$

$$x_{i\sigma p} \in \{0,1\}, \forall i \in I, \forall \sigma \in O, \forall p \in P$$

$$a_{ip} \in \{0,1\}, \forall i \in I, \forall p \in P$$

$$v_{ip} \geq 0, \forall i \in I, \forall p \in P$$

The objective function (1) minimizes the total expected retrieval time to pick all the products in the given order set. Constraints (2) make sure that at least one pod containing the requested product in each order is retrieved. Constraints (3) ensure that a product may be picked from a pod only if it is already assigned to that pod. Constraints (4) guarantee that a pod will be retrieved for an order if at least one product from that order is requested from the pod. $M = |I|$. Constraints (5) ensure that partial assignments of each product to different pods add up to its inventory. Constraints (6) ensure that partial assignments respect the capacity limit of each pod. Constraints (7) and (8) define the quantity assigned to each pod. Constraints (9) ensure that each pod can serve as many orders as its inventory allows.

4.4.2. Symmetry Breaking Constraints

The model introduced in Section 4.4.1 provides the optimal assignment for the stock quantities of products to pods, pods to locations, and inventory to orders. It can be shown that the CDA model is NP-hard because it can be reduced to a bin packing problem when the order set is reduced to one order. Suppose that pods are equivalent to bins with the same capacity and $\tau_p = 1, p \in P$. Volume of the items are calculated as $\mu_i \pi_i$ for all the products in the order. Now, the optimal assignment of all products in the order to pods in order to minimize the retrieval time of items in the order is equivalent to the minimum number of bins needed to pack inventory of the items. This is a bin packing problem which is a classic NP-hard problem (Garey and Johnson, 1979).

We can simplify the model by assuming we no longer assign pods to locations but to storage zones, i.e. groups of locations in a certain proximity to the pick stations. The retrieval time of a pod assigned to a random location in a zone is then estimated by calculating the average retrieval time of the pods in that storage zone to the closest pick station. This simplification reduces the search space, as pods are now identical. However, this introduces symmetry in the problem, as all pods in the zone have the same expected retrieval time, which gives many identical (optimal) solutions in product allocation. We, therefore, add symmetry breaking constraints to the model.

The assignment of a cluster of products to pod p_1 in zone 1 with expected retrieval time of τ_1, for instance, is similar to the assignment of this cluster to pod p_2 in the same zone. This symmetry arises for each given set of correlated products within each zone and makes the solution space large and complex to solve. Jans (2009) addresses a similar issue in lot-sizing problem at identical parallel machines and includes a set of lexicographic ordering constraints to break the symmetry of assigning jobs to identical machines. In a similar fashion, we use the following constraint to assign product 1 to the identical pods, with smaller ordinal numbers.

$$a_{11} \geq a_{12} \geq \cdots \geq a_{1p}.$$

This constraint breaks the symmetry by assigning product 1 to the pods with smaller ordinal number such that $a_{11} = a_{12} = \cdots = a_{1b} = 1$ and $a_{1b+1} = a_{1b+2} = \cdots = a_{1p} = 0$, where b is the required number of pods. Extra hierarchical constraints are needed to break the tie for the remaining products. The following constraint imposes the ordering on product 2.

$$2a_{11} + a_{21} \geq 2a_{12} + a_{22} \geq \cdots \geq 2a_{1p} + a_{2p}.$$

The coefficients are powers of two and ensure the assignments of different products are distinguished. This is done for all the items to obtain a unique ordering. All the ordering constraints can be summarized in Constraint (10).

$$\sum_{i \in I} 2^{|I|-i} a_{i,p-1} \geq \sum_{i \in I} 2^{|I|-i} a_{i,p}, \forall p \in P \backslash \{1\}. \tag{10}$$

Constraint (10) ensures each product is assigned to the pods with smaller ordinal number. This constraint can strengthen the model by eliminating equivalent solutions via ordering and speeds up the solution finding. Note that if assignment of a product leads to a different correlation rank of products on the pods, the pods orders should be adjusted according to decreasing correlation of the pods.

4.4.3. Retrieval Time Calculation of the Pods

A key parameter in the model is the retrieval time of the pods. The model assigns the pods to zones, but within each zone, they are stored randomly. The average retrieval time of pods in each zone is denoted by $\tau_m, m \in Z$ where Z represents the set of all zones. Given a rectangular storage area, τ_m depends on the zone size, the layout of the zones, and the position of the zones with respect to the pick stations. We assume the size of the zones is given. The impact of the layout of the zones and position of the pick stations on the expected retrieval time of the pods are studied in this section.

For each storage location, we calculate the travel time to the closest pick station using Manhattan distance. However, the aisles only allow one-directional traffic. Pods stored at some of the storage locations will move away from the closest pick station in the first part of the retrieval process. The Manhattan distance travel time, therefore, must be adapted accordingly as $\tau^l = t^l + \delta^l$, $\forall l \in L$, where L is the set of storage locations, t^l is the one-way travel time from location l to the closest pick station according to the Manhattan distance and δ^l is the additional retrieval time of this location due to single-directional traffic in the aisles. The value of δ^l depends on many factors, including storage capacity, size of the storage blocks and the number and position of the pick stations. For the system shown in Figure 1(b) for instance, δ is on average smaller than the travel time along 3 storage space units.

This implies that the Manhattan distance can be used to access each storage location, with a fixed penalty.

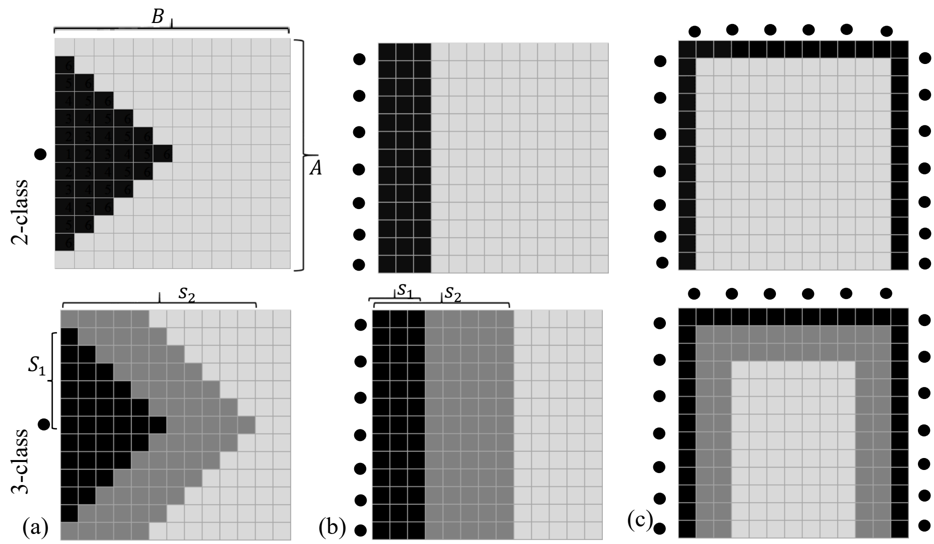


Figure 3. Storage zones for 2-class (top) and 3-class (bottom) storage when there is/are (a) one pick station on the middle-left, (b) many pick stations on the left side and (c) many pick stations on the left, top and right sides.

Figure 3 illustrates how the storage area can be zoned in two storage zones (top) and in three storage zones (bottom), depending on the number and position of the pick stations. The aisles are removed from the figure for simplicity. We consider three configurations of the pick stations. Figure 3(a) shows one pick station on the left side in the middle. Figure 3(b) shows the zones and boundaries for a storage system with many pick stations on the left side, i.e. equal to the number of aisles. The system in Figure 3(c) has many pick stations on the left, top and right sides. The black-colored area shows the pods with fast-moving products and the light gray-colored area shows the pods with slow-moving products. The dark gray-colored area in the 3-zone systems is used for pods with medium turnover speed. These areas are defined based on the Manhattan distances travel time of each storage location to the closest pick station. Figure 3 shows that in the system with one pick station, the fast-moving class has a triangular shape and in the system with many pick stations, the fast-moving area has a rectangular or reverse U shape.

Retrieval Time Distribution Functions. To simplify calculations, we assume a continuous storage location space for each zone. Figure 4(a) shows a continuous representation of a rectangular zone. The area within the dashed triangle, set $\Delta_t = \{T | T \leq t\}$, includes all the locations within travel time *t* of the closest pick station, where T is a random variable representing the retrieval time of a pod at those storage locations. The conditional expected retrieval time from a random location in the storage zone $m \in Z$ is computed as:

$$\tau_m = E(T \leq t | s_{m-1} < T \leq s_m) = \int_{t=s_{m-1}}^{s_m} tf(t)\, dt\,,, s_{m-1} < t < s_m, \tag{11}$$

where $f(t)$ is the conditional probability density function (pdf) of T, s_m is the travel time to the furthest storage location of zone m and $s_0 = 0$. The conditional cumulative distribution function (CDF) of retrieval time T, given that T is in zone m, is computed as:

$$F_m(t) = P(T \leq t | s_{m-1} < T \leq s_m) = \frac{Area\ of\ the\ region\ \{T \leq t\ and\ s_{m-1} < T \leq s_m\}}{Area\ of\ the\ region\ \{s_{m-1} < T \leq s_m\}}, s_{m-1} < t < s_m \tag{12}$$

The expressions for the expected retrieval time of the pods in each zone can be obtained. Now, we derive the corresponding expressions for the configurations shown in Figure (3).

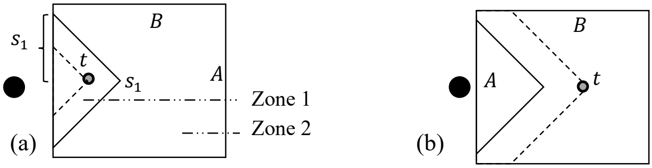


Figure 4. A continuous representation of the storage area (in travel time units) divided into zone 1 and zone 2.

Retrieval Time Expressions for a Single Pick Station on the Left Side in the Middle. The conditional CDF for the system with one single pick station (denoted as S) can be derived using Formula (12). Figure 4(a) shows that the area of zone 1 has a triangular shape with the hypotenuse of $2s_1$ and height of s_1. We assume $A \leq B$ for calculation purposes, where A and B denote the height and the width of the system respectively. A and B are sufficiently large. We also assume $s_1 \leq \frac{A}{2}$ and $s_2 \geq \frac{A}{2}$, because it is not common to dedicate more than 25% of the storage area to zone 1, or to assign less than that to zone 1 and zone 2. $s_m, m \in Z$ are given. If these assumptions are violated, it is still possible to derive relevant retrieval time expression for each zone. The area that covers set Δ_t also has a triangular shape, with the hypotenuse of $2t$ and height of t. In figure 4(b), the area that covers set Δ_t has a pentagon shape. The conditional CDF can be derived for the retrieval time in a storage area with two and three zones as follows:

$$F_{2-z}^{S}(t) = P(T \leq t | T \leq s_1) = \begin{cases} \dfrac{t^2}{s_1^2}, & if\ t \leq s_1, \\ \dfrac{A\left(t - \frac{A}{4}\right) - s_1^2}{A.B - s_1^2}, & if\ s_1 \leq t \leq B. \end{cases}$$

$$F^S_{3-\mathrm{z}}(t) = P(T \le t | T \le s_1) = \begin{cases} \dfrac{t^2}{s_1^2}, & if\ t \le s_1, \\ \dfrac{A\left(t - \frac{A}{4}\right) - s_1^2}{A\left(s_2^2 - \frac{A}{4}\right) - s_1^2}, & if\ s_1 \le t \le s_2, \\ \dfrac{A\left(t - \frac{A}{4}\right) - s_2^2}{A.B - s_2^2}, & if\ s_2 \le t \le B, \end{cases}$$

where F^S_{2-z} and F^S_{3-z} are the conditional CDFs of retrieval time in systems with 2 and 3 storage zones respectively, and single pick station. The corresponding conditional pdfs can be derived as follows:

$$f^S_{2-\mathrm{z}}(t) = \frac{dF^S_{2-\mathrm{z}}(t)}{dt} = \begin{cases} \dfrac{2t}{s_1^2}, & if\ t \le s_1, \\ \dfrac{A}{A.B - s_1^2}, & if\ s_1 \le t \le B. \end{cases}$$

$$f^S_{3-\mathrm{z}}(t) = \frac{dF^S_{3-\mathrm{z}}(t)}{dt} = \begin{cases} \dfrac{2t}{s_1^2}, & if\ t \le s_1, \\ \dfrac{A}{A\left(s_2^2 - \frac{A}{4}\right) - s_1^2}, & if\ s_1 \le t \le s_2, \\ \dfrac{A}{A.B - s_2^2}, & if\ s_2 \le t \le B. \end{cases}$$

The conditional expected retrieval time of an arbitrary pod stored in zone 1 and zone 2 in a 2-zone system can now be calculated using Formula (11) respectively as:

$$\tau^S{}_{2-z,1} = \int_0^{s_1} t f^S_{2-\mathrm{z}}(t)dt = \int_0^{s_1} \frac{2t^2}{S_1^2}dt = \frac{2s_1^3}{3s_1^2} = \frac{2}{3}s_1\,,$$

$$\tau^S{}_{2-z,2} = \int_{s_1}^{B} t\frac{A}{A.B - s_1^2} = \frac{A}{2(A.B - s_1^2)}(B^2 - s_1^2),$$

where $\tau^S{}_{2-z,1}$ and $\tau^S{}_{2-z,2}$ are the conditional expected retrieval times of zone 1 and 2 respectively, in a 2-zone system with single pick station. Similarly, the conditional expected retrieval time for each zone in a 3-zone system can be calculated as follows:

$$\tau^{S}{}_{3-z,1} = \frac{2}{3}s_1 ,$$

$$\tau^{S}{}_{3-z,2} = \frac{A(s_2^2 - s_1^2)}{A\left(2s_2^2 - \frac{A}{2}\right) - 2s_1^2},$$

$$\tau^{S}{}_{3-z,3} = \frac{A}{2(A.B - s_2^2)}(B^2 - s_2^2),$$

where $\tau^{S}{}_{3-z,1}$, $\tau^{S}{}_{3-z,2}$ and $\tau^{S}{}_{3-z,3}$ are the conditional expected retrieval times of zone 1, 2 and 3 respectively, in a 3-zone system with single pick station. In the special case of $A = B$, we have $\tau^{s}{}_{2-z,2}(t) = \tau^{s}{}_{3-z,3}(t) = \frac{A}{2}$.

Retrieval Time Expressions for Many Pick Station on the Left Side. In a similar fashion, we can derive the conditional CDF in the case of a system with many pick stations on the left side of the system (denoted as *ML*). Let A be the length of the left side of the area in Figure 3(b) and s_m be the depth (in travel time) of zone *m* in the aisle. Each zone has a rectangular shape with the length of A and width of $(s_m - s_{m-1})$. The area that covers set Δ_t has a rectangular shape with the following conditional CDFs:

$$F_{2-z}^{ML}(t) = \begin{cases} \frac{t}{s_1}, & if\ t \leq s_1, \\ \frac{t - s_1}{B - s_1}, & if\ s_1 < t < B. \end{cases}$$

$$F_{3-z}^{ML}(t) = \begin{cases} \frac{t}{s_1}, & if\ t \leq s_1, \\ \frac{t - s_1}{s_2 - s_1}, & if\ s_1 < t \leq s_2, \\ \frac{t - s_2}{B - s_2}, & if\ s_2 < t < B, \end{cases}$$

where F_{2-z}^{ML} and F_{3-z}^{ML} are the conditional CDFs of retrieval time in systems with 2 and 3 storage zones respectively, and many pick stations on the left side. The respective conditional pdfs can be derived as:

$$f_{2-z}^{ML}(t) = \begin{cases} \frac{1}{s_1}, & if\ t \leq s_1, \\ \frac{1}{B - s_1}, & if\ s_1 < t < B. \end{cases}$$

$$f_{3-z}^{ML}(t) = \begin{cases} \frac{1}{s_1}, & if\ t \leq s_1, \\ \frac{1}{s_2 - s_1}, & if\ s_1 < t \leq s_2, \\ \frac{1}{B - s_2}, & if\ s_2 < t < B. \end{cases}$$

Thus, the conditional expected retrieval time of the systems with 2 zones, is computed as:

$$\tau^{ML}{}_{2-z,m} = \frac{s_{m-1}+s_m}{2}.$$

where $\tau^{ML}{}_{2-z,m}$ is the conditional expected retrieval time of zone $m \in Z$ in a 2-zone system with many pick stations on the left side. For a 3-zones system, identical expressions are derived.

Retrieval Time Expressions for Many Pick Station on Three Sides. In a similar fashion, the conditional CDFs in the case of a system with many pick stations on the left, top and right sides of the system (denoted as *M3*) can be obtained.

$$F_{2-z}^{M3}(t) = \begin{cases} \frac{At + 2t(B-t)}{As_1 + 2s_1(B - s_1)}, & if\ t \leq s_1, \\ \frac{At + 2t(B-t) - [As_1 + 2s_1(B - s_1)]}{(A - 2s_1)(B - s_1)}, & if\ s_1 < t < B. \end{cases}$$

$$F_{3-z}^{M3}(t)=\begin{cases}\dfrac{At+2t(B-t)}{As_1+2s_1(B-s_1)}, & if\ t\le s_1,\\ \dfrac{At+2t(B-t)-[As_1+2s_1(B-s_1)]}{As_2+2s_2(B-s_2)-[As_1+2s_1(B-s_1)]}, & if\ s_1<t\le s_2,\\ \dfrac{At+2t(B-t)-[As_2+2s_2(B-s_2)]}{(A-2s_2)(B-s_2)}, & if\ s_2<t<B,\end{cases}$$

where F_{2-z}^{M3} and F_{3-z}^{M3} are the conditional CDFs of retrieval time in systems with 2 and 3 storage zones respectively, and many pick stations on three sides. The respective conditional pdfs can be derived as:

$$f_{2-z}^{M3}(t)=\begin{cases}\dfrac{A+2B-4t}{As_1+2s_1(B-s_1)}, & if\ t\le s_1,\\ \dfrac{A+2B-4t}{(A-2s_1)(B-s_1)}, & if\ s_1<t<B.\end{cases}$$

$$f_{3-z}^{M3}(t)=\begin{cases}\dfrac{A+2B-4t}{As_1+2s_1(B-s_1)}, & if\ t\le s_1,\\ \dfrac{A+2B-4t}{As_2+2s_2(B-s_2)-[As_1+2s_1(B-s_1)]}, & if\ s_1<t\le s_2,\\ \dfrac{A+2B-4t}{(A-2s_1)(B-s_1)}, & if\ s_2<t<B.\end{cases}$$

The conditional expected retrieval time of an arbitrary pod stored is zone 1 or 2 in a 2-zone system can now be calculated respectively as:

$$\tau^{M3}{}_{2-z,1}=\frac{\left(\frac{A}{2}+B\right)s^2-\frac{4}{3}{s_1}^3}{As_1+2s_1(B-s_1)},$$

$$\tau^{M3}{}_{2-z,2}=\frac{\left(\frac{A}{2}+B\right)B^2-\frac{4}{3}B^3-[\left(\frac{A}{2}+B\right){s_1}^2-\frac{4}{3}{s_1}^3]}{(A-2s_1)(B-s_1)},$$

where $\tau^{M3}{}_{2-z,1}$ and $\tau^{M3}{}_{2-z,2}$ are the conditional expected retrieval times of zone 1 and 2 respectively, in a 2-zone system with many pick stations on three sides. Similarly, the conditional expected retrieval time for each zone in a 3-zone system can be calculated as follows:

$$\tau^{M3}{}_{3-z,1} = \frac{\left(\frac{A}{2}+B\right){s_1}^2 - \frac{4}{3}{s_1}^3}{As_1 + 2s_1(B-s_1)},$$

$$\tau^{M3}{}_{3-z,2} = \frac{\left(\frac{A}{2}+B\right)B^2 - \frac{4}{3}B^3 - [\left(\frac{A}{2}+B\right){s_1}^2 - \frac{4}{3}{s_1}^3]}{As_2 + 2s_2(B-s_2) - [As_1 + 2s_1(B-s_1)]},$$

$$\tau^{M3}{}_{3-z,3} = \frac{\left(\frac{A}{2}+B\right)B^2 - \frac{4}{3}B^3 - [\left(\frac{A}{2}+B\right){s_1}^2 - \frac{4}{3}{s_1}^3]}{(A-2s_1)(B-s_1)},$$

where $\tau^{M3}{}_{3-z,1}$, $\tau^{M3}{}_{3-z,2}$ and $\tau^{M3}{}_{3-z,3}$ are the conditional expected retrieval times of zone 1, 2 and 3 respectively, in a 3-zone system with many pick stations on three sides. The expected retrieval time expressions are used in Section 4.5 to conduct the analysis.

4.5. Solution Approach

Grouping pods into zones and introducing symmetry breaking constraints do not reduce the complexity of the model. General-purpose optimization solvers can only solve small instances for this model. In our tests, Gurobi 9.0 was able to solve instances up to 50 order lines from 20 products within one day of execution. In reality, one may face instances of thousands of products and order lines. Due to the complexity of the model, we propose an efficient and effective construction and improvement heuristic that enables us to solve the model for real size instances.

4.5.1. Preprocessing the Data

In order to design an efficient solution algorithm, the available data of assortment and historical customer orders should be processed to generate useful data. We compute the COI, $F_i, i \in I$ based on product turnover frequency in historical orders and volume of the product. The product loads are sorted in decreasing order so that i_1 represents the most popular product. Since all the products have the same rank in a product load, we use the term products rank instead of product loads rank. Additionally, we compute the product correlation measure $R_{ij} = \sum_{\sigma \in O} q_{i\sigma} \times q_{j\sigma}, i \neq j \in I$, based on the joint turnover frequency of each pair of products in the historical order set. A subset $\mathcal{J}_i, i \in I$ includes all the products correlated with product $i \in I$ where $R_{ij} > 1, j \in I$. To maintain dispersion of product units over the pods, at most u_i units of a product are allowed in each pod. In this algorithm we store the maximum allowed quantity in each pod. The maximum number of pods needed to store product $i \in I$ is therefore $n_i = \left\lceil \frac{\mu_i}{u_i} \right\rceil$. This means that $(n_i - 1)$ batches of size u_i of product i are assigned to $(n_i - 1)$ pods and one batch of size $\mu_i - (n_i - 1)u_i$ is assigned to another pod.

4.5.2. Step I: Initial Feasible Solution Construction

Conventionally, products with higher turnover frequency are assigned to locations with shorter travel times to a pick station. However, we use both turnover frequency and correlation of products to construct an initial feasible solution for the model. The procedure of constructing solution is demonstrated in Algorithm 1. The empty pods are pre-assigned to two zones, such that $b\%$ and $(100 - b)\%$ of the pods are in zone 1 and zone 2, respectively. The value of b is usually less than 20 (Hausman *et al.*, 1976; Zaerpour, Yu and R. B. M. de Koster, 2017). In Step I, we first assign product pairs i and j with the highest correlation to $\min(n_i, n_j)$ pods. u_i and u_j are small enough for pairs i and j to fit on a pod. The remaining batches of i or j are assigned randomly to $|n_i - n_j|$ empty pods. i and j are then removed from I. Then, we select the next product $k \in I$ with the highest total correlation with products in

I^c, the complement of set I with regards to the assortment. Assign k to n_k pods with i and j on them. In doing so, we only compute the correlation of k with the products on the pods with positive remaining capacity (see Algorithm 1 line 8). If there are multiple products with the same total correlation with assigned products, we prioritize the one with the most homogeneous individual correlation, i.e. the one with correlations close to the average. The capacity and product set are updated afterward, and the procedure is repeated until all the products are assigned to pods.

4.5.3. Step II: Improvement of Feasible Solution

To improve the solution, we propose two types of procedure which update the storage assignment and evaluate the solution. Based on the initial assignment, we define for each pod an aggregated pod turnover frequency and pod correlation measures. The aggregated pod turnover frequency is computed by $\mathcal{F}_p = \sum_{i \in I | a_{ip}=1} F_i$. Pod correlation is computed by $\mathcal{R}_p = \sum_{i,j \in I | a_{ip}=1, a_{jp}=1} R_{ij}$. The improvement policies that work with these two measures toward a better solution are as follows.

Within Zone Update Policy: Pods in the same zone, have the same expected retrieval time to the closest pick station. A higher pod correlation increases the benefit of the correlated assignment. In this iterative updating policy, we try to increase $\mathcal{R}_p$ as much as possible by swapping products among the pods within a zone. Pods of each zone are sorted according to $\mathcal{R}_p$ in descending order. For the first ρ-th percentile of the pods, this policy removes product c from the pod if $E_c = \sum_{j \in I | a_{jp}=1} R_{cj} < \varepsilon_z$, where ε_z is the threshold for the minimum acceptable pod correlation level of zone z. Ideally, this product should be replaced with product c'

with the highest correlation with existing products on the pod, $E_{c'} = \sum_{j\in I|a_{jp}=1} R_{c'j}$. Then, starting from the pod with the lowest $\mathcal{R}_p$, the first pod that contains c' is selected. A product swap of c and c' takes place if $\sum_{p\in P} \mathcal{R}_p$ increases by doing so. Swapping is repeated until the improvement is marginal at a given level.

Across Zone Update Policy: The pods (cluster of products) with a higher $\mathcal{F}_p$ should generally be assigned closer to the pick stations, namely to a zone with a lower ordinal position. However, the initial solution might deviate from this to achieve a higher correlation. Therefore, in this updating policy, we sort the pods according to $\mathcal{F}_p$ in descending order, and re-assign them to the zone accordingly.

Algorithm 1: Pseudocode for Correlated Dispersed Storage Assignment

input I, P, R_{ij}, C_p and n_i

Step I- initial feasible solution

Select the product pair with the highest R_{ij}.

Assign batches of this pair to $\min(n_i, n_j)$ random pods. %Batches are defined in Section 4.5.1%

Assign the remaining batches of these products to $|n_i - n_j|$ random empty pods.

Reduce C_p by $u_i\pi_i$ and $u_j\pi_j$ for the pods carrying i and j.

Update $I = I/\{i, j\}$

while $I \neq \emptyset$ **do**

  Select the product $k := \text{Argmax}\{\sum_{i\in I^c} R_{ik} \,|k \in I\}$; in case of a tie $k := \text{Argmax}\{\prod_{i\in I^c} R_{ik}\}$.

  Assign k to n_k pod with the lowest C_p, sufficient to fit a batch of k.

  Update $C_p = C_p - u_k\pi_k$ for pods carrying k, and Set $I = I/\{k\}$.

end while

Step II- improvement within/across zone

Compute $\mathcal{R}_p = \sum_{,i,j\in I|a_{ip}=1,a_{jp}=1} R_{ij}, p \in P$.

Sort $\mathcal{R}_p$, $G :=$ ρ-th percentile of the pods.

while $G \neq \emptyset$ **do**

  On pod $p \in G$, find the product with the lowest $E_c = \sum_{j\in I|a_{jp}=1} R_{cj}$.

    Removes product c from the pod p.
    Find product c' with the highest $E'_{c'} = \sum_{j \in I | a_{jp}=1} R_{c'j}$.
    If $\sum_{p \in P} \mathcal{R}_p$ increases **then**
        Swap c and c'.
    $G = G - p$.
end while
Compute $\mathcal{F}_p = \sum_{i \in I | a_{ip}=1} F_i, p \in P$.
Re-assign the pods to the zones according to $\mathcal{F}_p$.
return product assignment

The algorithm is demonstrated through a simple example of a set of 10 orders from 10 products with homogeneous size: {1, 2, 3}, {4, 5}, {1,2}, {2, 3}, {6, 7}, {8}, {1, 8}, {9, 10}, {1, 2, 8, 10}, {5, 9}. We can compute $F_i = 4, 4, 2, 1, 2, 1, 1, 3, 2, 2$ for $i = 1 \dots 10$ respectively, and $R_{12} = 3, R_{13} = 1, R_{18} = 2, R_{110} = 1, R_{23} = 2, R_{28} = 1, R_{210} = 1, R_{45} = 1, R_{56} = 1, R_{67} = 1, R_{810} = 1, R_{910} = 1$. The products should be assigned to 4 pods with 6 compartments. Assume that each product unit fits in one compartment. Pod 1 is in zone 1 with $\tau_1 = 1$s and pod 2,3 and 4 are in zone 2 with $\tau_p = 4s, p = 2,3,4$. R_{12} is the product pair with the highest correlation. According to Step I of Algorithm 1, we first assign one pair of products 1 and 2 to pods 1-4. Then product 3 which has the highest total correlation with products 1 and 2 is assigned to pod 1 and 2. See Figure 5(a). This is repeated until all the products are assigned. A dashed line separates zone 1 and 2. The results of assignment Step I is shown in Figure 5(b), which leads to total order picking time of 23s. In Step II, we start with within zone improvement. The total correlation on each pod is $\mathcal{R}_1 = 13, \mathcal{R}_2 = 12, \mathcal{R}_3 = 8$ and $\mathcal{R}_4 = 4$. Pod 4 has the lowest total correlation. Products 5 and 6 from this pod can be swapped with products 8 and 10 from pod 2 which changes the total correlation on each pod to $\mathcal{R}_1 = 13, \mathcal{R}_2 = 7, \mathcal{R}_3 = 8$ and $\mathcal{R}_4 = 11$. Since the collective correlation has increased by two, they swap. The results of assignment Step I is shown in Figure 5(c), which leads to total order picking time of 21s. No across zone improvement can be made since the collective turnover frequencies of the pods $\mathcal{F}_1 = 17, \mathcal{F}_2 = 15, \mathcal{F}_3 = 15$ and $\mathcal{F}_4 = 16$, are already in good order.

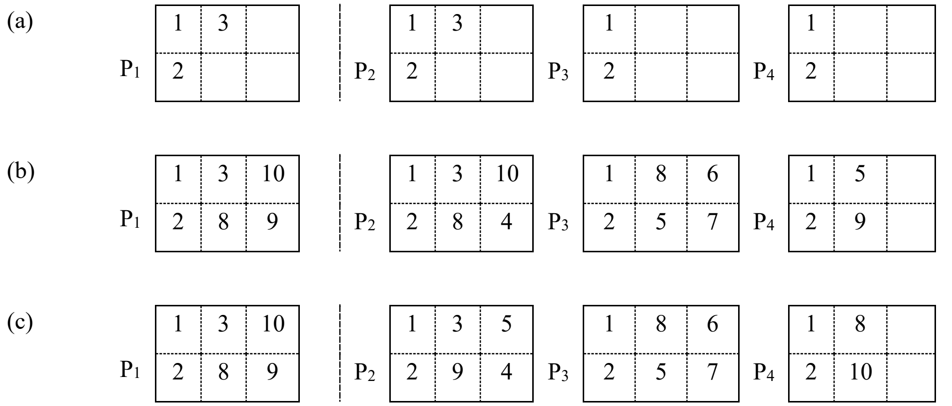


Figure 5- Illustrative progressive example of assignment of 10 products to 4 pods: (a) Step I one iteration, (b) Step I result, (c) Step II within zone improvement result.

4.6. Numerical Analysis

Correlation and turnover frequency of products have a significant impact on the performance of the storage policies. (Hausman *et al.*, 1976)) and Bender (1981) propose two different functions to approximate the turnover frequency of stored products. Hausman *et al.* (1976) formulate the ranked cumulative demand percentage, G, versus the percentage of inventoried items, i, as $G(i) = i^A, 0 < A \leq 1$. Bender (1981) shows experimentally that $G(i) = (B+1)i/(B+i), B \geq 0$ provides a better fit to the ABC curve. Parameters A and B represent the shape factor of the demand curve of the two functions, respectively. For both cases, a smaller shape factor means the demand curve is more skewed to the left, which corresponds to larger contribution to the demand of smaller inventoried items. Inspired by these functions, in Section 4.6.1, we introduce two expressions to approximate the correlation level of the product pairs in customer demand. Section 4.6.2 presents the case description. Section 4.6.3 presents the numerical results.

4.6.1. Assortment Correlation Expressions

The correlation of each product pair R_{ij} is the number of times products i and j are jointly ordered. To approximate the ranked cumulative correlation $Cor(\varphi)$, as a function of the cumulative percentage of product pairs φ, we introduce two correlation expressions as follows.

$$Cor(\varphi) = \varphi^s, 0 < s \leq 1 \tag{13}$$

$$Cor(\varphi) = \frac{(r+1)\varphi}{(r+\varphi)}, r \geq 0 \tag{14}$$

where s and r are the shape factors of the correlation curve. Smaller shape factors mean a more skewed correlation curve. This indicates a smaller number of pairs contribute to a larger cumulative correlation. In Section 4.6.2, we show that both expressions fit well and are statistically significant for the case tested. However, expression (14) provides a better fit.

4.6.2. Case Description

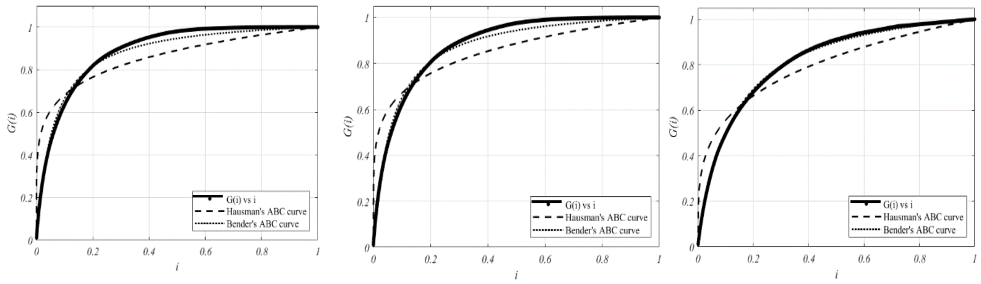
We use a dataset of an international distributor of personal care products in our numerical analysis to obtain realistic insights into the proposed models. In the dataset, there are requests for pallet and case pickings. The pallet picking is excluded from the analysis because it is picked from a different inventory. The data are anonymized due to the privacy concerns of the company. It includes customer demand for three consecutive months in 2018.

Analysis of the Dataset: Table 2 summarizes the order profile in the dataset. The number of active products in the distribution center and the number of orders in the three-month period are given. Each order consists, on average, of 37 product lines. Each line requests on average 1.5 units of a product. Units of products on average require 8 liters of storage space. Figure 6(a) shows the turnover frequency curves of the products in the dataset. The solid-line curve represents the real data points. The dashed-line curve shows the Hausman's Pareto curve and the dotted-line curve shows the Bender's Pareto curve (Hausman *et al.*, 1976 and Bender, 1981). The goodness-of-fit statistics which are presented below each curve support the significance of both fits. Figure 6(b) shows the turnover frequency curve of the training dataset and Figure 6(c) for a random sample of 325 orders from the test dataset. The training and test datasets are obtained by dividing the company's dataset into two sets. See Section 4.6.3 for details. All three analyses in Figure 6 support a statistically significant fit of Bender's function with a higher R^2 value. Therefore, we use the Bender's curve with this coefficient B to parameterize the turnover frequency.

Table 2. Information of order profile in the dataset.

# Products	# Orders	# Order Lines	Order Line Quantity	Product Volume (liter)
2,362	28,139	(37.2, 47.9)*	(1.5, 1.4)*	(8, 13)*

* *(Average, Standard Deviation)*



(a)
B = 0.0594 (0.0589, 0.0599)*
SSE: 1.1, R^2: 0.986, *SE*: 0.022 **

A = 0.1662 (0.1625, 0.17) *
SSE: 17.0, R^2: 0.788, *SE*: 0.085

(b)
B = 0.0636 (0.0631, 0.0642)*
SSE: 1.0, R^2: 0.988, *SE*: 0.021**

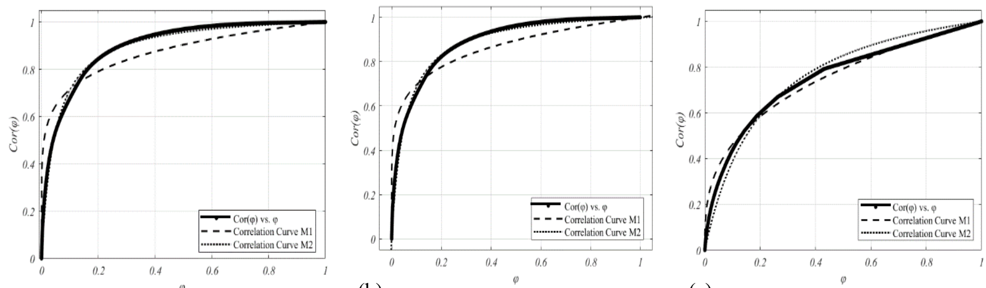
A = 0.1732 (0.1694, 0.177)*
SSE: 15.7, R^2: 0.801, *SE*: 0.084

(c)
B = 0.1248 (0.1243, 0.1253)*
SSE: 0.1, R^2: 0.998, *SE*: 0.008**

A = 0.2548 (0.2499, 0.2597)*
SSE: 5.2, R^2: 0.905, *SE*: 0.066

Figure 6. Turnover frequency curve for (a) full dataset, (b) training set, and (c) a random sample.
* *Coefficients with 95% confidence bounds.* ** *SSE: Sum of squares due to error, SE: standard error*

The analysis result of the correlation between products is presented in Figure 7. The solid-line curve represents the real data points. The dashed-line curve is obtained using Formula (13), and the dotted-line curve is obtained using Formula (14) which are called *M1* and *M2,* respectively. Figure 7(a) shows the correlation in the dataset, 7(b) shows the correlation in the training set and 7(c) shows the correlation in a random sample from the test set. The coefficients of *M1* and *M2* are set out below each graph with a 95% confidence interval. The goodness-of-fit statistics presented below each graph support the fit. We choose model *M2* with parameter r to approximate the correlation curve in our experiment. Because, the results in Figure 7(a) and (b) show a better visual fit, lower sum of squared errors and higher R^2 for *M2* compared to *M1*, while Figure 7(c) shows a nearly equal performance of the two models.



(a)
r = 0.04713 (0.04712, 0.04715)*
SSE: 238.4, R^2: 0.991, *SE*: 0.015**

s = 0.1458 (0.1456, 0.1459)*
SSE: 4687, R^2: 0.819, *SE*: 0.068**

(b)
r = 0.05372 (0.0537, 0.05374)*
SSE: 242.3, R^2: 0.991, *SE*: 0.016**

s = 0.1575 (0.1573, 0.1576)*
SSE: 4244, R^2: 0.841, *SE*: 0.066**

(c)
r = 0.2131 (0.2128, 0.2134)*
SSE: 175.1, R^2: 0.975, *SE*: 0.033**

S = 0.3342 (0.3340, 0.3344)*
SSE: 134.5, R^2: 0.981, *SE*: 0.029**

Figure 7. Correlation curve for (a) full dataset, (b) training set, and (c) a random sample.
* *Coefficients with 95% confidence bounds.* ** *SSE: Sum of squares due to error, SE: standard error*

MRF System Set-Up: We consider a system with 1300 pods, sufficiently large to accommodate all inventoried products. Each storage pod has a size of 1.5×1.5×1.8 meters, with 20 compartments of 100-liter capacity each. At least one unit of each product in the assortment fits in one compartment. In order to maximize space utilization, once a compartment is allocated to a product, the maximum number of units that fits into the compartment q_i, will be assigned to that pod. The storage area has a square shape and is divided into two storage zones. The capacity of zone 1 is 20% of the total capacity. We use the zoning and pick station configuration shown in Figure 3(b) with many pick stations on the left side. The robots move with a speed of 1.5 m/s. Acceleration, deceleration, turning, lifting and congestion are ignored.

4.6.3. Analytical Models of the Retrieval Times

We split the dataset randomly into two parts. 70% of the data is used as the training set (to assign products to pods and pods to storage zones), and the remaining data is used as the test set (to calculate the expected order retrieval time). This means that we solve the assignment model using the training set and use the test set to validate the model. Products and pods are assigned using the information from the training set according to all storage policies in Table 1. Then, we draw 500 random samples of 325 orders, equal to average daily demand, from the test order set. In the next sections, the expected retrieval times are calculated and analyzed.

Retrieval Time Comparison of the Assignment Policies: The total expected order picking retrieval times of the sample orders for the RND, ABC, CRL and CDA policies are presented in Table 3. The first row shows the average and the standard deviation of the expected retrieval times of the sample order sets using each policy. The next rows show the relative benefits (in percentage) of using this policy compared to other policies together with the minimum and maximum savings in percentage. The results show that the policies can be ranked according to their performance as CDA, CRL, ABC, and RND, where CDA outperforms the other three policies by up to 58% on average. CRL takes the second place after CDA, indicating that correlation-based policies perform better than turnover-based ones in this experiment.

Table 3. Evaluation results of the effect of using different storage assignment policies on order picking retrieval time.

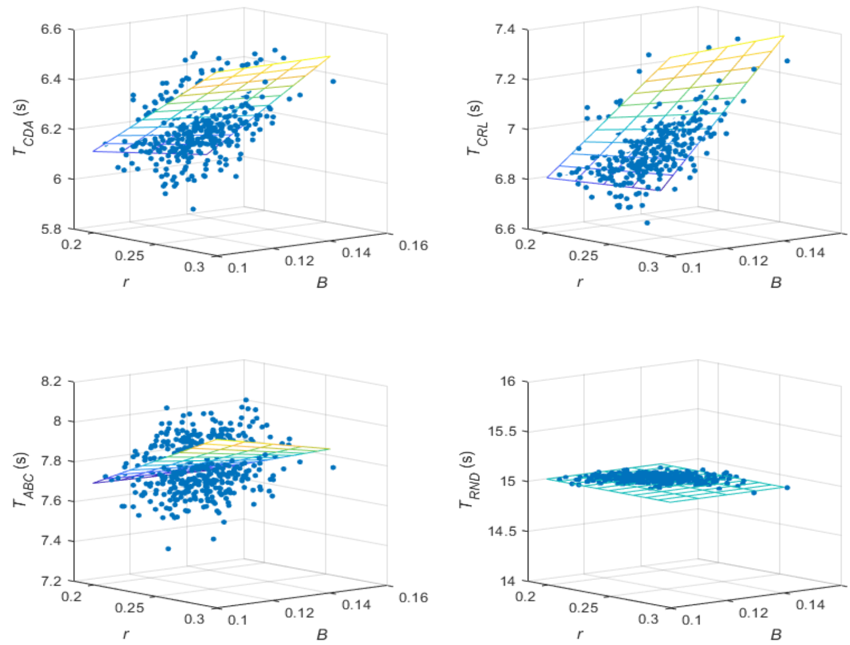
Policy	**CDA**	**CRL**	**ABC**	**RND**
	(69,570, 4,242) *	(77,850, 4,369) *	(86,901, 5,744) *	(167,853, 10,614) *
CDA**		11 (9, 12)	20 (14, 37)	58 (56, 61)
CRL**			10 (4, 15)	54 (51, 56)
ABC**				48 (46, 51)

* *(Average, Standard Deviation).* ** *Average relative savings (Minimum Value, Maximum Value) in percentage.*

Prediction of Performance of Policies with varying *r* and *B*: To explain the variation in expected retrieval times when using different assignment policies, a multivariate regression analysis is used. We investigate the relationship between variables, namely the turnover frequency coefficient (B) and correlation coefficient (r) as the explanatory variables and the expected retrieval time per order line of the sample orders as the dependent variable. The expected retrieval time per order line is obtained by normalizing the expected retrieval time of each sample order over the total number of order lines in the sample order. This normalization eliminates the influence of varying order size on the total retrieval time of each sample. When the interaction effect $r \times B$ was included in the regression models, the *p-values* of the coefficients were not significant at a 95% confidence level. Therefore, we exclude the interaction effect from the regression models. This change did not weaken the explanatory power of the regression model and R^2 remained at the same level. Figure 8 shows the 3D graphs of the regression analysis of 500 sample order sets for all the policies. The vertical axis shows the expected retrieval time per order line for each assignment policy. Below the graphs, the statistical results of regression models are set out. We make the following observations.

Observation 1- The regression of the CDA policy shows that the expected retrieval times per order line increases when r increases. This supports that when the products are highly correlated in the order set (smaller r), the CDA policy has a better performance.

Observation 2- The regression of the CRL policy has a similar behavior as CDA, with a steeper curve. This means the performance of the CRL policy is very sensitive to r values. This is expected behavior since the correlation is the main decision factor in this policy.



(a) CDA

R^2: 0.34, F: 84.30, p-value: 0.00

	Coef.	*S.E*	*t Stat*	*P-value*
b_0	5.37	0.08	68.37	0.000**
r	4.57	0.30	15.11	0.000**
B	-1.18	0.61	-1.94	0.05*

(b) CRL

R^2: 0.54, F: 194.05.49, p-value: 0.00

	Coef.	*S.E*	*t Stat*	*P-value*
b_0	5.73	0.07	76.45	0.000**
r	6.60	0.29	22.87	0.000**
B	-1.64	0.58	-2.83	0.005**

(c) ABC

R^2: 0.10, F: 17.70, p-value: 0.00

	Coef.	*S.E*	*t Stat*	*P-value*
b_0	7.34	0.10	74.55	0.00**
r	2.69	0.38	7.09	0.00**
B	-1.22	0.76	-1.60	0.11

Figure 8. Multivariate regression results of the retrieval times with respect to r and B for different storage policies.

* *Significant at 0.95 confidence level.* ** *Significant at 0.995 confidence level.*

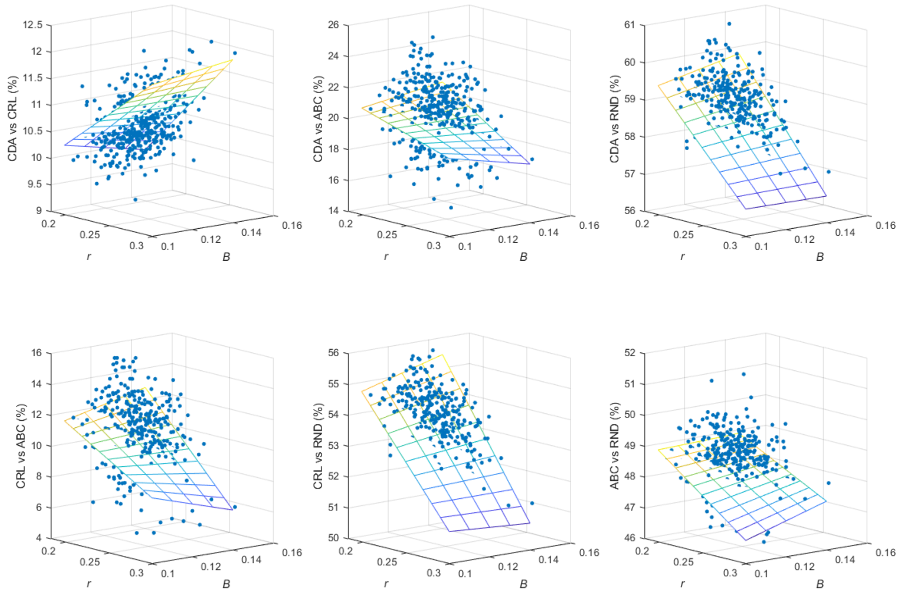
Observation 3- The regression of the ABC policy has a much smaller r coefficient. The coefficient B is not significant at 95%. The small range of B values can serve as an explanation for an insignificant correlation.

Observation 4- The regression curve of the RND policy is flat. This policy is not sensitive to any of the factors, and the expected retrieval time per order line remains constant.

Comparison of Performance of Policies for varying *r* and *B*: To explain the variation in relative savings in total retrieval time of different assignment policies, a multivariate regression analysis is used. For example, the relative saving of the CDA policy compared to the ABC policy is defined as $(T_{ABC} - T_{CDA}) / T_{ABC}$. The turnover frequency coefficient (B) and correlation coefficient (r) are used as explanatory variables, and the relative saving of each two methods is used as the dependent variable. Figure 9 shows the 3D graphs of the regression analysis of 500 sample order sets. The interaction effect of the explanatory variables is excluded due to insignificant *p-values*. We make the following observations.

Observation 5- The relative savings of the CDA policy compared to CRL policy increase with r. This can be explained by the fact that CRL policy loses power with an increasing r while CDA policy is less sensitive to it.

Observation 6- The relative savings of the CDA and CRL policies compared to ABC and RND policies decrease with r. This shows that for order sets with smaller r (higher correlation), implementing the CDA or CRL policy can make a big difference compared to ABC and RND policy. The relative savings of CDA and CRL compared to RND are always substantially higher than when compared to ABC.



(a) CDA vs CRL

R^2: 0.32, *F*: 118.80, *p*-value: 0.00

	Coef.	*S.E*	*t Stat*	*P-value*
b_0	0.07	0.00	21.02	0.00
r	0.19	0.01	14.55	0.00
B	-0.04	0.03	-1.56	0.12

(b) CDA vs ABC

R^2: 0.07, *F*: 18.67, *p*-value: 0.00

	Coef.	*S.E*	*t Stat*	*P-value*
b_0	0.26	0.01	18.40	0.00
r	-0.31	0.06	-5.63	0.00
B	0.03	0.11	0.23	0.82

(c) CDA vs RND

R^2: 0.34, *F*: 128.47, *p*-value: 0.00

	Coef.	*S.E*	*t Stat*	*P-value*
b_0	0.64	0.01	122.53	0.00
r	-0.30	0.02	-15.11	0.00
B	0.08	0.04	1.94	0.05

(d) CRL vs ABC

R^2: 0.16, *F*: 46.50, *p*-value: 0.00

	Coef.	*S.E*	*t Stat*	*P-value*
b_0	0.21	0.02	13.59	0.00
r	-0.54	0.06	-8.96	0.00
B	0.07	0.12	0.58	0.56

(e) CRL vs RND

R^2: 0.54, *F*: 290.42, *p*-value: 0.00

	Coef.	*S.E*	*t Stat*	*P-value*
b_0	0.62	0.00	123.65	0.00
r	-0.44	0.02	-22.87	0.00
B	0.11	0.04	2.83	0.00

(f) ABC vs RND

R^2: 0.10, *F*: 26.36, *p*-value: 0.00

	Coef.	*S.E*	*t Stat*	*P-value*
b_0	0.51	0.01	77.88	0.00
r	-0.18	0.03	-7.09	0.00
B	0.08	0.05	1.60	0.11

Figure 9. Multivariate regression results of relative savings of different storage policies with respect to *r* and *B*.

Observation 7- Coefficient r is always significant. B is either insignificant or has a small effect size which can be due to a small variation domain. This means that r is the primary determinant in the variation in relative saving.

4.7. Conclusion and Research Directions

This chapter studies the design and analysis of storage assignment policies in robotic and automated warehouses, based on historical customer orders, in order to reduce the order picking retrieval time. Previous literature has studied correlated assignment and dispersed assignment separately, mainly in a manual warehouse environment, where an order picker walks along the storage locations and picks the orders. In this chapter, we study the effect of using correlated and dispersed storage policies on retrieval times in robotic mobile fulfillment systems. The correlation between products is derived from the joint requests of products in historical customer demand. Dispersing a product involves assigning its inventory to more than the minimum required number of storage units. A mixed-integer linear program is formulated that clusters highly correlated products on the storage pods while allowing inventory to be dispersed at the desired level. The model assigns clusters to storage zones, and inventory to customer orders in order to minimize the total order picking retrieval time. This model is NP-hard and can be solved for small instances by Gurobi 9.0. A construction and improvement heuristic is developed that can solve real size problems. To evaluate the performance of the model, we use a real dataset of a warehouse distributing personal care products. We define two parameters, B and r, based on Bender's Pareto curve function to capture the product turnover frequency and the correlation between them in the orderset. Regression models are developed for predicting the total retrieval time using these B and r values.

The analytical results show that the correlated dispersed assignment (CDA) outperforms the random (RND), class-based (ABC), and correlated (CRL) assignments. In the case we tested, with large order sizes, the CRL policy has better performance than the ABC and RND policies. The regression models show that the total retrieval time for the CDA and CRL policies increases when r increases. Additionally, we observe that the relative savings of the CDA policy compared to the CRL policy increase with r, but decreases compared to the ABC and RND policies. In the case tested, r has always a significant effect while B has rather a small or insignificant effect, which may be due to small variation in B. These results suggest that managers should pay special attention to the customer order profile, such as turnover frequency and correlation of the products to reduce the operational time and cost. In automated warehouses, the correlated dispersed assignment may result in shorter retrieval times than common policies such as ABC and RND. Furthermore, when the correlation of the products has a highly skewed Pareto curve, the performance of the CDA increases.

Future research should evaluate the performance of correlated dispensed assignment for different cases, especially where B and r show higher variation than in the case we tested. This chapter does not include the replenishment effort in the model and focuses primarily on retrieval effort to reduce the customer response time. Since dispersing inventory may have a high impact on the replenishment time, future research should study the effect of inventory dispersion on the total effort.

Chapter 5

Summary and Conclusion

5.1. Summary

Automated storage and retrieval systems are adopted by many companies to reduce the operational costs in warehouses and to rapidly fulfill customer demand. Automation technology also allows better space usage, e.g. by more narrow transport aisles. Compact automated storage systems do not need transport aisles and allow very high density storage. Deciding on storage and retrieval policies in such systems is an important choice as it affects product retrieval time performance. Chapter 2 discusses optimal and near-optimal retrieval methods in compact storage systems. Chapters 3 and 4 model correlated and dispersed storage and retrieval methods using information on the historical customer demand.

Chapter 2 studies efficient retrievals in puzzle-based storage (PBS) systems, a new type of compact storage system that operates without transport aisles. In these systems, loads are stored on shuttles. Retrieving loads resembles solving a 15-tile sliding puzzle. These systems bring extreme space usage efficiency but can result in long storage and retrieval times. Previous research studies optimal retrieval methods for single loads in which only one load is retrieved at a time. In practice, often, multiple loads are requested together. This chapter proposes a multiple-load retrieval method that minimizes the total retrieval time. The main research question is "*What is the optimal retrieval method in PBS systems (i.e. minimizing the number of required moves) to retrieve multiple requested loads, using one open location?*" This question is answered by first modeling simultaneous retrieval of two requested loads. The optimal retrieval paths for two loads, which go through an intermediary joining location, are obtained. Based on this model, an efficient heuristic is developed that obtains near-optimal retrieval paths for multiple requested loads. Numerical analysis shows that up to 23% savings in total retrieval time can be achieved compared to sequential optimal single-load retrievals.

Chapter 3 studies a correlated storage policy, which assigns product pairs that are ordered frequently together to the same storage bin to save retrieval time. This assignment policy considers information on both turnover frequency of products and correlation between them. In the literature, models use a sequential approach that first cluster correlated products, and then assigns the clusters to storage zones. These models are suboptimal because the objectives of the decomposed problems are to maximize the total product correlation in clusters and then minimize the order picking time. On the other hand, in an integrated approach, the objective is to minimize the order picking time by simultaneously considering product turnover and correlation. Furthermore, current models are developed for manual order picking and are not directly applicable to automated storage and retrieval systems, where each cluster is assigned to a storage pod that is retrieved automatically. The main research question in this chapter is "*How does integrated clustering and storage assignment of correlated products affect the order picking performance in automated warehouses?*" To answer this question, an integrated mathematical model is developed that clusters the products and assigns the cluster to storage locations in order to minimize the total order picking retrieval time. The model is tested for two types of automated systems: crane-based automated storage and retrieval (AS/R) systems and robotic mobile fulfillment (RMF) systems. The performance of the integrated model is evaluated by comparing it to product turnover-based assignment and sequential correlated assignment. The model is solved using Gurobi 7.5. The numerical analysis shows that applying the integrated model, saves up to 40% and 26%, respectively, on retrieval time compared to the benchmark policies.

Chapter 4 studies a dispersed correlated storage policy, which clusters correlated products and, in addition, allows the inventory of each product to be broken up and spread over the forward storage area. Several papers have studied the benefits of random dispersion (Onal *et al.*, 2017, 2018) and evenly spreading (Weidinger and Boysen, 2018) product inventory, but not dispersion considering historical order information. RMF systems, that use robots to move storage pods, are good candidates for implementing such policies for the reason that each pod has several compartments, each providing space for part of product inventory. RMF systems, additionally, are good candidates for the correlated assignment because each storage pod can carry multiple correlated products, which can be retrieved to pick order lines requesting those products. The main research questions in this chapter are "*What is the effect of product dispersion and storage clustering on the expected order picking retrieval time in RMF systems? How do product correlation and product turnover frequency contribute to the performance of the policies?*" to answer this, we develop travel time expressions for different warehouse layouts in robotic mobile fulfillment systems. A mixed-integer program is presented to disperse the inventory, cluster correlated products on pods, and assign the inventory to customer orders. The objective function of the model minimizes the total expected retrieval time of picking all orders. The performance of the dispersed correlated assignment (CDA) policy is compared with random, class-based, correlated and dispersed assignments using a real warehouse dataset. The results show significant benefits of using the CDA policy. Further numerical analysis reveals that a more skewed product correlation (Pareto curve) results in higher performance of the CDA model.

5.2. Outlook

Developments in automation and robotic technology are moving fast. This suggests an increasing need for advanced storage and retrieval policies to control such systems. This section gives an overview of research directions on storage assignment and retrieval methods in compact storage robotized systems.

Puzzle-based storage systems are very compact with a long throughput time. To reduce the throughput time, in addition to optimal two-load retrieval, chapter 2 studies near-optimal multiple-load retrieval in presence of one open location. Future research should study the retrieval methods of such systems when multiple open locations are available. High space utilization may result in reduced system performance due to sub-optimal storage assignment and additional internal relocation which, in general, elongates the storage and retrieval time. Since the number of open locations has a strong effect on the performance of the system, presence of more open locations may speed up the operations. Future research should particularly consider multiple load retrievals in systems with multiple open locations. Also, note that different variants of these systems exist, e.g., a system that allows 'block' movements, that is all loads is a row or column can move simultaneously. We assume the loads move sequentially. Numerical analysis and simulation models can be used to evaluate the performance of such systems.

Chapter 3 highlights the benefits of using correlated storage assignment in AS/R and RMF systems that can facilitate simultaneous picking of multiple lines of an order from the same retrieved storage unit. In the model presented, products are stored using dedicated storage. A dedicated storage policy requires a system with higher storage capacity and longer replenishment time compared to systems using shared storage allocations such as a class-based storage policy. In addition, this model only considers the assignment problem and does not consider the replenishment. Future research should investigate a 'dynamic' correlated storage assignment that takes into account the changing assortment and is flexible in inventory allocation. Faster and more robust solution approaches are needed to handle real size problems.

Chapter 4 introduces the correlated dispersed assignment (CDA) policy that allows inventory to be spread over multiple locations. Numerical analysis on one case, with a large number of lines in each order, shows considerable benefits of such a policy. Further numerical analysis is needed to support the applicability of

CDA to more general cases, especially where orders have smaller sizes, or the order profile has a different turnover and correlation pattern. Furthermore, this chapter does not consider the extra replenishment effort required due to inventory dispersion which may be addressed in further research. Another research direction is to evaluate and optimize the sequencing and assignment of orders to pick stations so that pick requests at each station include correlated products, which are already assigned to the same storage pod, in order to minimize the retrieval times.

Bibliography

Alfieri, A., M. Cantamessa, A. Monchiero, and F. Montagna. 2012. "Heuristics for puzzle-based storage systems driven by a limited set of automated guided vehicles." Journal of Intelligent Manufacturing 23 (5): 1695-1705.

Amirhosseini, M. M. and Sharp, G. P. (1996) 'Simultaneous Analysis of Products and Orders in Storage Assignment', in. American Society of Mechanical Engineers, p. 803812.

Azadeh, K., De Koster, R. and Roy, D. (2019) 'Robotized and automated warehouse systems: Review and recent developments', Transportation Science, 53(4), pp. 917–945. doi: 10.1287/trsc.2018.0873.

Ballou, R. H. (2004) Business logistics/supply chain management : planning, organizing, and controlling the supply chain. 5th edn. Pearson Prentice Hall.

Bartholdi, J. J., and S. T. Hackman (2016) "Warehouse & Distribution Science." v. 0.97, Georgia Institute of Technology, available online at: www.warehouse-science.com (accessed on: 10 Nov. 2016).

Bender, P. S. (1981) 'Mathematical modeling of the 20/80 rule: Theory and practice', *ournal of Business Logistics*, 2(2), pp. 139–157.

Van den Berg, J. P. and Gademann, A. J. R. M. (2000) 'Simulation study of an automated storage/retrieval system', International Journal of Production Research, 38(6), pp. 1339–1356. doi: 10.1080/002075400188889.

Bindi, F. et al. (2009) 'Similarity-based storage allocation rules in an order picking system: An application to the food service industry', International Journal of Logistics Research and Applications, 12(4), pp. 233–247. doi: 10.1080/13675560903075943.

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

Bozer, Y. A. and White, J. A. (1984) 'Travel-Time Models for Automated Storage/Retrieval Systems', IIE Transactions, 16(4), pp. 329–338. doi: 10.1080/07408178408975252.

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

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

Coyle, J. J., Bardi, E. J. and Langley, C. J. (1996) The management of business logistics. 6th edn. West Pub. Co..

Drury, J. (1988) Towards more efficient order picking, 2nd edt.Institute of

Materials Management.

Eweco (2013) “Space parking optimization technology (SPOT).” Video available online at: https://www.youtube.com/watch?v=NozbyQJAnlQ (Accessed 2017).

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

Flake, G.W., and E. B. Baum (2002) “Rush Hour is PSPACE-complete, or why you should generously tip parking lot attendants.” Theoretical Computer Science 270 (1–2): 895–911.

Frazelle, E. H. (1989) Stock location assignment and order picking productivity. PhD thesis, Georgia Institute of Technology.

Garey, M. R. and 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. PhD thesis, Georgia Institute of Technology.

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

Graves, S. C., Hausman, W. H. and Schwarz, L. B. (1977) ‘Storage-Retrieval Interleaving in Automatic Warehousing Systems’, Management Science, 23(9), pp. 935–945. doi: 10.1287/mnsc.23.9.935.

Gue, K.R., K. Furmans, Z. Seibold, and O.Uludag (2014) “Gridstore: a puzzle-based storage system with decentralized control.” Automation Science and Engineering, 11(2): 429 – 438.

Gue, K. R., and B. S. Kim (2007) “Puzzle-based storage systems.” Naval Research Logistics 54 (5): 556–567.

Hansen, P. and Jaumard, B. (1997) ‘Cluster analysis and mathematical programming’, Journal of the American Statistical Association, 79(1–3), pp. 191–215. doi: 10.1007/BF02614317.

Hausman, W. H., Schwarz, L. B. and Graves, S. C. (1976) ‘Optimal Storage Assignment in Automatic Warehousing Systems’, Management Science, 22(6), pp. 629–638. doi: 10.1287/mnsc.22.6.629.

Hearn, R. A. , and E. D. Demaine (2005) “PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation.” Theoretical Computer Science 343 (1-2): 72 – 96.

Heskett, J. L. (1963) ‘Cube-per-order index - a key to warehouse stock location’, Transport and Distribution Managemen, 3, pp. 27–31.

Heskett, J. L. (1964) ‘Putting the cube-per-order index to work in warehouse layout’, Transportation and Distribution Management, 4(8), pp. 23–30.

Hwanga, H., C. Kim, and K. Ko (1999) “Performance analysis of carousel systems with double shuttle.” Computers & Industrial Engineering 36 (2): 473-485.

Jane, C. C. and Laih, Y. W. (2005) ‘A clustering algorithm for item assignment in a synchronized zone order picking system’, European Journal of Operational Research, 166(2), pp. 489–496. doi: 10.1016/j.ejor.2004.01.042.

Jans, R. (2009) ‘Solving lot-sizing problems on parallel identical machines using symmetry-breaking constraints’, INFORMS Journal on Computing, 21(1), pp. 123–136. doi: 10.1287/ijoc.1080.0283.

De Koster, R. B. M., Le-Duc, T. and Roodbergen, K. J. (2007) ‘Design and control of warehouse order picking: A literature review', European Journal of Operational Research, (182), pp. 481–501. doi: 10.1016/j.ejor.2006.07.009.

De Koster, R. B. M., Le-Duc, T. and Yu, Y. (2008) 'Optimal storage rack design for a 3-dimensional compact AS/RS', International Journal of Production Research, 46(6), pp. 1495–1514. doi: 10.1080/00207540600957795.

Kota, R. V., D. Taylor, and K. R. Gue (2015) "Retrieval time performance in puzzle-based storage systems." Journal of Manufacturing Technology Management 26 (4): 582 – 602.

Lamballais, T., Roy, D. and De Koster, M. B. M. (2017) 'Estimating performance in a Robotic Mobile Fulfillment System', *European Journal of Operational Research*, 256(3), pp. 976–990. doi: 10.1016/j.ejor.2016.06.063.

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

Lee, H. F. 1997. "Performance analysis for automated storage and retrieval systems." IIE Transactions 29: 15-28.

Li, J., Moghaddam, M. and Nof, S. Y. (2016) 'Dynamic storage assignment with product affinity and ABC classification—a case study', International Journal of Advanced Manufacturing Technology, 84(9–12), pp. 2179–2194. doi: 10.1007/s00170-015-7806-7.

Litvak, N. (2006) "Optimal picking of large orders in carousel systems." Operations Research Letters 34 (2): 219 – 227

Liu, C. M. (1999) 'Clustering techniques for stock location and order-picking in a distribution center', Computers and Operations Research, 26(10–11), pp.

989–1002. doi: 10.1016/S0305-0548(99)00026-X.

Malmborg, C. J. and Bhaskaran, K. (1987) ‘On the optimality of the cube per order index for conventional warehouses with dual command cycles’.

Malmborg, C. J. and Bhaskaran, K. (1990) ‘A revised proof of optimality for the cube-per-order index rule for stored item location’, Applied Mathematical Modelling, 14(2), pp. 87–95. doi: 10.1016/0307-904X(90)90076-H.

Mirzaei, M., De Koster, R. B. M. and Zaerpour, N. (2017) ‘Modelling load retrievals in puzzle-based storage systems’, International Journal of Production Research, 55(21), pp. 6423–6435. doi: 10.1080/00207543.2017.1304660.

Mirzaei, M., Zaerpour, N. and Koster, R. De (2020) The Effect of Integrated Cluster-Based Storage Assignment on Warehouse Performance. Rotterdam.

Multi-Story Car Park (2016) “Multi storey car parks.” Accessed November 20, 2016, http://www.multi-storey-car-parks.com.

Onal, S., Zhang, J. and Das, S. (2017) ‘Modelling and performance evaluation of explosive storage policies in internet fulfilment warehouses’, International Journal of Production Research, 55(20), pp. 5902–5915. doi: 10.1080/00207543.2017.1304663.

Onal, S., Zhang, J. and Das, S. (2018) ‘Product flows and decision models in Internet fulfillment warehouses’, Production Planning and Control, 29(10), pp. 791–801. doi: 10.1080/09537287.2018.1469800.

Petersen, C. G., Aase, G. R. and Heiser, D. R. (2004) ‘Improving order-picking performance through the implementation of class-based storage’, International Journal of Physical Distribution & Logistics Management, 34(7), pp. 534–544. doi: 10.1108/09600030410552230.

Rao, M. R. (1971) ‘Cluster Analysis and Mathematical Programming’, Journal of the American Statistical Association, 66(335), p. 622. doi: 10.2307/2283542.

Roodbergen, K. J. and Vis, I. F. A. (2009) ‘A survey of literature on automated storage and retrieval systems’, European Journal of Operational Research, 194(2), pp. 343–362. doi: 10.1016/j.ejor.2008.01.038.

Roodbergen, K. J., and R. De Koster (2001) “Routing methods for warehouses with multiple cross aisles.” International Journal of Production Research 39(9): 1865-1883.

Sadiq, M., Landers, T. L. and Taylor, G. D. (1996) ‘An Assignment Algorithm for Dynamic Picking Systems’, IIE Transactions, 28(8), pp. 607–616. doi: 10.1080/15458830.1996.11770706.

Sari, Z., C. Saygin, N. Ghouali (2005) “Travel-time models for flow-rack automated storage and retrieval systems.” International Journal of Advanced Manufacturing Technology 25: 979-987.

Sharp, G., Amirhosseini, M. M. and 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.

Tompkins, J. A. et al. (2010) “Facilities Planning.” New York: John Wiley & Sons, Inc., 2nd edition.

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

Weidinger, F. and Boysen, N. (2018) ‘Scattered Storage: How to Distribute Stock Keeping Units All Around a Mixed-Shelves Warehouse’, Transportation

Science. INFORMS, p. trsc.2017.0779. doi: 10.1287/trsc.2017.0779.

Xiao, J. and Zheng, L. (2010) ‘A correlated storage location assignment problem in a single-block-multi- aisles warehouse considering BOM information’, International Journal of Production Research, 48(5), pp. 1321–1338. doi: 10.1080/00207540802555736.

Xiao, J. and Zheng, L. (2012) ‘Correlated storage assignment to minimize zone visits for BOM picking’, International Journal of Advanced Manufacturing Technology, 61(5–8), pp. 797–807. doi: 10.1007/s00170-011-3740-5.

Yang, J. H., and K. H. Kim (2006) “A grouped storage method for minimizing relocations in block stacking systems.” Journal of Intelligent Manufacturing 17: 453–463.

Yu, Y., De Koster, R. B. M. and Guo, X. (2015) ‘Class-Based Storage with a Finite Number of Items: Using More Classes is not Always Better’, Production and Operations Management, 24(8), pp. 1235–1247. doi: 10.1111/poms.12334.

Yu, Y., R. De Koster (2012) “Sequencing heuristics for storing and retrieving unit loads in 3D compact automated warehousing systems.” IIE Transactions 44(2): 69-87.

Yu, Y., R. De Koster (2009) “Designing an optimal turnover-based storage rack for a 3D compact automated storage and retrieval system.” International Journal of Production Research 47(6): 1551-1571.

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

Zaerpour, N., Yu, Y. and de Koster, R. B. M. (2015) ‘Storing Fresh Produce for

Fast Retrieval in an Automated Compact Cross-Dock System', Production and Operations Management, 24(8), pp. 1266–1284. doi: 10.1111/poms.12321.

Zaerpour, N., Yu, Y. and de Koster, R. B. M. (2017) 'Optimal two-class-based storage in a live-cube compact storage system', IISE Transactions, 49(7), pp. 653–668. doi: 10.1080/24725854.2016.1273564.

Zaerpour, N., Yu, Y. and de Koster, R. (2017) 'Small is Beautiful: A Framework for Evaluating and Optimizing Live-Cube Compact Storage Systems', *Transportation Science*, 51(1), pp. 34–51. doi: 10.1287/trsc.2015.0586.

Zhang, Y. (2016) 'Correlated Storage Assignment Strategy to reduce Travel Distance in Order Picking', IFAC-PapersOnLine. Elsevier B.V., 49(2), pp. 30–35. doi: 10.1016/j.ifacol.2016.03.006.

Zou, B., De Koster, R. and Xu, X. (2018) 'Operating policies in robotic compact storage and retrieval systems', Transportation Science, 52(4), pp. 788–811. doi: 10.1287/trsc.2017.0786.

About the Author

Masoud Mirzaei was born on 20 March 1985 in Arak, Iran. He received his BSc and MSc degrees in Industrial Engineering from Iran University of Science and Technology (2007) and Shahed University (2010), respectively, in Tehran, Iran. For two years, he worked as operations and data analyst in industry, before becoming a PhD candidate at Rotterdam School of Management, Erasmus University Rotterdam in February 2013. In 2013, he was a visiting scholar at the Department of Information Management and Decision Science, University of Science and Technology of China, Hefei, China. His research interests include supply chain management, design and analysis of warehouses and fulfillment centers, logistics and material handling, operations research. His research findings has been presented in many international conferences including INFORMS Annual Meeting (2015), POMS (2012) and European Conference on Operational Research (2018, 2019). He is involved in coordinating and teaching several courses, and supervising students at master and bachelor levels. Masoud is currently a Postdoctoral fellow at Department of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology.

Portfolio

Publications

JOURNAL PUBLICATIONS

Mirzaei, Masoud; De Koster, René; Zaerpour, Nima: Modelling Load Retrievals in Puzzle-based Storage Systems, International Journal of Production Research 55: 6423-6435 (2017). DOI: 10.1080/00207543.2017.1304660.

Bashiri, Mahdi; Mirzaei, Masoud, Randall, Marcus: Modeling Fuzzy Capacitated *p*-hub Center Problem and a Genetic Algorithm Solution, Applied Mathematical Modelling 37: 3513–3525 (2013). DOI: 10.1016/j.apm.2012.07.018.

WORKING PAPERS

Mirzaei, Masoud; Zaerpour, Nima; De Koster, René: The Impact of Integrated Cluster-based Storage Assignment on Warehouse Performance, (*submitted).*

Mirzaei, Masoud; Zaerpour, Nima; De Koster, René: Correlated Dispersed Storage Assignment in Robotic Warehouses.

Mirzaei, Masoud: Cycle Time Models for Load Retrievals in High Density Automated Storage Systems.

BOOK

Mirzaei, Masoud; Bashiri, Mahdi: Ant Colony Optimization: Concepts, Algorithms and Programming, Tehran, the Commerce Printing and Publishing Company (2010) - *in Persian.*

PROCEEDINGS

Mirzaei, Masoud; Bashiri, Mahdi: Multiple Objective Multiple Allocation Hub Location Problem, The 40th International Conference on Computers & Industrial Engineering, Awaji, Japan, IEEE (2010). DOI: 10.1109/ICCIE.2010.5668249.

PhD Courses

Facility Logistics Management
Data-analysis and Statistics
Non-cooperative Games
Algorithmic Game Theory
Inventory Management in Supply Chains
Theories and methods

Writing a Literature Review
Writing and Publishing a Research Article
Operations Research in Healthcare
Innovations in Transport and Logistics
Fundamental Domain Knowledge

Teaching

COORDINATOR AND/OR LECURER

Quantitative Methods and Techniques: Mathematics, BAP053 (2018 – 2019)
Supply Chain Simulation, BMME069 (2013 – 2019)
Research Training and Bachelor Thesis, BAD10 and BKBBTH (2016 – 2019)

TUTURIAL LECURER

Global Supply Chain Management, BM-IM03CC (2013 – 2018)
Operations Management (2014 – 2016)
Facility Logistics Management, BM04SCM (2015 – 2016)

Conferences Attended

European Conference on Operational Research, Valencia (2018) and Dublin (2019).
International Conference on Logistics and Maritime Systems, Rotterdam (2014) and Singapore (2019).
Odysseus, Cagliari, Italy (2018).
INFORMS Annual Meeting, Nashville, TN, USA (2016).
POMS Annual Conference 2016, Orlando, FL, USA (2016).
CEMS Research Seminar on supply chain management, Riezlern, Austria (2015,18, 19).
European Logistics Association Doctorate Workshop, Wroclaw, Poland (2017).
TRAIL PhD congress, Utrecht, Netherlands (2016).
International Conference on Operations Research, Rotterdam, the Netherlands (2013).

Summary

Warehouses are key components in supply chain. They facilitate the product flow from production to distribution. The performance of supply chains relies on the performance of warehouses and distribution centers. Being able to realize short order delivery lead times, in retail and ecommerce particularly, is important for warehouses. Efficient and responsive storage and retrieval operations can help in realizing a short order delivery lead time. Additionally, space scarcity has brought some companies to use high-density storage systems that increase space usage in the warehouse. In such storage systems, most of the available space is used for storing products, as little space is needed for transporting loads. However, the throughput capacity of high-density storage systems is typically low. New robotic and automated technologies help warehouses to increase their throughput and responsiveness. Warehouses adapting such technologies require customized storage and retrieval policies fit for automated operations. This thesis studies storage and retrieval policies in warehouses using several common and emerging automated technologies.

Chapter 2 studies puzzle-based storage systems, in which loads are stored on transport shuttles, which carry the unit loads autonomously in a high-density storage

system. Loads are stored next to each other. The system does not have transport aisles. Only few open locations are available for the shuttles to move. The system shuffles loads consecutively to make space available to retrieve a requested load. As such, it resembles the traditional 15-tile sliding puzzle. The system has been studied in literature, in particular how to retrieve one load at a time. This chapter proposes a multiple-load retrieval method that brings two or more requested loads together to an optimal joining location, and then retrieves them simultaneously. Closed form expressions are derived for the number of moves required to retrieve multiple loads. A fast heuristic is developed to find near-optimal joining locations for the loads. Numerical analysis shows that multiple-load retrieval results in a shorter retrieval time than optimal sequential single-load retrieval.

Chapter 3 studies the impact of correlated storage assignment on order retrieval time in automated storage and retrieval (AS/R) systems. In an AS/R system, automated cranes move within narrow aisles and transport storage bins between storage shelves and pick stations. The assignment of products to storage bins impacts the order retrieval time, especially multiple line orders. A correlated storage policy groups product that appear jointly in customer orders frequently in product clusters. These clusters are then assigned to storage bins. Each bin has multiple compartments that can house the products of a cluster. This correlated policy can reduce the number of bin retrievals required to pick all the lines of an order. In this chapter, an integrated linear program is formulated that clusters the correlated products and assigns the clusters to storage bins to minimize the total retrieval time. The numerical analysis shows that it performs better than sequentially clustering products and assigning clusters to storage bins. Additionally, it outperforms the turnover frequency-based assignment when the skewness of the Pareto curve of the turnover frequency of the products in the assortment is low to moderate, for even low correlation between products.

Chapter 4 extends the correlated storage policy of the previous chapter by including dispersion of product inventory. In such a case, the inventory of each product is split and dispersed over multiple storage locations. Dispersion makes each product more accessible from different pick stations. A product can now also be stored in multiple product clusters, depending on its correlation with other products. This can reduce the order picking retrieval time, especially for robotic mobile fulfilment systems where autonomous robots move the storage pods carrying inventory of multiple products to a pick station and return it after the customer order has been picked. This chapter formulates a integer linear program for correlated dispersed storage in which products are assigned to storage pods, storage pods are assigned to storage locations and the inventory is assigned to customer orders to minimize the total retrieval time. Retrieval time expressions are developed for different layouts of the warehouse. Since the model is NP-hard, a simple and efficient heuristic is developed that is capable to solve real size problems. To evaluate the performance of the model, we apply it to a dataset consisting of three-month order history of a warehouse in personal care products. The outcome is compared with that of random, class-based, correlated but not dispersed, and dispersed but not correlated storage policies. The results show that the correlated dispersed storage outperforms the other policies for the instances tested. We use regression models to predict the performance of the policies based on correlation and turnover frequency Pareto curves. The results show significant association between the total retrieval time and the skewness of the correlation Pareto curve in the correlated dispersed policy for the case tested.

Samenvatting (Summary in Dutch)

Magazijnen zijn een belangrijk onderdeel van de supply chain. Ze ondersteunen de productstroom van productie naar distributie. De prestatie van een supply chain hangt af van de prestaties van magazijnen en distributiecentra. Het realiseren van korte levertijden is belangrijk voor magazijnen, met name in de detailhandel en e-commerce. Efficiënte en responsieve opslag- en orderverzamelprocessen kunnen helpen bij het realiseren van een korte levertijd van bestellingen. Gebrek aan beschikbare ruimte heeft sommige bedrijven ertoe gebracht zeer compacte opslagsystemen te gebruiken die de ruimte in het magazijn maximaal benutten. In dergelijke opslagsystemen wordt het grootste deel van de beschikbare ruimte gebruikt voor het opslaan van producten, aangezien er weinig ruimte nodig is voor het transport van ladingen. De doorzet van compacte opslagsystemen is echter laag. Nieuwe robot- en geautomatiseerde technologieën helpen magazijnen hun doorzet en responsiviteit te vergroten. Dergelijke magazijnen vereisen bedrijfsspecifieke in- en uitslagstrategieën die geschikt zijn voor geautomatiseerde processen. Dit proefschrift bestudeert in- en uitslagstrategieën in magazijnen voor verschillende veelgebruikte en nieuwe geautomatiseerde technologieën.

Hoofdstuk 2 bestudeert ‘puzzelgebaseerde’ compacte opslagsystemen, waarin ladingen worden opgeslagen op shuttles, die zich autonoom kunnen verplaatsen in een compact opslagsysteem. Ladingen worden dicht naast elkaar opgeslagen. Het systeem heeft geen transportgangen. Om de shuttles zichzelf te laten verplaatsen is een beperkt aantal open locaties beschikbaar. Het systeem verplaatst achtereenvolgende ladingen zodat er ruimte ontstaat om een gevraagde lading op te halen. Dit systeem heeft wat weg van de bekende schuifpuzzel met 15 tegels. Een dergelijk systeem is al eerder onderzocht in de literatuur, met name hoe één lading tegelijk kan worden opgehaald. In dit hoofdstuk wordt een ophaalmethode voorgesteld voor meerdere ladingen, waarbij twee of meer gevraagde ladingen telkens naar een optimale verbindingslocatie worden gebracht en vandaar gezamenlijk worden opgehaald. Analytische uitdrukkingen worden afgeleid voor het aantal bewegingen dat nodig is om meerdere ladingen op te halen en een snelle heuristiek wordt ontwikkeld om een nagenoeg optimale verbindingslocaties voor de ladingen te vinden. Numerieke analyse toont aan dat het gecombineerd ophalen van meerdere ladingen resulteert in een kortere ophaaltijd dan het optimale sequentiële ophaalplan met één lading per keer.

In Hoofdstuk 3 bestudeert de impact van gecorreleerde opslagtoewijzing op de benodigde uitslagrijtijd voor het ophalen van ladingen in geautomatiseerde opslagsystemen (zogeheten automated storage and retrieval, AS/R, systemen). In een AS/R-systeem verplaatsen geautomatiseerde kranen zich in smalle gangpaden en transporteren ze opslagladingen (bakken of pallets) tussen opslaglocaties en pickstations. De toewijzing van producten aan opslagbakken en opslagbakken aan locaties heeft invloed op de benodigde tijd voor het ophalen van orders, met name orders bestaande uit meerdere regels. Een (vraag)gecorreleerde opslagstrategie groepeert producten die vaak gezamenlijk voorkomen in klantorders, in productclusters. Deze clusters worden vervolgens toegewezen aan opslagbakken. Elke bak heeft meerdere compartimenten waarin de verschillende producten van een cluster kunnen worden ondergebracht. Deze gecorreleerde opslagstrategie kan het aantal ophaalopdrachten verminderen dat nodig is om alle regels van een bestelling te verzamelen. In dit hoofdstuk wordt een geïntegreerd lineair programma geformuleerd dat de gecorreleerde producten clustert en de clusters toewijst aan opslagbakken om de totale ophaaltijd te minimaliseren. Uit de numerieke analyse blijkt dat deze methode beter presteert dan sequentiële clustering van producten en het vervolgens toewijzen van clusters aan opslagbakken. Bovendien overtreft het de op omloopsnelheid gebaseerde toewijzing wanneer de scheefheid (skewness) van de Pareto-curve van de omzetfrequentie van de producten in het assortiment laag tot matig is, zelfs als de vraagcorrelatie tussen producten laag is.

Hoofdstuk 4 breidt het gecorreleerde opslagbeleid van het vorige hoofdstuk uit met de verspreiding van productvoorraad. In een dergelijk geval wordt de voorraad van elk product opgesplitst en verspreid over meerdere opslaglocaties. Dispersie maakt een product sneller toegankelijk vanaf verschillende startposities. Een product kan nu ook worden opgeslagen in meerdere productclusters, afhankelijk van de correlatie met andere producten. Dit kan de ophaaltijd voor het orderverzamelen verminderen, vooral voor mobiele fulfilment-systemen waarbij autonome robots de opslagrekken met inventaris van meerdere producten naar een pickstation verplaatsen en retourneren nadat de klantbestelling is verzameld. Dit hoofdstuk formuleert een lineair geheeltallig programma voor gecorreleerde verspreide opslag waarin producten worden toegewezen aan opslagrekken,

opslagrekken worden toegewezen aan opslaglocaties en de productvoorraad wordt toegewezen aan klantorders om de totale ophaaltijd te minimaliseren. Voor verschillende indelingen van het magazijn worden uitdrukkingen ontwikkeld voor de uitslagtijd. Aangezien het model NP-hard is, wordt een eenvoudige en efficiënte heuristiek ontwikkeld die in staat is om problemen van praktische grootte op te lossen. Om de prestaties van het model te evalueren, wordt het toegepast op een dataset bestaande uit een bestelgeschiedenis van drie maanden van een magazijn met producten voor persoonlijke verzorging. Het resultaat wordt vergeleken met een random opslagstrategie, een strategie gebaseerd op opslagklassen, een gecorreleerde, maar niet verspreide strategie, en een verspreide, maar niet gecorreleerd opslagstrategie. De resultaten laten zien dat voor de geteste instanties de gecorreleerde en verspreide opslag beter presteert dan de andere strategieën. We gebruiken regressiemodellen om de prestaties van het beleid te voorspellen op basis van correlatie- en omzetfrequentie Pareto-curven. De resultaten tonen voor de geteste casus een significant verband aan tussen de totale uitslagtijd en de scheefheid van de correlatie Pareto-curve in de gecorreleerde verspreide strategie.

چکیده (Summary in Farsi)

انبارها از بخش‌های کلیدی در زنجیره تامین محسوب می‌شوند. آنها جریان کالا از تولید تا توزیع را تسهیل می‌کنند. عملکرد زنجیره تامین وابسته به عملکرد انبارها و مراکز توزیع است. کوتاه کردن زمان تحویل سفارش، به‌ویژه در خرده‌فروشی و فروش اینترنتی، برای انبارها حائز اهمیت است. عملیات انبارش و بازیابی کارامد و پاسخگو، مسیر دستیابی به زمان کوتاه تحویل سفارش مشتری را هموار می‌کند. افزون بر این، کمبود فضا برخی از شرکت‌ها را به استفاده از سامانه‌های انبارش متراکم سوق داده است تا از فضای موجود استفاده‌ی بهتری نمایند. در این نوع از سامانه‌های انبارش، بخش عمده‌ی فضای موجود برای انبارش کالا استفاده می‌گردد، زیرا به فضای اندکی برای جابجایی کالا نیاز است. با این وجود، بهره‌وری سامانه‌های انبارش متراکم پایین است. فن‌آوری‌های نوین خودکار و رباتیک به انبارها کمک می‌کنند تا بهره‌وری و سطح پاسخ‌گویی خود را افزایش دهند. انبارهایی که از این‌گونه فناوری‌ها استفاده می‌کنند نیازمند راهبردهای انبارش و بازیابی جدیدی هستند. این پایان‌نامه به مطالعه راهبردهای انبارش و بازیابی در انبارهایی که از فن‌آوری‌های خودکار نوظهور یا رایج بهره می‌برند می‌پردازد.

فصل ۲ به بررسی سامانه‌های «انبارش متراکم مبتنی بر پازل» می‌پردازد که در آن‌ها کالا بر روی شاتل‌های نقلیه قرارداده شده که قابلیت جابجایی خودکار کالاها در سامانه انبارش متراکم را دارند. این سامانه فاقد راهروهای جابجایی کالا است. تنها تعداد محدودی فضای خالی برای جابجایی شاتل‌ها در دسترس می‌باشد. سامانه با حرکت دادن مداوم کالاها به اطراف فضای مورد نیاز برای بازیابی کالای درخواستی را فراهم می‌کند. این کارکرد، مشابه بازی قدیمی پازل ۱۵تایی است که در آن با جابجایی کاشی‌های متحرک شامل اعداد یک تا پانزده، ترتیب صعودی از اعداد ایجاد می‌گردد. پژوهش‌های پیشین به مطالعه این سامانه به ویژه روش‌های بازیابی انفرادی کالا پرداخته‌اند. این فصل یک روش برای بازیابی دو یا تعداد بیشتری کالا به صورت همزمان ارائه می‌دهد. در این روش ابتدا کالاهای درخواستی جابجا می‌شوند تا در یک «مکان همگرایی بهینه» کنار یکدیگر قرار بگیرند، سپس همزمان به محل نهایی استخراج کالا انتقال داده می‌شوند. معادلات ریاضی فرم بسته برای محاسبه تعداد حرکت‌های مورد نیاز برای بازیابی چند کالا ارائه شده است. یک روش ابداعی سریع برای یافتن مکان همگرایی نزدیک به بهینه نیز توسعه داده شده است. تحلیل‌های عددی نشان می‌دهند بازیابی چند کالا منجر به زمان بازیابی کوتاه‌تری در مقایسه با بازیابی انفرادی می‌گردد.

فصل ۳ به بررسی تاثیر «تخصیص موجودی همبسته» بر بازیابی سفارش در سامانه‌های خودکار انبارش و بازیابی می‌پردازد. در این سامانه‌ها، جرثقیل‌های خودکار در میان راهروهای باریک حرکت کرده و قفسه‌های انبارش را مابین قفسه‌ها و ایستگاه آماده‌سازی سفارش جابجا می کنند. نحوه تخصیص کالاها به قفسه‌های انبارش تاثیر بسزایی روی زمان بازیابی سفارش دارد، به ویژه در سفارش‌های مشتمل بر چند کالا. تخصیص موجودی همبسته، کالاهایی را که مکررا در سفارش‌های مشتری با هم دیده می‌شوند، در گروه‌های مجزا دسته‌بندی می‌کند. سپس، این دسته‌های کالا به قفسه‌های انبارش تخصیص داده می‌شوند. هر قفسه متشکل از چندین محفظه بوده که می‌تواند کالاهای یک دسته را در خود جای دهد. این تخصیص همبسته می‌تواند تعداد دفعات بازیابی که برای آماده‌سازی اقلام موجود در سفارش مشتری مورد نیاز است را کاهش دهد. در این فصل

یک برنامه‌ریزی خطی یکپارچه برای دسته‌بندی کالاهای همبسته و تخصیص دسته‌ها به جعبه‌های انبارش ارائه شده که کل زمان بازیابی را کمینه می‌کند. تحلیل‌های عددی نشان می‌دهد مدل یکپارچه عملکردی بهتری از دسته‌بندی کالا و تخصیص دسته‌ها به جعبه‌های انبارش به صورت متوالی دارد. به علاوه، در مواردی که چولگی منحنی پارتو فراوانی سفارش کم تا میانه باشد، این مدل عملکرد بهتری نسبت به تخصیص مبتنی بر فراوانی سفارش کالا دارد، حتی در شرایط همبستگی پایین بین کالاها.

فصل ۴ راهبرد تخصیص همبسته در فصل گذشته را توسعه داده تا پراکندگی موجودی کالا را نیز در نظر بگیرد. در این حالت، موجودی هر کالا تقسیم شده و در چندین مکان انبارش می‌شود. پراکندگی، دستیابی به کالاها را برای ایستگاه‌های مختلف آماده‌سازی سفارش مشتری آسان‌تر می‌کند. همچنین یک کالا با توجه به همبستگی آن با دیگر کالاها می‌تواند در چندین گروه کالایی دسته‌بندی شود. این روش می‌تواند زمان بازیابی سفارش مشتری را کاهش دهد، به ویژه در سامانه‌ی «رباتیک سیار تکمیل سفارش» که در آن ربات‌های خودکار قفسه‌های انبارش را جابجا می‌کنند که حاوی موجودی چندین کالا می‌باشد. ربات‌ها قفسه‌ها را به ایستگاه آماده‌سازی سفارش مشتری برده و پس از بازیابی اقلام مورد نیاز به فضای انبار بازمی‌گردانند. این فصل یک برنامه‌ریزی خطی عدد صحیح برای تخصیص همبسته پراکنده ارایه می‌دهد که کالاها را به قفسه‌های انبارش، قفسه‌های انبارش را به مکان‌های انبارش و موجودی را به سفارش مشتری تخصیص می‌دهد تا کل زمان بازیابی کالا کمینه گردد. معادلات زمان بازیابی کالا برای جانمایی‌های مختلف انبار توسعه داده شده است. به دلیل پیچیده (NP-hard) بودن مساله، یک روش ابداعی ساده و کارامد توسعه داده شده که قادر به حل مساله در ابعاد واقعی می‌باشد. ما این روش را بر روی داده‌های سه‌ماهه انباری از کالاهای بهداشت شخصی اجرا کردیم تا کارایی آن را ارزیابی کنیم. نتایج حاصله با راهبردهای تخصیص تصادفی، مبتنی بر دسته‌بندی، همبسته بدون پراکندگی و پراکنده بدون همبستگی مقایسه شده است. یافته‌ها حاکی از عملکرد بهتر راهبرد تخصیص همبسته پراکنده برای داده‌های موجود است. ما از مدل‌های رگرسیون برای پیش‌بینی عملکرد راهبردهای

مختلف از روی منحنی پارتو همبستگی و فراوانی سفارش استفاده کردیم. نتایج نشان‌گر وابستگی معنی‌داری بین کل زمان بازیابی و چولگی منحنی پارتو همبستگی در راهبرد تخصیص همبسته پراکنده برای داده‌های موجود می‌باشد.

The ERIM PhD Series

The ERIM PhD Series contains PhD dissertations in the field of Research in Management defended at Erasmus University Rotterdam and supervised by senior researchers affiliated to the Erasmus Research Institute of Management (ERIM). All dissertations in the ERIM PhD Series are available in full text through the ERIM Electronic Series Portal: http://repub.eur.nl/pub. ERIM is the joint research institute of the Rotterdam School of Management (RSM) and the Erasmus School of Economics (ESE) at the Erasmus University Rotterdam (EUR).

Dissertations in the last four years

Ahmadi, S., *A motivational perspective to decision-making and behavior in organizations*, Promotors: Prof. J.J.P. Jansen & Dr T.J.M. Mom, EPS-2019-477-S&E, https://repub.eur.nl/pub/116727

Akemu, O., *Corporate Responses to Social Issues: Essays in Social Entrepreneurship and Corporate Social Responsibility,* Promotors: Prof. G.M. Whiteman & Dr S.P. Kennedy, EPS-2017-392-ORG, https://repub.eur.nl/pub/95768

Albuquerque de Sousa, J.A., *International stock markets: Essays on the determinants and consequences of financial market development*, Promotors: Prof. M.A. van Dijk & Prof. P.A.G. van Bergeijk, EPS-2019-465-F&A, https://repub.eur.nl/pub/115988

Alserda, G.A.G., *Choices in Pension Management*, Promotors: Prof. S.G. van der Lecq & Dr O.W. Steenbeek, EPS-2017-432-F&A, https://repub.eur.nl/pub/103496

Arampatzi, E., *Subjective Well-Being in Times of Crises: Evidence on the Wider Impact of Economic Crises and Turmoil on Subjective Well-Being,* Promotors: Prof. H.R. Commandeur, Prof. F. van Oort & Dr. M.J. Burger, EPS-2018-459-S&E, https://repub.eur.nl/pub/111830

Avci, E., *Surveillance of Complex Auction Markets: a Market Policy Analytics Approach*, Promotors: Prof. W. Ketter, Prof. H.W.G.M. van Heck & Prof. D.W. Bunn, EPS-2018-426-LIS, https://repub.eur.nl/pub/106286

Balen, T.H. van, *Challenges of Early Stage Entrepreneurs: the Roles of Vision Communication and Team Membership Change*,
Promotors: Prof. J.C.M. van den Ende & Dr M. Tarakci, EPS-2019-468-LIS, https://repub.eur.nl/pub/115654

Bernoster, I., *Essays at the Intersection of Psychology, Biology, and Entrepreneurship,* Promotors: Prof. A.R. Thurik, Prof. I.H.A. Franken & Prof. P.J.F Groenen, EPS-2018-463-S&E, https://repub.eur.nl/pub/113907

Bouman, P., *Passengers, Crowding and Complexity: Models for Passenger Oriented Public Transport*, Prof. L.G. Kroon, Prof. A. Schöbel & Prof. P.H.M. Vervest, EPS-2017-420-LIS, https://repub.eur.nl/pub/100767

Bunderen, L. van, *Tug-of-War: Why and when teams get embroiled in power struggles*, Promotors: Prof. D.L. van Knippenberg & Dr. L. Greer, EPS-2018-446-ORG, https://repub.eur.nl/pub/105346

Burg, G.J.J. van den, *Algorithms for Multiclass Classification and Regularized Regression*, Promotors: Prof. P.J.F. Groenen & Dr. A. Alfons, EPS-2018-442-MKT, https://repub.eur.nl/pub/103929

Chammas, G., *Portfolio concentration*, Promotor: Prof. J. Spronk, EPS-2017-410-F&E, https://repub.eur.nl/pub/94975

Dalmeijer, K., *Time Window Assignment in Distribution Networks*, Promotors: Prof A.P.M. Wagelmans & Dr R. Spliet, EPS-2019-486-LIS, https://repub.eur.nl/pub/120773

Dennerlein, T. *Empowering Leadership and Employees' Achievement Motivations: the Role of Self-Efficacy and Goal Orientations in the Empowering Leadership Process*, Promotors: Prof. D.L. van Knippenberg & Dr J. Dietz, EPS-2017-414-ORG, https://repub.eur.nl/pub/98438

Dolgova, E., *On Getting Along and Getting Ahead: How Personality Affects Social Network Dynamics, Promotors*: Prof. P.P.M.A.R Heugens & Prof. M.C. Schippers, EPS-2019-455-S&E, https://repub.eur.nl/pub/119150

Duijzer, L.E., *Mathematical Optimization in Vaccine Allocation*,
Promotors: Prof. R. Dekker & Dr W.L. van Jaarsveld, EPS-2017-430-LIS, https://repub.eur.nl/pub/101487

Eijlers, E., *Emotional Experience and Advertising Effectiveness: on the use of EEG in marketing*, Prof. A. Smidts & Prof. M.A.S. Boksem, Eps-2019-487-MKT, https://repub.eur.nl/pub/124053

El Nayal, O.S.A.N., *Firms and the State: An Examination of Corporate Political Activity and the Business-Government Interface,* Promotor: Prof. J. van Oosterhout & Dr. M. van Essen, EPS-2018-469-S&E, https://repub.eur.nl/pub/114683

Feng, Y., *The Effectiveness of Corporate Governance Mechanisms and Leadership Structure: Impacts on strategic change and firm performance*, Promotors: Prof. F.A.J. van den Bosch, Prof. H.W. Volberda & Dr J.S. Sidhu, EPS-2017-389-S&E, https://repub.eur.nl/pub/98470

Frick, T.W., *The Implications of Advertising Personalization for Firms, Consumer, and Ad Platfroms,* Promotors: Prof. T. Li & Prof. H.W.G.M. van Heck, EPS-2018-452-LIS, https://repub.eur.nl/pub/110314

Fytraki, A.T., *Behavioral Effects in Consumer Evaluations of Recommendation Systems*, Promotors: Prof. B.G.C. Dellaert & Prof. T. Li, EPS-2018-427-MKT, https://repub.eur.nl/pub/110457

Ghazizadeh, P. *Empirical Studies on the Role of Financial Information in Asset and Capital Markets,* Promotors: Prof. A. de Jong & Prof. E. Peek, EPS-2019-470-F&A, https://repub.eur.nl/pub/114023

Giurge, L., *A Test of Time; A temporal and dynamic approach to power and ethics*, Promotors: Prof. M.H. van Dijke & Prof. D. De Cremer, EPS-2017-412-ORG, https://repub.eur.nl/pub/98451

Gobena, L., *Towards Integrating Antecedents of Voluntary Tax Compliance*, Promotors: Prof. M.H. van Dijke & Dr P. Verboon, EPS-2017-436-ORG, https://repub.eur.nl/pub/103276

Groot, W.A., *Assessing Asset Pricing Anomalies*, Promotors: Prof. M.J.C.M. Verbeek & Prof. J.H. van Binsbergen, EPS-2017-437-F&A, https://repub.eur.nl/pub/103490

Hanselaar, R.M., *Raising Capital: On pricing, liquidity and incentives,* Promotors: Prof. M.A. van Dijk & Prof. P.G.J. Roosenboom, EPS-2018-429-F&A, https://repub.eur.nl/pub/113274

Harms, J. A., *Essays on the Behavioral Economics of Social Preferences and Bounded Rationality*, Prof. H.R. Commandeur & Dr K.E.H. Maas, EPS-2018-457-S&E, https://repub.eur.nl/pub/108831

Hendriks, G., *Multinational Enterprises and Limits to International Growth: Links between Domestic and Foreign Activities in a Firm's Portfolio,* Promotors: Prof. P.P.M.A.R. Heugens & Dr. A.H.L Slangen, EPS-2019-464-S&E, https://repub.eur.nl/pub/114981

Hengelaar, G.A., *The Proactive Incumbent: Holy grail or hidden gem? Investigating whether the Dutch electricity sector can overcome the incumbent's curse and lead the sustainability transition*, Promotors: Prof. R.J. M. van Tulder & Dr K. Dittrich, EPS-2018-438-ORG, https://repub.eur.nl/pub/102953

Jacobs, B.J.D., *Marketing Analytics for High-Dimensional Assortments*, Promotors: Prof. A.C.D. Donkers & Prof. D. Fok, EPS-2017-445-MKT, https://repub.eur.nl/pub/103497

Jia, F., *The Value of Happiness in Entrepreneurship*, Promotors: Prof. D.L. van Knippenberg & Dr Y. Zhang, EPS-2019-479-ORG, https://repub.eur.nl/pub/115990

Kahlen, M. T., *Virtual Power Plants of Electric Vehicles in Sustainable Smart Electricity Markets*, Promotors: Prof. W. Ketter & Prof. A. Gupta, EPS-2017-431-LIS, https://repub.eur.nl/pub/100844

Kampen, S. van, *The Cross-sectional and Time-series Dynamics of Corporate Finance: Empirical evidence from financially constrained firms*, Promotors: Prof. L. Norden & Prof. P.G.J. Roosenboom, EPS-2018-440-F&A, https://repub.eur.nl/pub/105245

Karali, E., *Investigating Routines and Dynamic Capabilities for Change and Innovation,* Promotors: Prof. H.W. Volberda, Prof. H.R. Commandeur & Dr J.S. Sidhu, EPS-2018-454-S&E, https://repub.eur.nl/pub/106274

Keko. E, *Essays on Innovation Generation in Incumbent Firms*, Promotors: Prof. S. Stremersch & Dr N.M.A. Camacho, EPS-2017-419-MKT, https://repub.eur.nl/pub/100841

Kerkkamp, R.B.O., *Optimisation Models for Supply Chain Coordination under Information Asymmetry,* Promotors: Prof. A.P.M. Wagelmans & Dr. W. van den Heuvel, EPS-2018-462-LIS, https://repub.eur.nl/pub/109770

Khattab, J., *Make Minorities Great Again: a contribution to workplace equity by identifying and addressing constraints and privileges*, Promotors: Prof. D.L. van Knippenberg & Dr A. Nederveen Pieterse, EPS-2017-421-ORG, https://repub.eur.nl/pub/99311

Kim, T. Y., *Data-driven Warehouse Management in Global Supply Chains*, Promotors: Prof. R. Dekker & Dr C. Heij, EPS-2018-449-LIS, https://repub.eur.nl/pub/109103

Klitsie, E.J., *Strategic Renewal in Institutional Contexts: The paradox of embedded agency*, Promotors: Prof. H.W. Volberda & Dr. S. Ansari, EPS-2018-444-S&E, https://repub.eur.nl/pub/106275

Koolen, D., *Market Risks and Strategies in Power Systems Integrating Renewable Energy*, Promotors: Prof. W. Ketter & Prof. R. Huisman, EPS-2019-467-LIS, https://repub.eur.nl/pub/115655

Kong, L. *Essays on Financial Coordination,* Promotors: Prof. M.J.C.M. Verbeek, Dr. D.G.J. Bongaerts & Dr. M.A. van Achter. EPS-2019-433-F&A, https://repub.eur.nl/pub/114516

Kyosev, G.S., *Essays on Factor Investing*, Promotors: Prof. M.J.C.M. Verbeek & Dr J.J. Huij, EPS-2019-474-F&A, https://repub.eur.nl/pub/116463

Lamballais Tessensohn, T., *Optimizing the Performance of Robotic Mobile Fulfillment Systems*, Promotors: Prof. M.B.M de Koster, Prof. R. Dekker & Dr D. Roy, EPS-2019-411-LIS, https://repub.eur.nl/pub/116477

Leung, W.L., *How Technology Shapes Consumption: Implications for Identity and Judgement*, Promotors: Prof. S. Puntoni & Dr G Paolacci, EPS-2019-485-MKT, https://repub.eur.nl/pub/117432

Li, X. *Dynamic Decision Making under Supply Chain Competition,* Promotors: Prof. M.B.M de Koster, Prof. R. Dekker & Prof. R. Zuidwijk, EPS-2018-466-LIS, https://repub.eur.nl/pub/114028

Liu, N., *Behavioral Biases in Interpersonal Contexts*, Supervisors: Prof. A. Baillon & Prof. H. Bleichrodt, EPS-2017-408-MKT, https://repub.eur.nl/pub/95487

Maas, A.J.J., *Organizations and their external context: Impressions across time and space*, Promotors: Prof. P.P.M.A.R Heugens & Prof. T.H. Reus, EPS-2019-478-S&E, https://repub.eur.nl/pub/116480

Maira, E., *Consumers and Producers*, Promotors: Prof. S. Puntoni & Prof. C. Fuchs, EPS-2018-439-MKT, https://repub.eur.nl/pub/104387

Nair, K.P., *Strengthening Corporate Leadership Research: The relevance of biological explanations*, Promotors: Prof. J. van Oosterhout & Prof. P.P.M.A.R Heugens, EPS-2019-480-S&E, https://repub.eur.nl/pub/120023

Nullmeier, F.M.E., *Effective contracting of uncertain performance outcomes: Allocating responsibility for performance outcomes to align goals across supply chain actors*, Promotors: Prof. J.Y.F.Wynstra & Prof. E.M. van Raaij, EPS-2019-484-LIS, https://repub.eur.nl/pub/118723

Okbay, A., *Essays on Genetics and the Social Sciences*, Promotors: Prof. A.R. Thurik, Prof. Ph.D. Koellinger & Prof. P.J.F. Groenen, EPS-2017-413-S&E, https://repub.eur.nl/pub/95489

Peng, X., *Innovation, Member Sorting, and Evaluation of Agricultural Cooperatives*, Promotor: Prof. G.W.J. Hendriks, EPS-2017-409-ORG, https://repub.eur.nl/pub/94976

Petruchenya, A., *Essays on Cooperatives: Emergence, Retained Earnings, and Market Shares*, Promotors: Prof. G.W.J. Hendriks & Dr Y. Zhang, EPS-2018-447-ORG, https://repub.eur.nl/pub/105243

Plessis, C. du, *Influencers: The Role of Social Influence in Marketing*, Promotors: Prof. S. Puntoni & Prof. S.T.L.R. Sweldens, EPS-2017-425-MKT, https://repub.eur.nl/pub/103265

Pocock, M., *Status Inequalities in Business Exchange Relations in Luxury Markets*, Promotors: Prof. C.B.M. van Riel & Dr G.A.J.M. Berens, EPS-2017-346-ORG, https://repub.eur.nl/pub/98647

Pozharliev, R., *Social Neuromarketing: The role of social context in measuring advertising effectiveness*, Promotors: Prof. W.J.M.I. Verbeke & Prof. J.W. van Strien, EPS-2017-402-MKT, https://repub.eur.nl/pub/95528

Reh, S.G., *A Temporal Perspective on Social Comparisons in Organizations,* Promotors: Prof. S.R. Giessner, Prof. N. van Quaquebeke & Dr. C. Troster, EPS-2018-471-ORG, https://repub.eur.nl/pub/114522

Riessen, B. van, *Optimal Transportation Plans and Portfolios for Synchromodal Container Networks*, Promotors: Prof. R. Dekker & Prof. R.R. Negenborn, EPS-2018-448-LIS, https://repub.eur.nl/pub/105248

Schie, R. J. G. van, *Planning for Retirement: Save More or Retire Later?* Promotors: Prof. B. G. C. Dellaert & Prof. A.C.D. Donkers, EOS-2017-415-MKT, https://repub.eur.nl/pub/100846

Schouten, K.I.M. *Semantics-driven Aspect-based Sentiment Analysis,* Promotors: Prof. F.M.G. de Jong, Prof. R. Dekker & Dr. F. Frasincar, EPS-2018-453-LIS, https://repub.eur.nl/pub/112161

Sihag, V., *The Effectiveness of Organizational Controls: A meta-analytic review and an investigation in NPD outsourcing*, Promotors: Prof. J.C.M. van den Ende & Dr S.A. Rijsdijk, EPS-2019-476-LIS, https://repub.eur.nl/pub/115931

Smolka, K.M., *Essays on Entrepreneurial Cognition, Institution Building and Industry Emergence*, Promotors: P.P.M.A.R. Heugens, & Prof. J.P. Cornelissen, Eps-2019-483-S&E, https://repub.eur.nl/pub/118760

Straeter, L.M., *Interpersonal Consumer Decision Making*, Promotors: Prof. S.M.J. van Osselaer & Dr I.E. de Hooge, EPS-2017-423-MKT, https://repub.eur.nl/pub/100819

Stuppy, A., *Essays on Product Quality,* Promotors: Prof. S.M.J. van Osselaer & Dr N.L. Mead. EPS-2018-461-MKT, https://repub.eur.nl/pub/111375

Subaşi, B., *Demographic Dissimilarity, Information Access and Individual Performance*, Promotors: Prof. D.L. van Knippenberg & Dr W.P. van Ginkel, EPS-2017-422-ORG, https://repub.eur.nl/pub/103495

Suurmond, R., *In Pursuit of Supplier Knowledge: Leveraging capabilities and dividing responsibilities in product and service contexts,* Promotors: Prof. J.Y.F Wynstra & Prof. J. Dul. EPS-2018-475-LIS, https://repub.eur.nl/pub/115138

Toxopeus, H.S. *Financing sustainable innovation: From a principal-agent to a collective action perspective,* Promotors: Prof. H.R. Commandeur & Dr. K.E.H. Maas. EPS-2019-458-S&E, https://repub.eur.nl/pub/114018

Turturea, R., *Overcoming Resource Constraints: The Role of Creative Resourcing and Equity Crowdfunding in Financing Entrepreneurial Ventures,* Promotors: Prof. P.P.M.A.R Heugens, Prof. J.J.P. Jansen & Dr. I. Verheuil, EPS-2019-472-S&E, https://repub.eur.nl/pub/112859

Vandic, D., *Intelligent Information Systems for Web Product Search*, Promotors: Prof. U. Kaymak & Dr Frasincar, EPS-2017-405-LIS, https://repub.eur.nl/pub/95490

Verbeek, R.W.M., *Essays on Empirical Asset Pricing*, Promotors: Prof. M.A. van Dijk & Dr M. Szymanowska, EPS-2017-441-F&A, https://repub.eur.nl/pub/102977

Visser, T.R. Vehicle Routing and Time Slot Management in Online Retailing, Promotors: Prof. A.P.M. Wagelmans & Dr R. Spliet, EPS-2019-482-LIS, https://repub.eur.nl/pub/120772

Vlaming, R. de.,*Linear Mixed Models in Statistical Genetics*, Prof. A.R. Thurik, Prof. P.J.F. Groenen & Prof. Ph.D. Koellinger, EPS-2017-416-S&E, https://repub.eur.nl/pub/100428

Vries, H. de, *Evidence-Based Optimization in Humanitarian Logistics*, Promotors: Prof. A.P.M. Wagelmans & Prof. J.J. van de Klundert, EPS-2017-435-LIS, https://repub.eur.nl/pub/102771

Wang, R., *Corporate Environmentalism in China*, Promotors: Prof. P.P.M.A.R Heugens & Dr F. Wijen, EPS-2017-417-S&E, https://repub.eur.nl/pub/99987

Wasesa, M., *Agent-based inter-organizational systems in advanced logistics operations*, Promotors: Prof. H.W.G.M van Heck, Prof. R.A. Zuidwijk & Dr A. W. Stam, EPS-2017-LIS-424, https://repub.eur.nl/pub/100527

Wessels, C., *Flexible Working Practices: How Employees Can Reap the Benefits for Engagement and Performance*, Promotors: Prof. H.W.G.M. van Heck, Prof. P.J. van Baalen & Prof. M.C. Schippers, EPS-2017-418-LIS, https://repub.eur.nl/

Wiegmann, P.M., *Setting the Stage for Innovation: Balancing Diverse Interests through Standardisation,* Promotors: Prof. H.J. de Vries & Prof. K. Blind, EPS-2019-473-LIS, https://repub.eur.nl/pub/114519

Wijaya, H.R., *Praise the Lord!: Infusing Values and Emotions into Neo-Institutional Theory*, Promotors: Prof. P.P.M.A.R. Heugens & Prof. J.P. Cornelissen, EPS-2019-450-S&E, https://repub.eur.nl/pub/115973

Williams, A.N., *Make Our Planet Great Again: A Systems Perspective of Corporate Sustainability,* Promotors: Prof. G.M. Whiteman & Dr. S. Kennedy, EPS-2018-456-ORG, https://repub.eur.nl/pub/111032

Witte, C.T., *Bloody Business: Multinational investment in an increasingly conflict-afflicted world*, Promotors: Prof. H.P.G. Pennings, Prof. H.R. Commandeur & Dr M.J. Burger, EPS-2018-443-S&E, https://repub.eur.nl/pub/104027

Ye, Q.C., *Multi-objective Optimization Methods for Allocation and Prediction*, Promotors: Prof. R. Dekker & Dr Y. Zhang, EPS-2019-460-LIS, https://repub.eur.nl/pub/116462

Yuan, Y., *The Emergence of Team Creativity: a social network perspective*, Promotors: Prof. D. L. van Knippenberg & Dr D. A. Stam, EPS-2017-434-ORG, https://repub.eur.nl/pub/100847

Zhang, Q., *Financing and Regulatory Frictions in Mergers and Acquisitions*, Promotors: Prof. P.G.J. Roosenboom & Prof. A. de Jong, EPS-2018-428-F&A, https://repub.eur.nl/pub/103871

Warehouses are key components in supply chain. They facilitate the product flow from production to distribution. Automation technology and robotics help warehouses to be efficient and responsive. Storage and retrieval policies determine the performance of a warehouse. Conventional storage and retrieval policies are not applicable to automated storage and retrieval system due to operational and technological disparities. This thesis studies several new storage and retrieval polices in automated warehouses. Puzzle-based storage systems are high-density storage systems that store loads on autonomous shuttles. Such systems have low throughput capacity due to lack of transport aisles. Chapter 2 studies an efficient multiple-load retrieval method that brings the loads together at an optimal joining location and then retrieves them simultaneously. This leads to shorter retrieval time compared to sequential sing-load retrievals. In another group of compact storage and retrieval systems, automated cranes transport storage bins using narrow aisles. The assignment of products to the bins and bins to the shelves are important choices that affect system's performance. Chapter 3 proposes a correlated assignment that groups products, that are frequently order together in historical customer demand, to the same product cluster. Each cluster is then assigned to a storage bin. The correlated assignment reduces the total retrieval time compared to turnover frequency-based assignment. Chapter 4 further investigates the impact of splitting the inventory of a product and dispersing it over multiple storage pod. Each pod is transported using autonomous robots and carries several dozens of correlated products.

ERIM

ERIM PhD Series
Research in Management

Erasmus University Rotterdam (EUR)
Erasmus Research Institute of Management
Mandeville (T) Building
Burgemeester Oudlaan 50
3062 PA Rotterdam, The Netherlands

P.O. Box 1738
3000 DR Rotterdam, The Netherlands
T +31 10 408 1182
E info@erim.eur.nl
W www.erim.eur.nl