IMPROVING ORDER-PICKING EFFICIENCY VIA STORAGE ASSIGNMENT STRATEGIES

MERSHA T. TSIGE

FEB. 08, 2013

UNIVERSITY OF LOUISVILLE®

LoDI LOGISTICS & DISTRIBUTION INSTITUTE

UNIVERSITEIT TWENTE.

UNIVERSITY OF LOUISVILLE
LoDI LOGISTICS & DISTRIBUTION INSTITUTE

UNIVERSITEIT TWENTE.

MASTER THESIS

INDUSTRIAL ENGINEERING & MANAGEMENT

UNIVERSITY OF TWENTE, ENSCHEDE, THE NETHERLANDS

Title:

IMPROVING ORDER-PICKING EFFICIENCY VIA STORAGE ASSIGNMENTS STRATEGIES

By:

MERSHA TETEMKE TSIGE

Student number:

1042750

E-mail:

m.t.tsige@student.utwente.nl

Supervisors:

DR. PETER C. SCHUUR, UNIVERSITY OF TWENTE

DR. SUNDERESH S. HERAGU, UNIVERSITY OF LOUISVILLE

IR. RONALD J. MANTEL, UNIVERSITY OF TWENTE

Date:

FEBRUARY 08, 2013

EXECUTIVE SUMMARY

Order picking at a fictitious warehouse is the field of the research. Which is the most labor intensive and costly activity of most warehouse, approximately 55% of the total warehouse operating expenses are related to order picking operations. Companies nowadays want to reduce supply chain costs and improve productivity within their warehouse; consequently the order picking process needs to be efficient. In order to increase the order picking efficiency different decisions at different levels can be taken. This research focuses on storage assignment strategies to increase the order picking efficiency. Storage assignment strategy is a set of rules which can be used to assign stock keeping units (SKUs) to storage locations.

The research is executed in the following way. We first perform a literature study regarding the general order picking process and storage assignment strategies which have a considerable impact on order picking efficiency. Following the literature review four storage assignment strategies are selected and implemented. These are random, cube-per-order index (COI) based on popularity (i.e., the number of orders that require a particular SKU), interaction frequency heuristic of order oriented slotting policy (IFH-OOS), and class-based (ABC-storage) policies. All but IFH-OOS are frequently used in a warehouse.

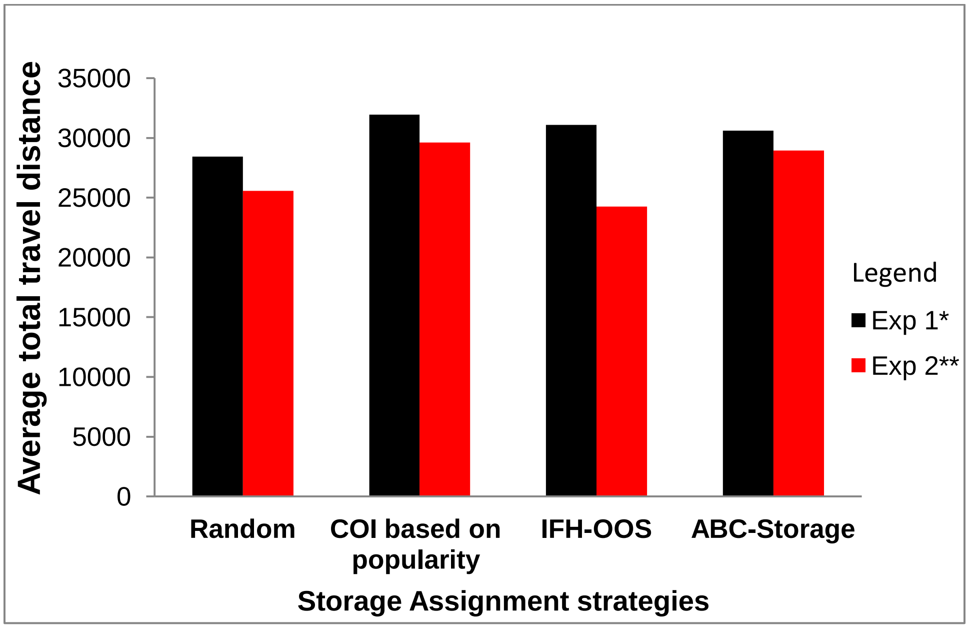
To briefly explain the basic idea behind the above mentioned storage policies:

In a *random* storage policy, each SKU is randomly assigned to an empty location in a warehouse. Where each empty location has the same probability of being selected for storage. In *class-based* storage policy, the most frequently requested item is assigned to the closest location to the Input and Output point (I/O-point). Items are first categorized into three classes – A, B, and C. Each class is then assigned to a dedicated area of the warehouse based on the number of transactions it generates. Generally, Class A items are closest to the I/O-point and Class C the farthest. Storage of items belonging to a class within the designated area is done randomly. Under *COI* based on popularity storage policy, the basic idea is to store SKUs with the highest popularity closest to the I/O-point. The *IFH-OOS* policy on the other hand allocates pairs of SKUs appearing in multiple orders in adjacent locations. In addition, frequently requested pairs of SKUs with high interaction frequency are stored close to the I/O-point.

Next, we made assumptions regarding the warehouse layout, the number of SKUs that the warehouse stocks, and the set of orders to be picked.

Finally, we develop a Monte Carlo simulation model using Visual Basic for Applications (VBA) in order to identify the most efficient storage assignment strategy. Storage assignment strategies were not the only experimental factors that we vary, but also the percentage of SKUs that appear in an order set. Our first experiment is conducted, when the order sets randomly generated from a normal distribution. In which all SKUs have equal probability of appearing in an order set. In the second experiment, we control the order generation in such a way that approximately 20% of SKUs to appear in 80% of the order set. With the intention that, the average interaction frequency between SKUs will have a relatively large value and favor the

interaction frequency based storage assignment strategies. The figure below shows the summarized results of these experiments.



Exp 1*: When the probability of appearing in an order set is equal for all items
Exp 2**: While generating order sets, we control 20% of SKUs to appear in 80% of Orders

Under the first experiment, the result shows a random storage assignment strategy has the highest reduction in total distance travelled by the order pickers. Based on our initial settings, random policy has 11% reduction in total distance travelled with respect to COI based on popularity policy. Similarly, it has 9% and 7% reduction with respect to IFH-OOS and ABC-storage respectively. On the other hand, in the second experiment IFH-OOS has the highest percentage of reduction compared to other frequently used storage assignment strategies. This is because under this second experiment the average interaction frequency between SKUs has a relatively large value. Again for our initial settings, the result shows that IFH-OOS has 18% reduction with respect to COI based on popularity storage policy and almost 5% reduction compared to random policy.

As a final point, we recommend a particular warehouse management team to adapt the suggested storage assignment strategies in different situations. For instance, we advise a warehouse manager to implement interaction frequency heuristic method of order oriented slotting policy when the average interaction frequency between SKUs in an order sets has a relatively large value and when there are not too many *singles*. Singles are SKUs that never share an order with other SKUs. Moreover, the order oriented slotting policy seems a logical approach that every single warehouse has to implement for lowering their operating expenses, except its complexity compared to other frequently used storage assignment strategies.

Table of Contents

PREFACE

This research project was executed within the context of my master thesis including an internship at the Logistics and Distribution Institute (LoDI), University of Louisville in Louisville, Kentucky. I am very grateful that I got the chance to participate in a very interesting research on the field of warehouse. It was a great and challenging time I spent at LoDI, identifying the storage assignment strategies that enhance efficiency of the order picking and improve the productivity in the warehouse.

I would like to express my sincere gratitude to my advisors: Dr. Peter C. Schuur from the University of Twente; and Dr. Sunderesh S. Heragu at the University of Louisville for their support and guidance throughout the research. Their continued support led me to the right way. I would also like to extend my appreciation to my graduation committee member: Ir. Ronald J. Mantel for his advice during my research. Special thanks to Emily J. Burks, Danielle Weires, and Randall Storey for their kind cooperation in providing all the materials I needed. In addition, I would like to extend my deepest gratitude to my beloved family for their encouragement and support. Last but not least, I would like to thank all my friends that often provided me with very useful and creative ideas for the successful completion of this project.

Thank you everyone!

Sincerely,

Mersha Tetemke Tsige

1. INTRODUCTION

In this chapter, a general introduction concerning the order picking is provided. We then present a brief literature review on order picking. At the end, the purpose of this thesis is stated.

Order picking is the process of retrieving stock keeping units (SKUs) from warehouse storage locations in response to specific customer orders. A stock keeping unit, is just the smallest physical unit of a product that is tracked by an organization. Order picking is the most labor-intensive and costly activity of most warehouse. Bartholdi and Hackman (2011) estimate the order picking operations on average accounts for 55% of the total warehouse operating expenses. Improving the order-picking efficiency plays a vital role in reducing supply chain costs and improving productivity in the warehouse. Besides, today's competitive environment has put an enormous pressure on warehouse managers to lower operating expenses by minimizing the total distance traveled by the order pickers.

Previous order picking research generally focuses on one of the following three areas: picking policies, storage policies, and routing policies. Researchers have examined different picking policies. A picking policy involves assigning SKUs or orders to pick list and subsequently retrieving from their storage locations. Common picking policies include single order picking (strict-order), batching, and zoning. Products must be put into storage locations before they can be picked to fulfill orders. This is done by using appropriate storage assignment policies. Frequently used storage assignment policies include random, dedicated, class-based, and cube-per-order index. In random storage policy, each incoming pallet is randomly assigned a location in the warehouse that is arbitrarily selected from all eligible empty locations with equal probability (Petersen, 1997). In *dedicated storage*, each SKU is assigned a specific storage location(s). Class-based (ABC) storage policy assigns the most frequently requested SKUs closest to the Input and Output point (I/O-point). The well-known Cube-per-order index (COI) policy stores frequently requested items requiring smaller storage space closest to the I/O-point (Heskett, 1963). The approach we used for COI policy is that we rank the SKUs from high to low according to their popularity. Thus, SKU's with the highest popularity will be stored closest to the I/O-point. Order oriented slotting (OOS) is an alternate storage assignment policy that is developed recently. Several methods are available to implement OOS. The interaction frequency heuristic (IFH) is among these, in which items with high interaction frequency are stored in adjacent locations and frequently requested pairs of items with high interaction frequency are stored close to the I/O-point (Mantel, Schuur, & Heragu, 2007). The interaction frequency heuristic introduced by Mantel et al. (2007) assumes each SKU has only one storage location. But we extend this to encompass a SKU which has more than one storage locations.

Routing policies determine the picking sequence of SKU on the pick list. These policies range from simple heuristics (De Koster, Le-Duc, & Roodbergen, 2007) to optimal procedures (Ratliff & Rosenthal, 1983). In practice, mostly heuristics are used and yield nearly optimal solutions. This is because warehouses have different layout and an optimal algorithm is not available for every layout. In addition, most heuristics are easy to use. *S-shaped* (*traversal*) *heuristic* is the most common and simplest heuristic used in practice. Under this policy, the order picker begins by entering the pick aisle closest to the I/O-point. Any aisle containing at least one item is traversed over the entire length. Aisles with no picks are skipped. After picking the last SKU, the order picker takes the shortest route back to the I/O-point.

Against this background, the purpose of this thesis is to determine the conditions under which alternate storage assignment strategies help minimize the total distance traveled by the order pickers. In addition, we use the results of a Monte Carlo simulation study to show the amount of reduction of total travel distance and ease of implementation. A Monte Carlo simulation is a problem solving technique used to approximate the probability of certain outcomes by running multiple trial runs using random variables. Furthermore, we suggest how a warehouse manager can use this as a decision support tool for lowering their operating expenses through implementing the right storage assignment strategy in different situations.

2. RESEARCH DESIGN

Order picking at a fictitious warehouse is the field of the research. In section 2.1, a short description about warehouse functions and flows are presented. An overview of order picking system is provided in section 2.2. The objective of the research as well as the research methodology is provided in sections 2.3 and 2.4 respectively.

2.1. WAREHOUSE FUNCTIONS AND FLOWS

Figure 1 shows the typical functional areas and flows within warehouses. The five functional areas include: receiving, staging for cross-docking, reserve, forward and shipping. Receiving, transfer and put away, order picking, cross-docking, and shipping are the main respective warehouse activities.

The receiving activity includes unloading of individual products from the suppliers or customers, updating inventory record and inspection to check whether there is any quantity and quality variation. Transfer and put away includes the transfer of incoming products to storage areas. The major activity of most warehouses is order picking. This is the process of obtaining the right products in the right amount from the right storage locations in response to specific customer requests. Cross-docking activity is when the received products are directly transferred to shipping area. In the cross-docking operation, products might also have a short stay on the staging area with or without order picking activity. Lastly, shipping is carried out to transfer the picked order items to the next destination. The reserve area (also known as bulk or overstock area) is for bulky products that will stay in the warehouse for longer period of time. In most cases, a pallet load is broken to cases and further into pieces in the reserve area. Fast order picking and value added activities can be performed in the forward (fast-pick) area. Forward area is typically smaller in size compared to the other storage areas.

Against this background, four material flows are possible in a warehouse. The first flow is the cross-docking activity, in which products are either stored in a staging area for a while or directly moved to shipping area. The second flow is when products stored in the reserve area relatively for longer period and order picking activities performed. The third flow is when products are first stored in reserve area and then moved to the forward area. In the fourth type of flow received products are directly moved into forward area so that the respective order consolidation can be carried out.

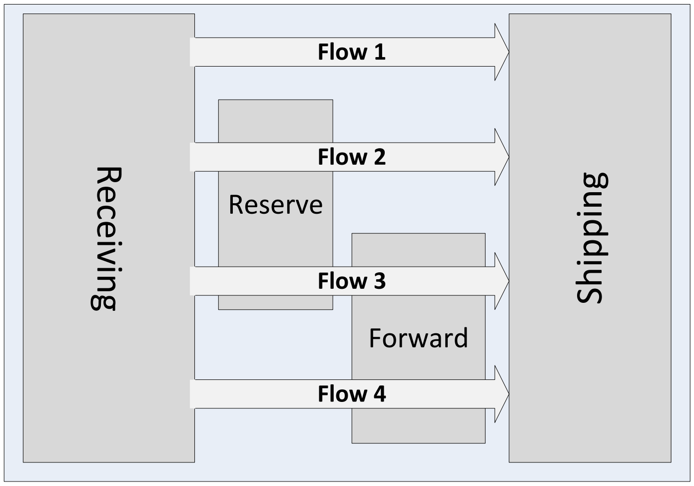


Figure 1 Typical functional areas and flows in a warehouse (Heragu, Du, Mantel, & Schuur, 2005)

In the next section, the order picking process is presented which is the focus of the research.

2.2. ORDER PICKING

Order picking involves the process of retrieving products from storage (or buffer areas) in response to a specific customer request (De Koster et al., 2007). It is the most labor-intensive and costly activity of most warehouse, approximately 55% of the total warehouse operating expenses are related to order-picking operations (Bartholdi & Hackman, 2011). That is why warehousing scholars consider order picking as the most promising area for productivity improvements. Either humans or machines can be used for order picking. The majority of warehouses prefer to use humans for order picking. According to De Koster (2004), the most common order picking system is *picker-to-parts* systems, in which the order pickers walks or drives along the aisle to pick items.

Figure 2 shows the typical manual (picker-to-parts) activities and their percentage distribution of order-picker's time. Notice that travel is the dominant component and has a greater contribution to order-picking time. According to Bartholdi and Hackman (2011) travelling is a waste, because it costs labor hours without adding value to the product. Accordingly, it is the first candidate for improvement. Petersen and Aase (2004) consider travel time as an increasing function of travel distance for manual order picking process. Thus, minimizing total travel distance is often considered as the primary objective in warehouse design and optimization. Other important objectives may include minimizing total cost (investment and operational), minimizing overall throughput time of an order, and maximizing the use of labor.

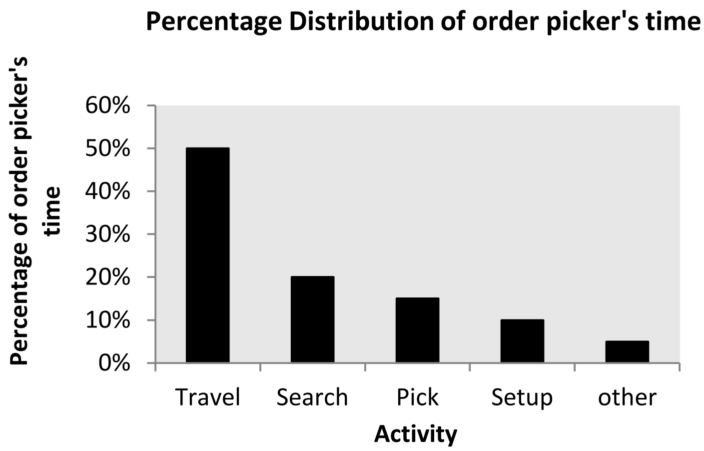


Figure 2 Distribution of order picker's time (Tompkins, White, Bozer, & Tanchoco, 2003)

Figure 13 in Appendix 1 shows the general order picking process flowchart. The process starts with receipt of customer orders known as order-line. Each order line consists of the item and quantity of the requested orders. A warehouse management system (WMS) software assists in checking the availability of inventory for orders. If there is sufficient inventory the WMS convert order-lines to pick-lines with instructions for order pickers. The pick-lines can be single or multiple lines depending on the number of location to be visited. Once the pick-lines are created, the WMS may sequence them to reduce the travel time. Next, pick lists will be created which consists of order-lines. Deferent media might be used for organizing the pick list. For example, printed sheet of papers, radio frequency or voice transmission. The next step is either strict-order or batch or split of orders. They depend on the capacity of pick carts and number of order-lines. Once batching and splitting of orders completed, humans or automated equipment may be used to pick the requested orders. In order to increase the service level checking of customer orders is crucial. Finally, consolidation of orders will be carried out and then products will be shipped to the next destination.

The design of order picking systems is often complicated, because it encompasses a wide range of internal and external factors. Figure 3 shows the complexity of order-picking systems design.

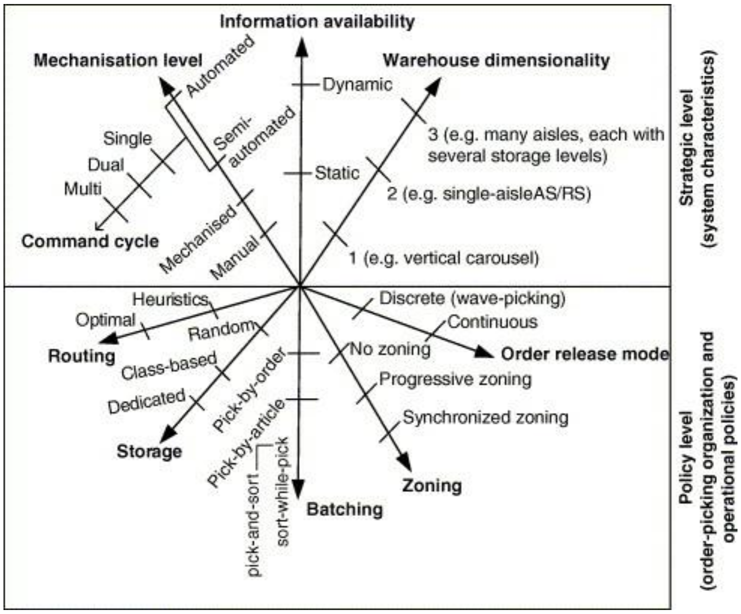


Figure 3 Complexity of order-picking systems (De Koster et al., 2007)

Rouwenhorst et al. (2000) classify the decisions to design and control of order picking systems as tactical or operational level. The common decisions at these levels include:

- Layout design and dimensioning of storage system
- Assigning products to storage locations
- Assigning orders to pick batches and grouping into work zones (batching and zoning)
- Order picker routing (routing)
- Sorting picked units per order and grouping all picks of the orders (order accumulation/sorting)

In order to meet the above mentioned objectives, decisions made at these levels are strongly interrelated. For example, De Ruijter, Mantel, Schuur, and Heragu (2009) show that certain storage assignment strategies are strongly interrelated with batching and slightly with zoning. However, considering these decisions simultaneously is not only difficult but also not realistic. Nowadays, companies are looking to cut costs and improve productivity within their warehouses by applying the above mentioned decisions. Even though, this research focuses on storage assignment strategies we will also try to include simple routing policies and picking policy.

The purpose of this research is also to help the warehouse manager to make a well-informed decision concerning the order-picking process. The research will investigate empirically the significant association between storage assignment strategies and order-picking efficiency. To do so, a Monte Carlo Simulation model will be developed. It will give a clear numerical and visual insight in to the order picking process.

The next section clearly states the objective of the research and related research questions.

2.3. RESEARCH OBJECTIVES

As we have already stated, companies nowadays want to reduce supply chain costs and improve productivity within their warehouse; consequently the order picking process needs to be efficient. Hence, the main objective of the research is to do a Monte Carlo simulation study which is used to determine the conditions under which alternate storage assignment strategies help minimize the total distance traveled by the order pickers. Also, use the results of the Monte Carlo simulation study to show the amount of reduction of total travel distance and ease of implementation.

In order to accomplish these objectives, there is a need to answer the following research questions:

RQ1. What are the storage assignment strategies which have an impact on order picking efficiency?

This research question will be answered by reviewing the relevant literature. Available storage assignment strategies are the independent variables. The most relevant independent variables that influence the manual order-picking process summarized from literature include: random storage, cube-per-order index (Heskett, 1963), class-based storage, and order oriented slotting methods (De Ruijter et al., 2009; Mantel et al., 2007). The dependent variables are the order picking efficiency measures. Order picking efficiency can be measured in different ways, but here we assumed they are measured in total distance traveled by the order pickers.

RQ2. Which storage assignment strategy is the most efficient?

The second research question will be answered by developing a Monte Carlo simulation model in Visual Basic for Applications, which will give a clear numerical and visual insight of the process. The hypothesis is that the storage assignment strategies positively influence the order picking efficiency. In order to compare the results between storage assignment strategies, two experimental designs are developed and they are described in chapter 4.

The figure below shows the conceptual model consisting of a set of related variables and previously stated hypothesis.

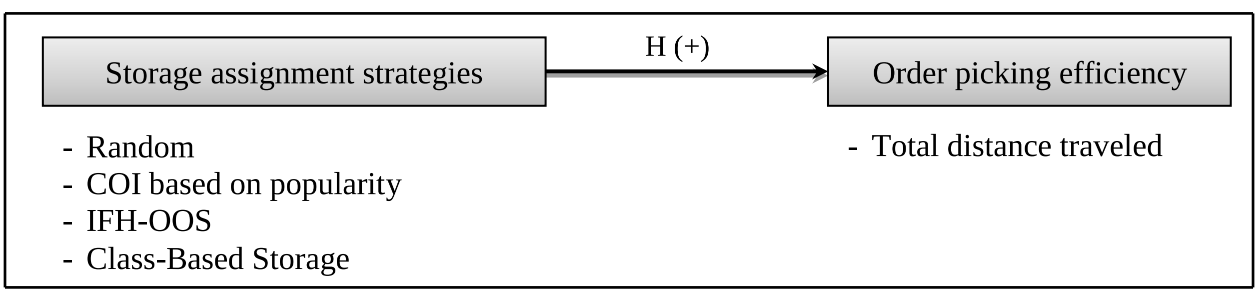


Figure 4 Conceptual framework

2.4. RESEARCH METHODOLOGY

We generate two different order sets using Visual Basic for Applications (VBA). It is assumed that the warehouse stocks a total of 100 SKU. Each SKU has varying sizes of items. Using these 100 SKUs, 160 orders of varying sizes are generated. Order size varies between 3 and 7 units and generated from a Normal Distribution, in which the probability of appearing in an order is equal for all SKUs. While generating the second order set, we control 20% of SKUs to appear in 80% of the orders.

Prior to generating the order sets, we perform a literature study regarding the general order picking process and storage assignment strategies which have a considerable impact on order picking efficiency. We select four storage assignment strategies and develop a Monte Carlo simulation in VBA. The assumptions made while conducting the experiments along with the essential experimental factors and settings are provided in chapter 4. We then determine the total distance traveled by the order pickers given previously generated order sets and analyze the results. We finally select the most efficient storage assignment strategy under each experiment. A warehouse manager could use these as a decision support tool for lowering their operating expenses through implementing the right storage assignment strategy in different situations.

2.5. OUTLINE OF THE THESIS

The aim of this thesis is to provide the warehouse manager with a short summary of the results, conclusions and recommendations. Thereafter, a table of contents is given to simplify the orientation within the thesis framework. A preface follows that section.

Chapter 1 provides a general introduction about the order picking process. It also briefly states the focus of prior order picking research and states the purpose of this thesis. In chapter 2, a brief highlight of warehouse functions and an overview of order-picking system followed by research objectives and research methodology are provided. In chapter 3, many relevant topics covered in the literature including: warehouse layout design, picking policies, storage assignment strategies, and routing policies are provided. Next, in chapter 4 we present the assumptions made while conducting the experiment and the experimental factors along with their ranges. Results of each experiment along with comparison of results are provided in chapter 5. In this chapter, we also provide a small example which will give the reader an insight about the general ideas behind this thesis. We provide the conclusions and discuss the potential research directions in chapter 6 and 7 respectively. The thesis report finishes with the reflection of the internship and master thesis at University of Louisville, Logistics and Distribution Institute in chapter 8.

3. Literature Review

There are many relevant topics that will be covered in the review of literature, taking into account factors that influence the order picking efficency. The primary focus of this research is on identifying the conditions under which alternate storage assignment policies help minimize the total distance traveled by the order pickers. However, there are other issues that are interconnected with storage assignment strategies, all of which make the order-picking more efficient. Examples include the warehouse layout, routing policies, batching, and zoning methods. Therefore, issues related to warehouse layout design are discussed in section 3.1. In section 3.2, a short introduction on storage assignment policies followed by exsiting storage assignment policies from literature are provided. Routing methods and picking policies are also discussed in this chapter.

3.1. Warehouse Layout Design

Warehouse layout is one important factor affecting the order picking process. Caron, Marchet, and Perego (2000) find that the warehouse layout has a considerable effect on order picking travel distance. They point out the layout design has an effect of more than 60% on the total travel distance, and also find the relationship between warehouse layout and order picking travel distance. Therefore, warehouse layout has to be taken into account while designing the order picking system. Usually, the unit-load warehouse layout is based on a rectangular shape, in which SKUs arrive on pallets and leave on pallets (Bartholdi & Hackman, 2011). Because pallets are mostly standardized and handled one-at- time, models of space and labor are simple linear models. It takes about *n* times the space to store *n* pallets; and it takes about n times the labor to pick n pallets. Furthermore, in such warehouses, the number of trips to pick full pallets is equal the number of pallets requested. And as long as replenishing is a unit-load process, the number of restocks is always equal to the number of pallets sold. When picking in volume, cartons are stored on pallets and so replenishing is a unit-load process, but picking is not, it creates extra complexities in models of space and labor. Once effect is that it becomes much more difficult to measure the convenience of an individual location. When we move to other complicated warehouse types, it could be in a circular fashion or any other irregular shapes. According to De Koster et al. (2007), and Heragu (2008) factors to be considered in internal warehouse layout design include: number of blocks, number, length and width of picking aisles, number and shape of cross aisles, level of racks, and position of Input and Output point (I/O-point). Figure 5 shows the typical layout decisions in order picking system design.

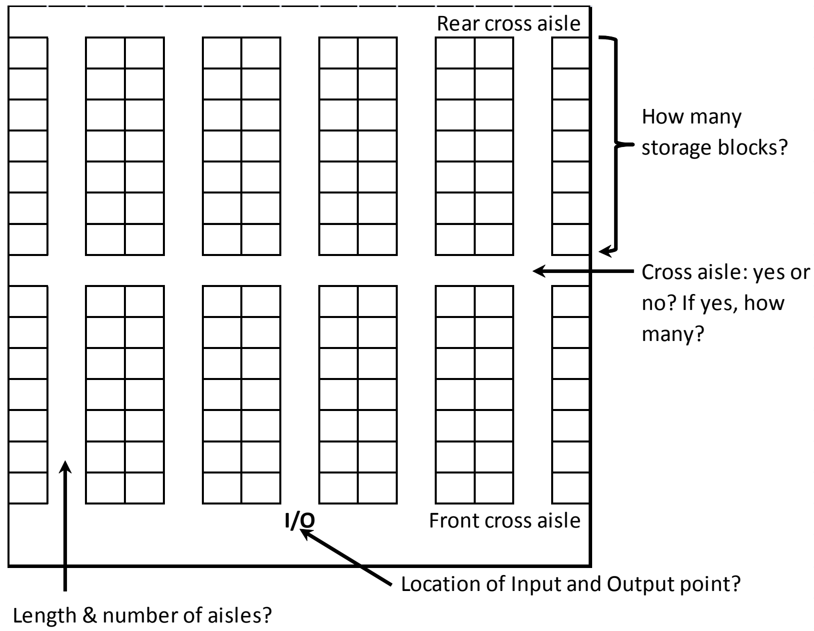


Figure 5 Typical layout decisions in order picking system design (De Koster et al., 2007)

Caron et al. (2000) propose three types of warehouse layouts. The first is a parallel storage aisle with I/O-point located in the middle; the second and third are vertical aisle, but I/O- point is located in the middle and lower left corner, respectively. Roodbergen and De Koster (2001b) consider to put a cross aisle in between the originally parallel aisles, and found a considerable improvement in average picking distance with cross aisle. Figure 6 shows the typical warehouse layout in the industry.

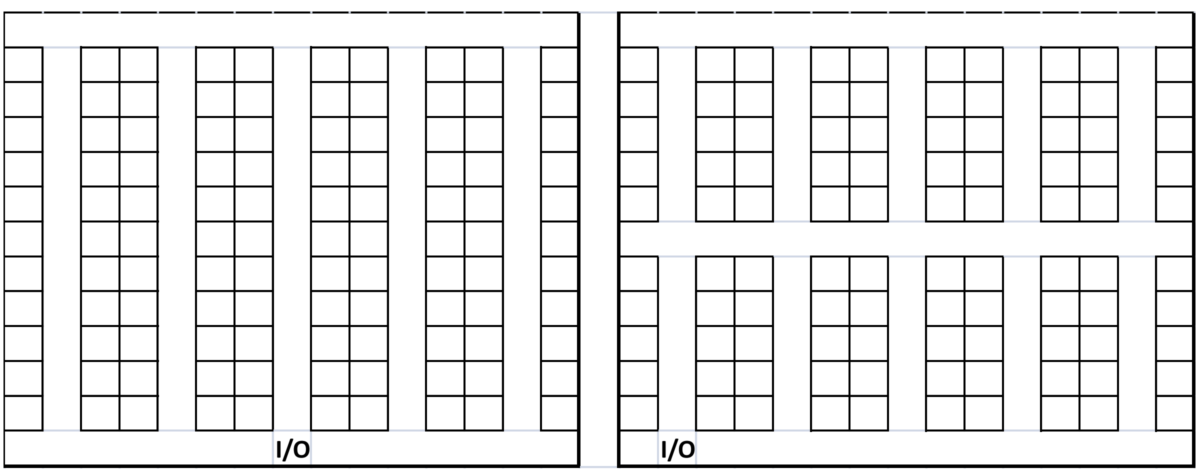


Figure 6 Typical warehouse layouts in industry

Recently, Gue and Meller (2009) described the non-traditional aisle designs in which the rules requiring parallel picking aisles and orthogonal cross aisles were relaxed. They developed two designs Flying-V and Fishbone (see Figure 7). The Flying-V design, contains a cross aisle that project diagonally in a piecewise linear fashion from the I/O-point, and offers approximately 10% reduction in the expected picking travel distance. The Fishbone design offers a reduction about 20%. However, the designs assume that travel begins and ends at a single I/O-point. In practice, this is not the case. For example, the warehouse may have two, three, or more material handling devices, so that travel begins and ends from two or more points. Further, Gue, Ivanović, and Meller (2012) propose two new aisle designs, the modified Flying-V cross aisle and the inverted-V cross aisle, to facilitate flow between multiple I/O-points and locations in a unit-load warehouse. The new designs offer less than 3% reduction in the expected travel distance compared with a traditional warehouse with single I/O-point. Multiple I/O-points have also been examined in the context of an order picking system by Eisenstein (2008). Because the fishbone design has a clear orientation toward a single I/O-point, we do not consider it further here.

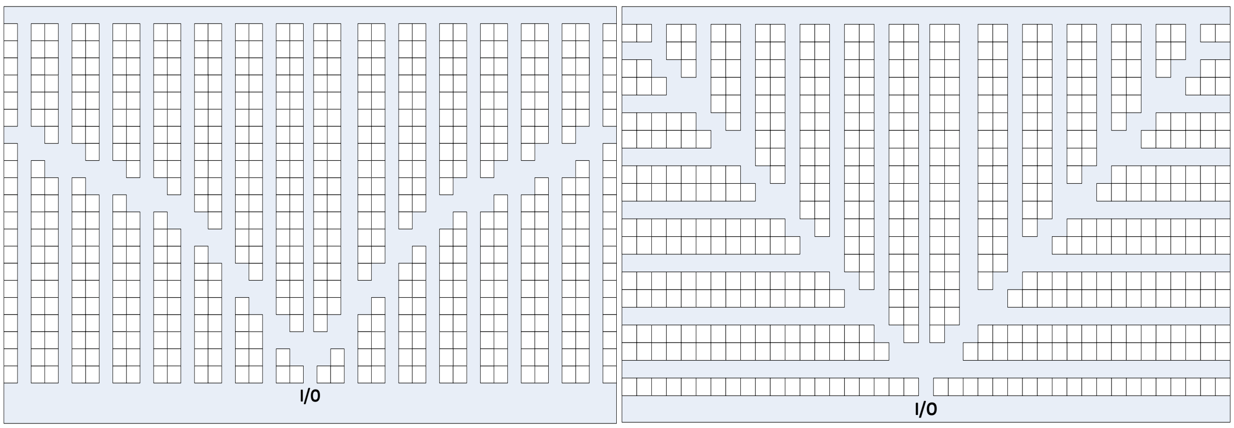


Figure 7 The Flying-V and Fishbone designs introduced by Gue and Meller (2009)

Compared to manual (picker-to-parts) order picking systems, the warehouse layout design for unit-load (mainly AS/RS) systems has received considerable attention, for example Johnson and Brandeau (1996), Lerher, Potrč, Šraml, and Tollazzi (2010), and Sarker and Babu (1995). In this thesis, we will consider the layout design for manual-pick order picking systems. Details on the warehouse layout can be found in section 4.1.

3.2. STORAGE ASSIGNMENT POLICIES

Stock keeping units (SKUs) need to be put into storage locations before they can be picked to fulfill customer requests. A storage assignment method (slotting strategy) is a set of rules which can be used to assign SKUs to storage locations. Frequently used types of storage assignments strategies are either *random* or *dedicated*. In random storage policy, each incoming pallet is randomly assigned a location in the warehouse that is arbitrarily selected from all eligible empty locations with equal probability (Petersen, 1997). If the order pickers can choose the location for storage themselves, they are likely to choose a location close to the I/O-point. Consequently, we can have more empty space on aisles those far away from the I/O-point. This system is also known as closest open location storage (De Koster et al., 2007). Therefore, the random storage policy will only work with a computer system.

In *dedicated storage*, each SKU is assigned a specific slot location(s). However, dedicated storage often requires more space than random storage policy. This is because no other item can be stored in a location assigned to another item, even if that location is empty. One advantage of this storage policy is that order pickers become familiar with product locations. It can also be used to obtain a good stacking sequence by storing products in order of weight and routing the order pickers accordingly.

In between random and dedicated, a *class-based (ABC) storage* policy assigns the most frequently requested SKUs to the best locations on the rack face. Class-based (ABC) storage policy is easy to use and simple to understand. Items are categorized into classes based on the number of times they appear in an order set. Class A items are relatively few in numbers but account for a large amount of the activity, while class C items are relatively large in numbers but account for a relatively small amount of the activity. Items between the above two classes constitute class B. The classification is based on the popular Pareto principle. Each class is then assigned to a dedicated area of the warehouse. Storage of items within the area is random. Fast-moving products can be stored close to the I/O-point and at the same time low storage space requirements of random storage can be applied.

Family grouping is an alternative to the previously mentioned storage assignment policies. Family grouping involves the possible relations between products. The statistical correlation between items is required to apply family grouping. Thus, products can be stored close to each other if they often appear together in an order (Brynzér & Johansson, 1996). Other more advanced methods found from literature are discussed in the following sections.

3.2.1. CUBE-PER-ORDER INDEX (COI)

Heskett (1963) is the first scholar who deals with the storage assignment policy. He introduced the well-known cube-per-order index (COI) storage assignment policy, which is aimed to store items that are frequently requested and require smaller storage space closest to the I/O-point. Below, an integer linear programming (ILP) model is formulated for this storage policy (Heskett, 1963). Let us first introduce the following relevant variables:

$q = number\ of\ storage\ locations$
$n =\ number\ of\ SKUs$
$m = number\ of input/output\ points\ (I/O - points)$
$S_j = number\ of storage\ locations\ required\ for\ SKU_j$
$T_j = number\ of\ trips\ in/out\ of\ storage\ for\ SKU_j, i.e.,\ \ throughput\ of\ SKU_j$
$p_i = percentage\ of\ travel\ in/out\ of\ storage\ to/from\ I/O - point\ i$
$d_{ik} = distance\ required\ to\ travel\ from\ I/O - point\ i\ to\ storage\ location\ k$

$$x_{jk} = \begin{cases} 1, & if\ SKU_j\ is\ assigned\ to\ storage\ location\ k \\ 0, & Otherwise \end{cases}$$

$f(x) = average\ distance\ traveled$

$COI_j = Cube - per - order\ index\ of\ item\ j$

The expected travel distance between location k and the I/O-points is defined as follows:

$$f_k = \sum_{i=1}^{m} p_i d_{ik}$$

Hence, the ILP model is given below:

$$Minimize\ z = \sum_{j=1}^{n} \sum_{k=1}^{q} \frac{T_j}{S_j} f_k x_{jk}$$

s.t. $\sum_{j=1}^{n} x_{jk} = 1 \qquad k = 1, 2, \dots, q$

$\sum_{k=1}^{q} x_{jk} = S_j \qquad j = 1, 2, \dots, n$

$x_{jk} = (0, 1) \qquad \forall j, k$

The above problem representation corresponds to a balanced transportation problem. Implicitly it has been assumed that $q = \sum_j S_j$. For the problem to be feasible, it must hold that $q \geq \sum_j S_j$. If $q - \sum_j S_j > 0$, the previous balanced formation is obtained by introducing a fictitious SKU 0, with $S_0 = q - \sum_j S_j\ and\ T_0 = 0$. The following procedures can be used to solve the above ILP model to optimality:

1. Compute $f_k = \sum_{i=1}^{m} p_i d_{ik}$
2. Renumber locations by $f_1 \leq f_2 \leq \cdots \leq f_q$
3. Renumber SKUs by $\frac{T_1}{S_1} \geq \frac{T_2}{S_2} \geq \cdots \geq \frac{T_n}{S_n}$
4. Assign locations 1, 2, …, S_1 to SKU_1, locations S_1+1, S_1+2, …, S_1+S_2 to SKU_2, etc.

The COI storage policy applies similar procedure as above by introducing the cube-per-order-index of an item. The cube-per-order index of an item is the ratio of the number of storage locations (storage space) it requires to the number of times it is requested. With the above notation: $COI_j = \frac{S_j}{T_j}$

The Cube-per-order index is first computed and recorded separately for each SKU. All SKUs on the list are then ranked based on their COI, SKU with the lowest index being ranked first. Next, the minimal distances of the locations to the I/O point have to be determined. The distances between the I/O-point and pick locations are measured taking into account the aisle structure of the picking area and these values are stored in a distance matrix D. Hence, SKUs with the lowest COI are given prime locations closest to the I/O point. Instead of using the cube-per-order index of a SKU, we use the popularity. Where popularity (f_{i0}) is the number of orders that require a SKU *i*. Therefore, the procedure that we applied here is: we rank the SKUs from high to low according to their popularity. And so, the SKUs with the highest popularity are given the closest storage locations to the I/O-point. From now on, to clear confusion we name this storage assignment policy as COI based on popularity.

3.2.2. ORDER ORIENTED SLOTTING METHODS

Mantel et al. (2007) introduce the concept of order oriented slotting. OOS policy stores the SKUs in the warehouse in such a way that the total distance traveled by the order pickers is minimized. Several methods are available to implement OOS. Let us first introduce the following parameters:

SKUs : $i = 1, 2, \dots, I$

$f_{ij} = interaction\ frequency,$ the number of orders that require both SKU i and SKU j

$f_{i0} = popularity,$ the number of orders that require SKU i

$d_{ij}^{r} = routing - specific\ distance\ between$ SKU i and SKU j

$d_{i0}^{r} = routing - specific\ distance\ between$ SKU i and the $\mathrm{I/O} -$ point

Mantel et al. (2007) argue that, SKUs with a high interaction frequency must be close to each other in order to minimize order picking effort. In addition, popular SKUs must be close to the I/O-point. They present several methods used to tackle the general OOS problem. We present the following three: interaction frequency heuristic (IFH), interaction frequency based quadratic assignment heuristic (IFH QAP), and order oriented product swapping (OPS) method.

3.2.2.1. INTERACTION FREQUENCY HEURISTIC

In this heuristic, first the locations of all SKU can be determined by applying COI storage assignment. Then *singles* (i.e., the SKUs that never share an order with other SKUs) are allocated in accordance with their COI-locations. Next, it ranks the interaction frequencies in non-increasing order. The basic idea is that SKU *i* and SKU *j* with a high interaction frequency f_{ij} should be placed close to each other and in accordance with their popularity. The heuristic has the following detail steps:

1. Determine the *popularities* f_{i0} and *interaction frequencies* f_{ij} of all SKUs. Store the computed values in a matrix.
2. Determine the *routing-specific* distances between any two pick locations d_{ij}^{r} (including the I/O-point,d_{i0}^{r}). Store these values in a distance matrix **D**.
3. Determine the locations of all SKUs in case a COI storage assignment is applied. Call these tentative locations as *COI-locations*.
4. Identify *singles* (SKUs that never share an order with other SKUs). Assign these SKUs to their COI-location.
5. Sort the positive interaction frequencies in decreasing order.
6. Consider a certain interaction frequencies(f_{ij}). If SKU *i* and SKU *j* have already been allocated, then process the next interaction frequency; otherwise proceed with the following two cases:

Case1- Neither SKU *i* nor SKU *j* has been allocated so far: Create for SKU *i* a set A_i and for SKU *j* a set A_j in which allowed locations can be stored. Add the COI-location of SKU *i* to set A_i and the COI-location of SKU *j* to set A_j. Check the distances from the COI-location to the I/O-point for both SKUs (d_{i0}^r and d_{j0}^r). Consider set A_i. Given a certain factor β (>0 and <<1), a free location *x* for which holds that $(1-\beta)\, d_{i0}^r \leq d_{x0}^r \leq (1+\beta)\, d_{i0}^r$ is added to set A_i. Certainly, only locations with suitable compartment sizes are considered. After set A_i has been filled with allowed locations, the same procedure is executed for set A_j. Then, we choose the locations of SKU *i* and SKU *j* from the sets A_i and A_j such that d_{ij}^r is minimal.

Case2 – Either SKU *i* or SKU *j* has been allocated so far: Apply the same procedure as in case1, but now the location of one SKU is already fixed. So, a set of allowed locations for only one SKU needs to be created. If no suitable location exists, then process the next interaction frequency.

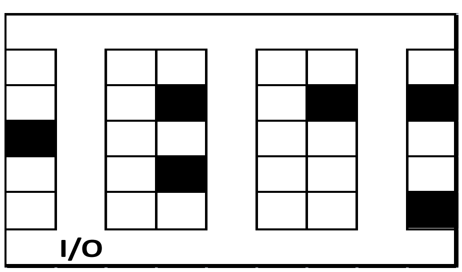
7. Finally, after processing all $f_{ij}'s$, some SKUs remain unassigned, because their allowed locations are already occupied by other SKUs. The popularities of unassigned SKUs are sorted in decreasing order and free locations are determined. The remaining unassigned SKUs are allocated based on the COI based on popularity storage assignment policy.

The interaction frequency heuristic presented above assumes each SKU has only one storage location. But we consider multiple items per SKU to fully stock the warehouse. And that means one SKU has more than one storage locations. So as to fit to this situation, we extend the above model and change the following parameters:

$d_{ij}^r = routing - specific\ distance\ between$ center of gravity of the locations SKU i and SKU j

$d_{i0}^r = routing - specific\ distance\ between$ center of gravity of locations of SKU i $and\ the\ I/O - point$

Where center of gravity (CoG) is the unique point at the center of a distribution of mass of SKUs in space that has the property of the weighted position vectors relative to this point sum to zero. The following diagram illustrate how to calculate the CoG of a particular SKU which has one item in first aisle, two items in the second aisle, and three items in the third aisle of the warehouse.



CoG of a particular SKU $= \frac{1*1+2*2+3*3}{1+2+3} = 7/3$

We can also formulate the following simple equation to calculate the CoG of a particular SKU as:

$$CoG = \frac{1}{M}\sum_{i=1}^{n} a_i I_i$$

Where: M is the total number of items that a particular SKU consists of;

a_i is the aisle index number; and

I_i is the total number of items of a SKU in a specific aisle

And take the absolute difference between values in order to find the routing-specific distance between the CoG of locations of two SKUs.

3.2.2.2. INTERACTION FREQUENCY BASED QUADRATIC ASSIGNMENT HEURISTIC

Mantel et al. (2007) formulate the OOS problem as a form of the quadratic assignment problem. Quadratic assignment optimization problem is an NP-hard. So, only small problem size can be solved to optimality. The problem formulation of this method is as follows:

$$\min_{a \in S} z(a) = \sum_{i=1}^{I-1}\sum_{j=i+1}^{I} f_{ij}\, d_{ij}^{r}(a) + \alpha \sum_{i=1}^{I} f_{i0}\, d_{i0}^{r}(a) \qquad (1)$$

In the above expression, *S* is the set of all storage assignments, $d_{i0}^{r}(a)$ denotes the *routing-specific* distance between SKU *i* and the I/O-point and $d_{ij}^{r}(a)$ equals the *routing-specific* distance between SKU *i* and SKU *j* (for storage assignment *a*). Finally, the constant α provides the relative weight of the term$\sum_{i=1}^{I} f_{i0}\, d_{i0}^{r}(a)$. In the above IFH QAP heuristics the first term in (1) assures frequently ordered SKUs placed close together, while the second term forces fast movers to be allocated close to the I/O-point. The objective of the heuristic is to minimize z over S. De Ruijter et al. (2009) argue that the value of α should be determined empirically. They propose the following rule of thumb for determining α (for storage assignment *a*):

$$\alpha = \frac{\sum_{i=1}^{I-1}\sum_{j=i+1}^{I} f_{ij}\, d_{ij}^{r}(a)}{\sum_{i=1}^{I} f_{i0}\, d_{i0}^{r}(a)} \qquad (2)$$

This rule of thumb sets the weight of the two terms in (1) about the same.

3.2.2.3. ORDER ORIENTED PRODUCT SWAPPING

De Ruijter et al. (2009) also developed the order oriented product swapping (OPS) method. They argue this method tackle the quadratic assignment problem directly by using simulated annealing together with a simple 2-exchange procedure to swap SKUs from their locations. The basic idea of this method is then to swap two SKUs from their storage locations and calculate the associated difference in objective value. They showed us this heuristic yield a good results for large problem instances with the cost of computation time.

In the next section, available routing policies from the literature are briefly presented. Routing policies in general have a great influence on the order picking efficiency. Some of the above mentioned storage assignment strategies depend on the routing specific distances.

3.3. ROUTING POLICIES

The objective of routing policy is to sequence the items on the pick list to ensure a route with efficient travel through the warehouse. Ratliff and Rosenthal (1983) provide an algorithm that is used to obtain the optimal order picking routes in a conventional warehouse which consists of parallel aisles, and two cross aisles, one in the front, and the other in the back. Their algorithm can solve the problem with run times that are linear in the number of aisles and the number of pick locations. In practice, mostly heuristics are used for solving the problem of routing order pickers in a warehouse. This is because many warehouses have different layout and an optimal algorithm is not available for every layout. Hence, the algorithm of Ratliff and Rosenthal (1983) cannot be applied for these cases. In addition, optimal routes may seem illogical to order pickers. Hence, the order picker deviates from the optimal route and creates his or her own route that is easier to follow. De Koster et al. (2007) summarize the commonly used heuristic methods for routing order pickers in single-block warehouses. Figure 8 shows example routing strategies.

One of the most common heuristic and often applied in practice for routing order pickers is the *S-shape* (*traversal*) *heuristic*. Under this policy, the order picker begins by entering the pick aisle closest to the I/O-point. Any aisle containing at least one item is traversed over the entire length. No backtracking is permitted in any aisle. Aisles with no picks are skipped. After picking the last item, the order picker takes the shortest route back to the I/O-point. Another common heuristic for order pickers is the *return policy*. Using the return method, an order picker enters and leaves each aisle from the same end. The *Mid-point strategy* splits the warehouse into two areas (See Figure 8). Picks in the front half are accessed from the front cross-aisle. Similarly, picks in the back half are accessed from the back cross-aisle. Either the first or the last aisles are traversed over the entire length if there are items in the half opposite to the I/O-point. The *largest gap strategy* looks similar to the mid-point strategy except the order picker enters an aisle as far as the largest gap within an aisle. The gap represents the separation between two adjacent picks, between the first pick and the front cross-aisle, or between the last pick and the rare cross-aisle. If the largest gap is between two adjacent picks, the order picker performs a return route from both ends of the cross-aisle. Otherwise, a return route from either the front or rear cross-aisle is used. So, the largest gap within an aisle is the portion of the aisle that the order picker does not traversed. Using the *composite (combined) heuristic*, aisles with picks are either entirely traversed or entered and left at the same end. But, for each visited aisle, the choice is made by using dynamic programming (Roodbergen & De Koster, 2001b). All these methods were originally developed for single-block warehouses. Roodbergen and De Koster (2001a) have modified these methods in such a way that they can be used for multiple block warehouses.

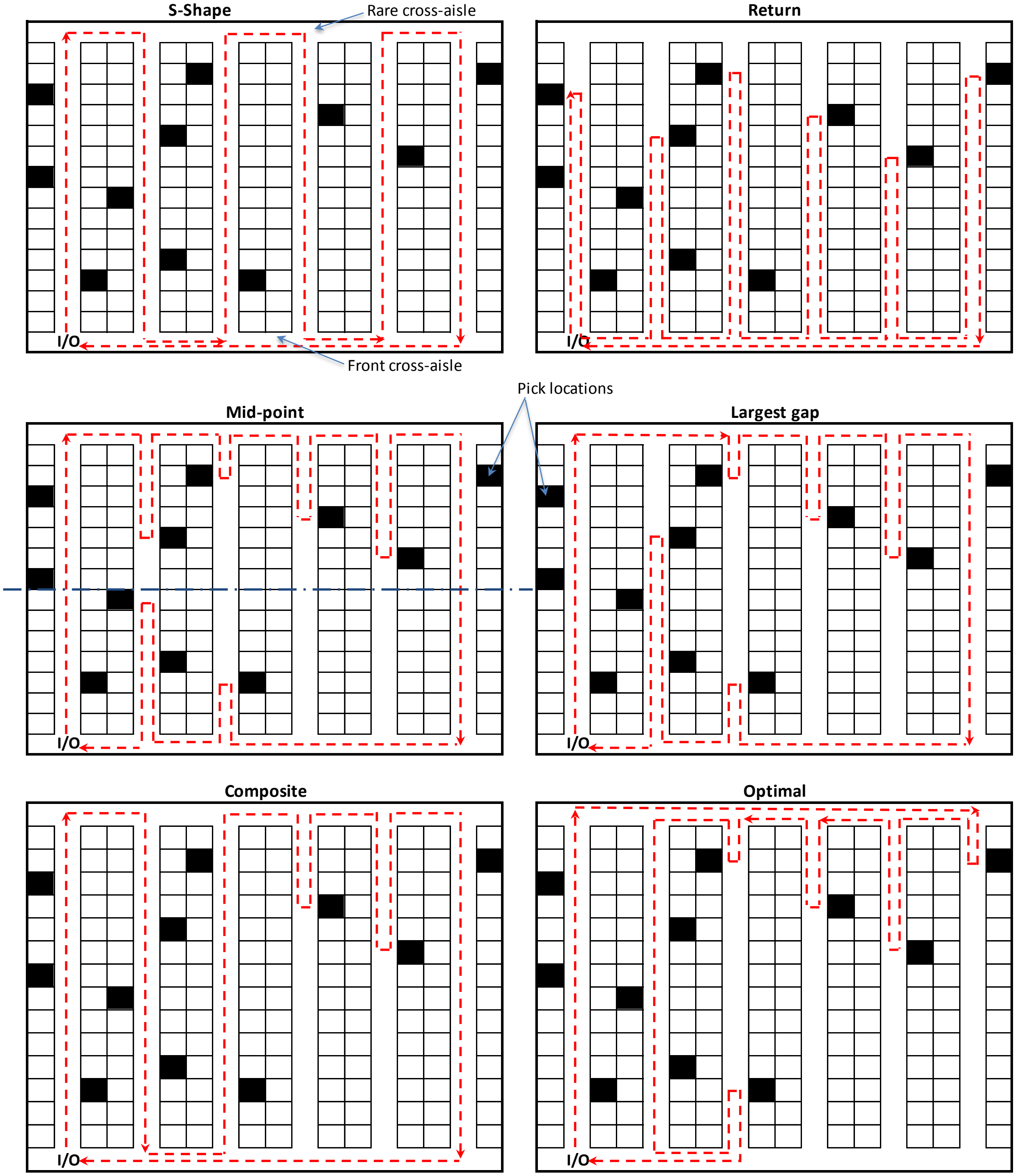


Figure 8 Example of routing strategies (De Koster et al., 2007)

3.4. SINGLE ORDER PICKING

Many researchers have examined the travel times for different picking policies. Picking policies encompass assigning SKUs or orders to pick list and subsequently retrieving them from their storage locations. Single order picking is one of the commonly used picking policies. The order pickers picks one order at a time. The order picker begins from the I/O-point, picks one item at a time and returns to the I/O-point after all the items in the assigned pick list are picked. So, every order is processed separately. We will apply this way of picking, as the order sizes we assume for this thesis is relatively large.

3.5. BATCHING

If the orders are small and we have enough picking capacity, a set of orders can be picked in a single picking tour. This way of picking is known as order batching or simply batching. It is a method of grouping a set of orders into a number of sub-sets, and subsequently retrieved in a single picking tour. Order batching is an NP-hard problem. That is why many researchers focus on developing heuristic methods for solving it. Seed and time saving algorithms can be taken as an example. In seed algorithms the batches are constructed in two steps: seed selection and order congruency. We first define a seed order for each batch applying seed selection rules. De Koster, Van der Poort, and Wolters (1999) provides us some seed selection rules. Some examples of seed selection rules are random order, an order with large number of positions, an order located furthest from the I/O-point, and an order with longest pick tour. Order congruency rules then determine which unassigned orders should be added next into the current batch. Examples of order congruency rules are the number of additional aisles which have to visit and the difference between the gravity center of the order and the gravity center of the seed order. The researchers show that this batching of orders has a significant impact on the performance of the order picking systems. Saving algorithms on the other hand are based on the algorithm of Clarke and Wright (1964) for the vehicle routing problem. A saving on the travel distance can be obtained by combining a set of small tours into a smaller set of large tours.

Order batching and the next zoning picking policies are not included in this thesis. We will apply the single order picking policy.

3.6. ZONING

It is also possible to divide the order picking area into zones. Each SKUs belonging to the same product group are then stored close to each other. The order picker is then assigned to pick the part of the order that is in his assigned zone. Compared to batching, the zoning does not have a significant impact on the performance of order picking systems (De Ruijter et al., 2009). Zoning might be advantageous as each order picker only needs to traverse a smaller area. Other advantages of zoning include reduce aisle congestion and also order pickers become familiar with the item locations in the zone. Aisle congestion occurs when an aisle is filled with frequently demanded items and only a single order picker at a time can access the aisle. The main drawback of zoning is that orders are split and must be consolidated again before they are shipped to the customer.

4. Experimental design: Experimental assumptions, factors, and ranges

We present the assumptions made while conducting the experiments in section 4.1. In section 4.2, experimental factors along with ranges are provided.

4.1. Assumptions made

The assumptions made in this thesis do not represent an actual warehouse, but the factors chosen allow us to conduct the experiments in a reasonable time.

Warehouse layout consideration: The warehouse considered in this thesis is a fixed two-dimensional rectangular warehouse with one Input and Output point (I/O-point) located at the extreme lower left hand corner. The warehouse consists of 800 pick locations and twenty aisles running between the North and South walls with 40 pick locations per aisle. Each aisle is two-sided and has 20 pick locations on each side. Two end-aisles are located near and parallel to the North and South walls of the warehouse, respectively. We assume there is only one level of rack. The width of the picking aisles is assumed to be one unit. This unit-load warehouse assumption eliminate the complexity of our model. In unit-load warehouse SKUs arrive on pallets and leave on pallets. And both the replenishing and picking are unit-load processes. The space and labor are then simple linear models, i.e., it takes about *n* times the space or storage locations to store *n* pallets; and it takes about *n* times the labor to pick *n* pallets. Figure 14 in Appendix 2 shows the layout of the warehouse.

Set of orders: It is assumed the warehouse stocks a total of 100 SKUs. Using these 100 SKUs, 160 orders of varying sizes are generated randomly. The order size varies between 3 and 7 units and generated from a normal distribution. While generating the second order sets, we control 20% of SKUs to appear in 80% of the orders. This also randomly generated from a normal distribution.

Single order picking: We assume the order picker picks only one order at a time. He or she begins from the I/O-point picks one item at a time and returns to the I/O-point after all the items in the assigned pick list are picked. The order picker travels up and down each aisle until the entire order is picked. The routing strategy considered in this thesis is presented below.

Routing strategy: For simplicity, we restrict the routing strategy to be the S-shaped routing policy. Under this policy, the order picker begins by entering the pick aisle closest to the I/O-point. Any aisle containing at least one item is traversed over the entire length. No backtracking is permitted in any aisle. Aisles with no picks are skipped. After picking the last item, the order picker takes the shortest route back to the I/O-point.

4.2. Experimental factors and ranges

The focus of this thesis is on determining the conditions under which alternate storage assignment strategies minimize the total distance traveled by the order pickers. We conduct experiments that consider two experimental factors: storage assignment policy and percentage of SKUs in an order set. The idea is to show the conditions under which specific storage assignment policies might be preferred over others. For example, if a subset of SKU appears in numerous orders, a storage assignment strategy that allocates these SKUs to nearby locations might minimize the total pick times. If all the items appear uniformly in all the orders, a random storage policy might be preferred, and so on. The only changeable variables in our model are the storage assignment strategies and the percentages of SKUs in an order set. We consider four storage assignment strategies and two percentages of SKU that appear in an order sets. This results in a total of 8 different combinations. Figure 9 shows the combination relationship of the two experimental factors.

Experimental Factor 1. Storage assignment strategies: We implement four storage assignment policies - random, class-based, cube-per-order index based on popularity and order oriented slotting (OOS) policies. All but OOS are frequently used in warehouses. The objective of this thesis is to identify the conditions under which OOS might outperform the first three. Several methods are available to implement OOS. We adopt the interaction frequency heuristic (IFH), which is an advanced order oriented slotting method. Next, we briefly present the basic idea behind the above mentioned storage policies. More details can be found in the literature review section.

In a random storage policy, each incoming pallet is randomly assigned to an empty location in a warehouse. Each empty location has the same probability of being selected for storage.

In class-based storage policy, the most frequently requested item is assigned to the closest location to the I/O-point. Items are first divided into three classes – A, B and C based on Pareto's law. This law states that a small percentage of items generates a large number of transactions. These items belong to Class A. Conversely, a large percentage of the items generate very few transactions. These items belong to Class C. The remaining items belong to Class B. Each class is then assigned to a dedicated area of the warehouse based on the number of transactions it generates. Generally, Class A items are closest to the I/O-point and Class C the farthest. Storage of items belonging to a class within the designated area is done randomly.

Under cube-per-order index (COI) based on popularity storage assignment policy, the basic idea is to store SKUs that are frequently requested closest to the I/O-point. In other words, SKUs with highest popularity (f_{i0}) will be stored closest to the I/O-point. Where popularity is the number of orders that require a particular SKU. This policy ranks SKUs in a non-increasing order of their popularity and storage locations in a non-decreasing order of their distance to the I/O-point. It then performs a one-to-one matching of the SKUs in the two rank ordered lists.

The interaction frequency heuristic of OOS policy on the other hand allocates pairs of SKUs appearing in multiple orders in adjacent locations. As mentioned previously, it can be implemented in multiple ways, one of which is the interaction frequency heuristic (IFH) method (Mantel et al., 2007). The interaction frequency for a pair of items is defined as the number of orders in which the two items appear. Items with high interaction frequency are stored in adjacent locations. In addition, frequently requested pairs of items with high interaction frequency are stored close to the I/O-point.

The second experimental factor is briefly discussed below.

Experimental Factor 2. Percentage of SKU's appears in an order: We not only vary storage assignment policies in this thesis, but also the percentage of SKUs that appear in an order set. The idea is to vary the subset of SKUs appearing in orders and show the conditions under which the total distance traveled is minimized. Thus, we will compare the results when all items appear uniformly in all orders (i.e., when the probability of appearing in an order is equal for all items), and 20% SKUs appear in 80% of the orders.

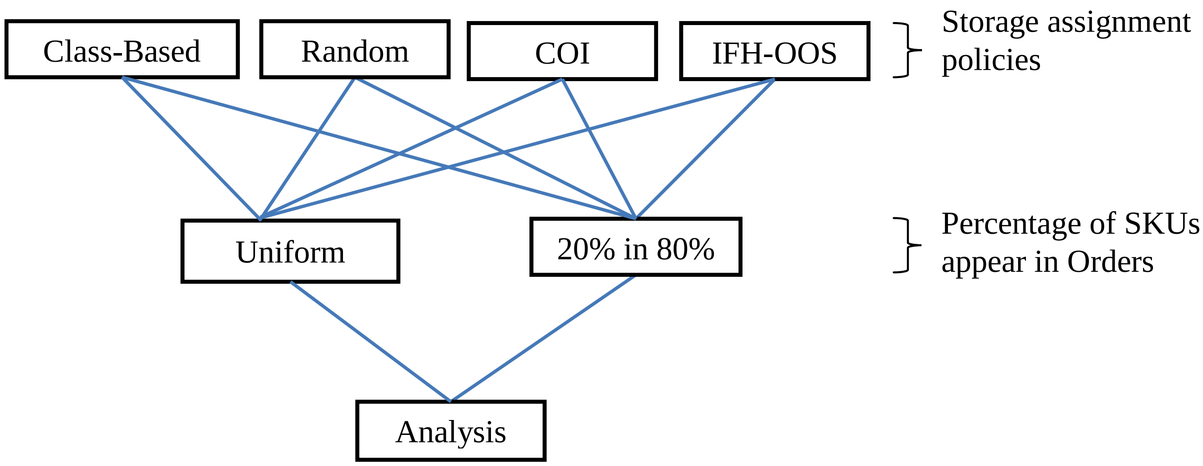


Figure 9 Combination relationship of the two experimental factors

5. COMPUTATIONAL RESULTS

The following chapter is about the computational results that could be obtained using the Monte Carlo simulation study. At the beginning, a small example is presented in order to give the reader an insight into the general ideas behind this thesis. In section 5.2, the majority of the computational results for our initial settings are presented and explained in detail. First, is the exploration phase with the various policies; next in section 5.3, we compare the alternate policies.

5.1. SMALL EXAMPLE

In this section, we provide a small example that is easy to explain and will give a reader an insight into the general ideas behind this thesis.

Let us assume, we have an empty fixed two-dimensional rectangular warehouse with one I/O-point located at the extreme lower left hand corner. The warehouse consists of 30 pick locations and 3 aisles running between the North and South walls with 10 pick locations per aisle. Each aisle is two-sided and has 5 pick locations on each side. Two end-aisles are located near and parallel to the North and South walls of the warehouse, respectively. It is also assumed that there is only one level of racks. Figure 10 shows the layout of the warehouse. Furthermore, we assume that the warehouse stocks a total of 10 SKUs. Every SKU has an average of 3 items to fully stock the warehouse. For instance, SKU 1 has 5 items, SKU 2 has 3 items, and SKU 10 has only 1 item, and so on. Using these 10 SKUs, 10 orders of varying sizes are generated. Table 1 shows a stable set of orders that needs to be picked. The order size varies between 1 and 5 units. For example, order 1 has five items and so its order size is 5. However, orders 7 and 8 have only a single item. We assumed that the order picker picks only one order at a time. In other words no batching of orders is permitted. Also, the process of replenishing and picking is just a unit-load approach. As a final point, we restrict the routing policy to be the *S-shaped (traversal) heuristic*. It is the most common heuristic and often applied in practice.

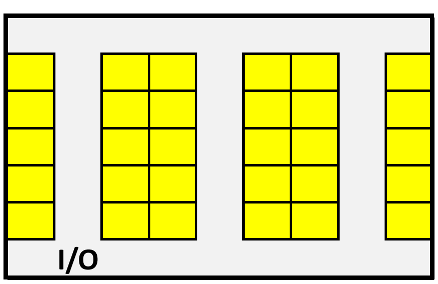


Figure 10 Empty rectangular warehouse for small instance example

Order	Order ID	SKUs				
1	1001	1	2	3	4	5
2	1002	1	6	7	8	
3	1003	3	2	8		
4	1004	1	4	5		
5	1005	5	6	7		
6	1006	1	5	6	7	
7	1007	9				
8	1008	10				
9	1009	1	2	3	4	
10	1010	5	4			

Table 1 Set of orders to be picked for small example

The next step is to store items in the warehouse according to random, COI based on popularity, IFH-OOS, and class based (ABC) storage assignment strategies. The objective is to determine the conditions under which alternate storage assignment strategies help minimize the total distance travelled by the order pickers to pick all orders.

Let us first consider the random storage assignment policy. Under this policy, each item can be randomly assigned to an empty location in a warehouse. Where each empty location has the same probability of being selected for storage. Hence, for this small example items 2 and 7 might take the first two closest positions to the I/O-point, while, items 4 and 7 could be assigned the farthest locations from the I/O-point. Figure 11 (a) shows the locations of items when we apply a random storage assignment strategy. The numbers in the square box represent the items stored in the locations.

For COI based on popularity strategy, we should first determine the popularity of each SKU and rank these popularities in a non-increasing order. The popularity is the number of orders that requires a particular SKU. For this small example, we can simply count the number of times each item of a SKU is requested from the set of orders table. So, for example SKU 1 has 5 popularity. That is 5 items of SKU 1 are requested in the order set. On the other hand, SKUs 9 and 10 have 1 popularity. Accordingly, SKU 1, 5, 4, 2, 3, 6, 7, 8, 9, and 10 are ranked in a non-increasing order of their popularity. Next, the minimal distances of the locations to the I/O point have to be determined. Once we determined the minimal distances, the next step is to assign SKUs with a higher popularity to a locations closest to the I/O-point. That is why, SKU 1 has given the prime locations to the I/O-point and SKUs 9 and 10 are assigned the farthest locations. We did not apply the original COI policy, because as per our assumption every SKU will have the same cube-per-order index of one. That is the ratio of the number of storage locations a SKU requires (S_j) to the throughput of that SKU (T_j) is always one. That way we will not able to make a difference between COI's of SKUs. So we used the above approach of COI based on popularity. Figure 11 (b) shows the locations of items according to COI based on popularity storage assignment strategy.

On the other hand, IFH-OOS policy allocates pairs of SKUs appearing in multiple orders in adjacent locations. In addition, frequently requested pairs of SKUs with high interaction frequency are stored close to the I/O-point. The detailed procedure of this storage assignment strategy can be found in section 3.2.2.1. First, we determine and allocate "*singles*" (SKUs that never share an order with other SKUs) to their COI-locations. For this small example, SKU 9 and 10 are singles and take their COI-locations accordingly. Next, we sort interaction frequencies f_{ij} in a non-increasing order and consider each f_{ij} step by step and assign SKUs. For this small example, 2 and 3 are the SKUs which have the highest interaction frequency of 3. So, we choose the storage locations of SKU 2 and SKU 3 in such a way that the *routing-specific* distance (d_{ij}^r) is minimal. In the same way, for SKU 1 and SKU 4 we choose the next locations with minimum routing-specific distances. SKU 1 and SKU 5 also have an interaction frequency of 3. However, SKU 1 has already been allocated. So, we only need to assign storage location for SKU 5 and we did that. We apply the same procedure to assign the remaining SKUs. Figure 11 (c) shows the locations of items according to IFH-OOS storage assignment strategy.

With ABC-storage assignment strategy, the idea is to allocate the most frequently requested items to the closest storage locations of the I/O-point. To do so, we first develop the three classes A, B, and C. Here the assumption is that approximately 20% of the items belong to Class A. The next 40% of the items belongs to class B and the remaining 40% of the items go to class C. We build this based on the cumulative percentage of the number of times that a stock keeping unit appeared in an order sets. For this small example, SKU 1 fit in class A; SKU 2, SKU 4, and SKU 5 belongs to class B. The remaining SKUs will be class C. Each class is then assigned to a dedicated area of the warehouse based on the number of transactions it generates. For instance, class B requires twelve dedicated storage locations. Also remember that, storage of items belonging to a class within the designated area is done randomly. Figure 11 (d) shows the locations of items according to ABC-storage assignment strategy.

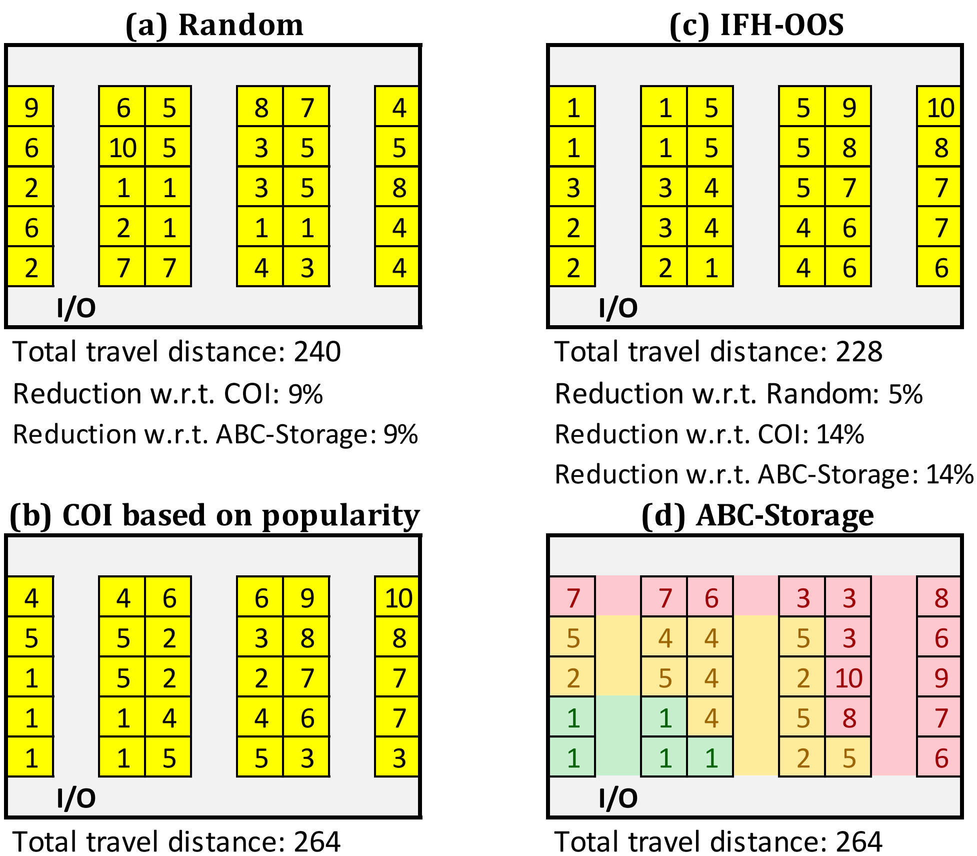


Figure 11 Comparison of storage assignment strategies for a small example

Up till now, we allocate items to the given warehouse storage locations with the above mentioned storage assignment strategies. We assign several locations to one SKU depending on the number of items that particular SKU consists of. The next step is to pick the set of orders and calculate the total travel distance. To do so, we assumed the order picker picks only one order at a time. He or she begins from the I/O-point picks one item at a time and returns to the I/O-point after all the items in the assigned pick list are picked. He or she will visit every locations of a particular SKU. And the first item closest to the I/O-point of that particular SKU in the assigned pick list will be picked for the first order that has this SKU. The next item closest to the I/O-point goes to the second order that has this SKU, and so on. Besides, a single aisle might contain all the items that are required for an order. In this case, it would be an optimal solution if we pick those items from the single aisle. But, we will not do that unless those items are the one close to the I/O-point of their specific SKU group. Furthermore, the order picker is restricted to follow S-shaped routing policy. Under this policy, the order picker begins by entering the pick aisle closest to the I/O-point. Any aisle containing at least one item is traversed over the entire length. No backtracking is permitted in any aisle. Aisles with no picks are skipped. Once the last item is picked, the order picker takes the shortest route back to the I/O-point. After applying these rules, the computed total travel distance of each storage assignment strategies are shown above in Figure 11.

The result shows, IFH-OOS has the highest reduction in the total distance travelled by the order pickers. It has 14% reduction in total travel distance with respect to COI based on popularity and ABC-storage and 5% reduction with respect to random policy. Random storage assignment strategy also has 9% reduction of total travel distance with respect to COI based on popularity and ABC-storage strategies. This result is for the above order sets and we expect different output for different set of orders. In the next section, we present the effect of percentage of SKUs that appear in an order set.

5.2. RESULTS

In this section, we present the computational results for the experimental assumptions made in section 4.1.

As we stated in section 4.2, two experimental factors were considered: storage assignment policy and percentage of SKUs that appear in an order set. We implement four storage policies: Random, COI based on popularity, IFH-OOS, and ABC-storage. We not only vary storage assignment policies, but also the percentage of SKUs that appears in an order set. That is, when all items appear uniformly in all orders and approximately 20% of SKUs appear in 80% of the orders. Thus, for all four storage assignment policies, we will first present the computational results when the order sets randomly generated from a normal distribution. In which the probability of appearing in an order is equal for all SKUs. Afterward, the results when 20% of SKUs appear in 80% of the orders will be presented and explained in detail.

5.2.1. EXPERIMENT 1 RESULTS

In this section, the computational results when the order sets randomly generated from a normal distribution are presented and explained in detail. Basically the same procedure with extension as the above small example is followed.

So to begin with, we randomly generate 160 orders of varying sizes using 100 SKUs. The order size varies between 3 and 7 units and generated from a normal distribution. The average interaction frequency under this experiment is 1.8. We don't have singles (SKUs that never share an order with other SKUs), because as per our assumption the order size varies between 3 and 7. That implies at least for a single time an SKU share an order with other two SKUs. We make use of these order sets for all four storage assignment strategies that we implement in this research. Then, we apply the methodologies described in section 3.2.

For a *random storage assignment strategy*, we first randomly shuffle SKUs before storing them in the warehouse locations. Because, according to this storage assignment strategy each SKU has to be randomly assigned to an empty location in a warehouse. Where the empty locations have the same probability of being selected for storage.

For a *COI based on popularity storage assignment strategy*, we first compute and record the popularity of each SKUs. All SKUs on the list are then ranked in a decreasing order. Next, we determine the minimal distances of the locations to the I/O point. The distances between the I/O-point and pick locations are measured taking into account the aisle structure of the picking area. Once we determined the minimal distances, the next step is to assign SKUs with a higher popularity to the prime locations closest to the I/O-point.

Then for an *IFH-OOS policy*, we begin by determining the popularities (f_{i0}) and interaction frequencies (f_{ij}) of all SKUs and store the computed values in a matrix. After that, we determine the routing-specific distances between two pick locations and store these in a distance matrix. We then determine the locations of all SKUs by applying COI based on popularity storage assignment strategy. The next step of this storage assignment strategy is to identify *singles* (i.e., SKUs that never share an order with other SKUs) and assign these SKUs to their COI-locations. However, we don't have singles because the order size that we assume varies between 3 and 7 units. Which implies at least for a single time an SKU will share an order with other SKUs. Following the detail steps in section 3.2.2.1, we sort the positive interaction frequencies in a non-increasing order. We then consider each f_{ij} step by step and assign SKUs to the warehouse storage locations. Furthermore items that belong to a specific SKU will be stored next to each other since we consider multiple items per SKU to fully stock the warehouse. While considering each f_{ij} and if SKU *i* and SKU *j* have already been assigned, then we process the next interaction frequencies in the following two situations. First when neither SKU *i* nor SKU *j* has been assigned so far and second when either SKU *i* or SKU *j* has been allocated so far. For both cases we follow the same procedure, except for the latter case the location of one SKU is already fixed. We create set A_i and A_j in which allowed locations can be stored for SKU *i* and SKU *j* respectively. We add COI-locations of each SKU to their respective sets and check the distances from the COI-location to the I/O-point for both SKUs. We separately consider each set and also add a free location *x* to the sets for which holds that $(1-\beta)\, d_{i0}^{r} \leq d_{x0}^{r} \leq (1+\beta)\, d_{i0}^{r}$ where β is a certain factor that is assumed to be >0 and <<1 and d_{i0}^{r} is the routing-specific distance between center of gravity of the locations of SKU *i* and the I/O-point. After each set has been filled with the allowed locations, we choose the locations of both SKUs from these sets in such a way that the routing-specific distance between center of gravity of the locations of SKU i and SKU j (d^r_{ij}) is minimal. After processing all $f_{ij}'s$, we found out that some SKUs are remain unassigned, because their allowed locations are already occupied by other SKUs. So, we sort the popularities of unassigned SKUs in non-increasing order and determine free locations. We then store the remaining unassigned SKUs based on the COI based on popularity storage assignment strategy. In brief we assign pairs of SKUs that appear in multiple orders close to each other and also frequently requested SKUs are stored close to the I/O-point.

When we come to *ABC-storage assignment strategy*, we first classify three classes A, B, and C based on Pareto's law. This law states that a small percentage of items generates a large number of transactions. So as to categorize the classes we first sort the number of times that an SKU appears in an order sets in a descending order and sum them up. Next, we compute the cumulative amount of the number of times each SKU appears in set of orders. After that the cumulative percentage is calculated as cumulative amount of each SKU divided by the total amount that all SKUs appeared in the order sets. We breakdown ABC classes in such a way that approximately 20% of the items as class A, the next 40% of the items as Class B, and the remaining percentage of items as Class C based on the cumulative percentage. We then dedicate an area in the given warehouse for each classes based on the number of transactions they generate. Items in a class are then assigned to the dedicated area of the warehouse. Storage of items belonging to a class within the designated area is done randomly.

Thus far, we store items to the given storage locations with the above mentioned storage assignment strategies. We assign several locations to one SKU depending on the number of items that particular SKU consists of. The next step is to pick the set of orders that has been generated from normal distribution and calculate the total travel distance. In general for routing we need to solve a combinatorial problem. So, to determine the feasible solution that is as good as possible and save computational time we apply the following heuristic. First of all, we assumed the order picker picks only one order at a time. He or she will visit every location of a particular SKU because of our unit-load assumption. Then the first item closest to the I/O-point of that specific SKU in the assigned pick list will be picked for the first order that has this SKU. The next item closest to the I/O-point goes to the second order that has this SKU, and so on. This is a reasonable assumption as most order pickers would like to pick an item which is close to the I/O-point. If a single aisle contains all the items that are requested for an order, it is an optimal solution if the order picker picks those items from that single aisle. However, we cannot do that unless and otherwise those items are the one close to the I/O-point of their particular SKU. Besides, the order picker is restricted to follow the most commonly used S-shaped routing policy.

Under this experiment settings, we expect the random storage assignment strategy help minimize the total distance traveled by the order pickers to pick all orders. Because, in order to save the computational time we used small number of orders. And so, the values of the interaction frequencies between SKUs are low. In other words SKUs are loosely coupled. That favors the random policy. However, if we have larger number of orders, the values of the interaction frequencies will be higher and the other storage assignment rules might outperform the random policy.

Finally, all storage assignment strategies are executed for thirty times in visual basic for applications and so the average total travel distances are provided in table 2.

Storage assignment strategy	Number of replications	Average total travel distance (units)	Standard deviation of distance
Random	30	28,442	401.0
COI based on popularity	30	31,948	296.9
IFH-OOS	30	31,096	255.1
ABC-Storage	30	30,596	407.2

Table 2 Results of average total travel distance (when order sets randomly generated from a normal distribution)

As we expected, the result shows a random storage assignment strategy has the highest reduction in the total distance travelled by the order pickers. The next section explains how IFH-OOS strategy outperforms all the other frequently used storage assignment strategies.

5.2.2. EXPERIMENT 2 RESULTS

In this section, the computational results when 20% of SKUs appear in 80% of the orders are provided. To do so, we manipulate the order generation in such a way that approximately 20% of SKUs to appear in 80% of the orders. With the intention that, the average interaction frequency between SKUs will have a relatively large value and favor the interaction frequency based storage assignment strategy. The average interaction frequency in number is 3.9. We don't have singles. The number of orders and order size are similar to experiment 1. And they are generated from a normal distribution.

Following the generation of the order sets, we apply the same methodology as described above in order to calculate the average total travel distance. Besides, all storage assignment strategies are executed for thirty times and the average total travel distance of this case is provided in table 3. Then the order picking process is done in a similar fashion with that of experiment 1.

Under this experiment, we expect IFH-OOS strategy to outperform all the other frequently used storage assignment strategies. It is because, firstly the average interaction frequency has a relatively large value, and secondly there are no singles in the order sets.

Storage assignment strategy	Number of replications	Average total travel distance (units)	Standard deviation of distance
Random	30	25,583	553.3
COI based on popularity	30	29,630	415.9
IFH-OOS	30	24,251	275.8
ABC-Storage	30	28,958	400.5

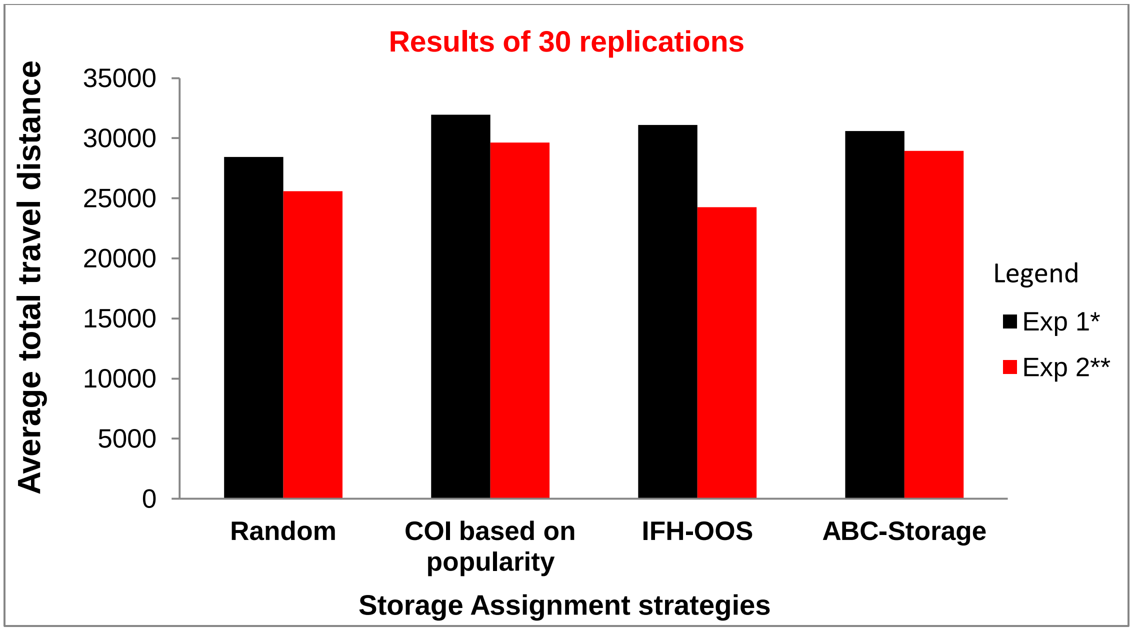
Table 3 Results of average total travel distance (when 20% of SKUs appear in 80% of the orders)

As we projected, the result shows IFH-OOS strategy has the highest reduction in the total distance travelled by the order pickers. It outperforms all the other frequently used storage assignment strategies: Random, COI based on popularity, and ABC-Storage.

5.3. COMPARISON OF RESULTS

In this section, the results that have been presented in detail earlier are again summarized in a compressed form which enables the reader to clearly visualize the outputs. We compare storage assignment strategies against each other according to the percentage of reduction in total distance travelled by the order pickers and ease of implementation.

Under the first experiment, the result shows a random storage assignment strategy has the highest reduction in the total distance travelled by the order pickers. Based on our initial settings, it has 11% reduction in total distance travelled with respect to COI based on popularity policy. Similarly, it has 9% and 7% reduction with respect to IFH-OOS and ABC-storage respectively. On the other hand, in the second experiment IFH-OOS has the highest percentage of reduction compared to other frequently used storage assignment strategies. As we have already explained, this is because the average interaction frequency between SKUs has a relatively large value. Again for our initial settings, the result shows that IFH-OOS has 18% reduction in total distance traveled with respect to COI based on popularity storage policy. It even has almost 5% reduction with respect to random policy.



Exp 1*: When the probability of appearing in an order is equal for all items
Exp 2**: While generating order sets, we control 20% of SKUs to appear in 80% of Orders

Figure 12 Results of average total travel distance for 30 replications

As a final point, the interaction frequency heuristic method of an order oriented slotting policy seems a logical approach that every single warehouse has to implement for lowering their operating expenses, but the complexity of the procedure compared to the other frequently used storage assignment strategies might be a problem to execute in a short period of time. On the other hand, if we consider class-based storage it will be easy to use and simple to understand by a warehouse manager.

6. CONCLUSIONS AND RECOMMENDATIONS

In this chapter, all the relevant conclusions and findings revealed in the previous chapters are summarized. The conclusions provided here are based on the computational experiments. Furthermore, we provide a set of recommendations that will further used to improve the order picking efficiency which plays a vital role to reduce supply chain costs and improve the productivity in the warehouse.

6.1. CONCLUSIONS

There are several conclusions that emerged throughout this research that should be mentioned in this section. First of all, order picking at a fictitious warehouse was the field of the research. Receiving, transfer and put away, order picking, cross-docking, and shipping is the main warehouse activities. Among these activities, order picking is the most labor-intensive and costly activity of most warehouses, approximately 55% of the total warehouse operating expenses are related to order picking operations. That is why improving the order-picking efficiency is indispensable. It plays a vital role to reduce supply chain costs and improve productivity in the warehouse. Either humans or machines can be used for order picking. The majority of warehouses prefer to use humans for order picking. Setup, travelling, searching, and picking are the main activities that an order picker has to perform while retrieving items from storage areas in response to a specific customer request. Notice that travel is the dominant component. Travelling is a waste, as it costs labor hours without adding value to the product and so it is the first candidate for improvement. Hence, the main objective of the research was to determine the conditions under which alternate storage assignment strategies help minimize the total distance traveled by the order pickers. In order to accomplish this objective, we formulate two research questions. The first research question focuses on identifying the storage assignment strategies which have a major impact on order picking efficiency.

We performed a literature review to answer the first research question. In general, previous order picking research focuses on one of the following three areas: picking policies, storage policies, and routing policies. Many researchers have looked the common picking policies for travel time estimates. Mostly order batching (i.e. grouping a set of orders into a number of sub-sets) and subsequently retrieve from their storage locations has a significant role in reducing travel times. However, in our research we assumed the order size is relatively large to be picked individually. This way of picking is often known as the *single order picking* policy. Next, we carry out literature research regarding storage assignment strategies. A storage assignment policy is a set of rules which can be used to assign SKUs to storage locations. Heskett (1963) first introduced the well-known cube-per-order index (COI) storage assignment policy, which is aimed to store items that are frequently requested and require smaller storage space closest to the I/O-point. COI storage policy is frequently used by warehouses. Our COI based on popularity storage assignment policy store SKUs with the highest popularity closest to the I/O-point. Other frequently used storage policies include random, dedicated and class-based. In random storage policy, each incoming pallet is randomly assigned a storage location in the warehouse that is arbitrarily selected from all eligible empty locations with equal probability (Petersen, 1997). In a dedicated storage policy each SKU is assigned a specific storage location(s). In between random and dedicated, a class-based (ABC) storage policy assigns the most frequently requested SKUs closest to the I/O-point. Brynzér and Johansson (1996) introduce the family grouping method which requires the statistical correlation between items. Thus, items can be stored close to each other if they appear together in an order. Mantel et al. (2007) introduce the approach called order oriented slotting (OOS) policies. OOS policy allocates pairs of items appearing in multiple orders in adjacent locations. Several methods are available to implement OOS. Mantel et al. (2007) introduce interaction frequency heuristic (IFH) and interaction frequency based quadratic assignment heuristic (IFH QAP). Following that, De Ruijter et al. (2009) develop order oriented product swapping (OPS) method, in which they tackle the very difficult combinatorial optimization OOS problem directly by using simulated annealing. Under the interaction frequency heuristic, items with high interaction frequency are stored in adjacent locations. Since the interaction frequency heuristic provided by Mantel et al. (2007) assumes each SKU has only one storage location we extend into one SKU to have more than one storage locations. The IFH QAP heuristic on the other hand assures frequently ordered SKUs placed together and also fast movers to be stored close to the I/O-point. After that, routing policies were also reviewed which has a significant effect on the order picking efficiency. In routing policies one can determine the picking sequence of SKUs on the pick list. These policies range from simple heuristics (De Koster et al., 2007) to optimal procedures (Ratliff & Rosenthal, 1983). In practice, mostly heuristics are used and yield nearly optimal solutions. For the reason that many warehouses have different layout and an optimal algorithm is not available for ever layout. S-shaped (traversal) heuristic is the most common and simplest heuristic used in practice. We implement this routing policy.

Following the literature review four storage assignment strategies are selected and implemented. The selected storage assignment strategies are random, COI based on popularity, IFH-OOS, and ABC-storage. These strategies have a significant impact on the order picking efficiency. The next step was investigating the conditions under which alternate storage assignment strategies help minimize the total distance traveled by the order pickers. We provided a small example that gives a reader an insight of the general ideas behind this thesis. We then made assumptions regarding the warehouse layout and the number of SKUs that the warehouse stocks. The assumptions we made might not represent an actual warehouse, but the factors chosen allowed us to conduct the experiments in a reasonable time.

Finally, we develop a Monte Carlo simulation model using VBA in order to identify the most efficient storage assignment strategy. Storage assignment strategies were not the only experimental factors that we vary, but also the percentage of SKUs that appear in an order set. Our first experiment was conducted, when all items appear uniformly in all orders (i.e., when the probability of appearing in an order is equal for all items). The order size varies between 3 and 7 units and randomly generated from normal distribution. The average interaction frequency is 1.8.

Under this experiment, the result shows a random storage assignment strategy has the highest reduction in the total distance travelled by the order pickers. When we conduct our second experiment, we control the order generation in such a way that approximately 20% of SKUs to appear in 80% of the order sets. With the intention that, the average interaction frequency between SKUs will have a relatively large value and favor the interaction frequency based storage assignment strategy. Thus, under this experiment, the result shows IFH method which is one of the complex OOS policy, outperforms all the other frequently used storage assignment strategies. Furthermore, a warehouse manager could use these as a decision support tool for lowering operating expenses through implementing the right storage assignment strategy in different situations.

6.2. RECOMMENDATIONS

In this section, we provide important recommendations that should be taken into account by a particular warehouse based on the conclusions found throughout the internship project.

We recommend a particular warehouse management team to implement the suggested storage assignment strategies on different situations. For instance, we advise a warehouse manager to implement a random storage assignment strategy if the SKUs in a specific set of orders are loosely coupled. On the other hand, we recommend to implement interaction frequency heuristic method of order oriented slotting policy if the average interaction frequency between SKUs in an order sets has a relatively large value and if there are not too many *singles*. IFH-OOS policy allocates pairs of items appearing in multiple orders in adjacent locations. Besides, frequently requested pairs of items with high interaction frequency are stored close to the I/O-point. So, applying the suggested storage assignment strategies will ensure improving the order picking efficiency. Improving the order picking efficiency by reducing the travelling distances plays a vital role in order to reduce supply chain costs and improve productivity in the warehouse.

7. SUGGESTIONS FOR FUTURE WORK

Although it was tried to conduct this research as thoroughly as possible due to time and other constraints perhaps not all the relevant issues could be treated. Generally we tried to show how to improve the order picking efficiency through the implementation of storage assignment strategies. We also determined the conditions under which alternate storage assignment strategies help minimize the total distance traveled by the order pickers. However, the warehouse layout that we considered in this thesis is just a fixed two-dimensional warehouse with one I/O-point located at the extreme lower left hand corner. In addition, there is only one level of racks. Therefore, we suggest considering a different warehouse layout and figuring out the effects on the order picking efficiency. And we also suggest increasing the problem size and incorporating more realistic issues. That would be very interesting but challenging problem. On the other hand, we restrict the routing strategy to be the S-shaped routing policy. Some of the storage assignment strategies that we implement in this research depend on the routing policies. Thus, it would be interesting to use different routing policy and see the effect on the storage assignment strategies. As a final point, in this research we tried to control the percentage of SKU that appear in an order sets. We compare the results when all items appear uniformly in all order sets, and approximately 20% of SKU appear in 80% of the order sets. The latter favor the interaction frequency based storage assignment strategies, since the average interaction frequency between SKU will have a relatively large value in this situation. So, we suggest varying the percentage of SKU that appear in an order sets which will favor the other storage assignment strategies.

8. Reflection

My master thesis and an internship at the Logistics and Distribution Institute (LoDI), University of Louisville in Kentucky, USA was a very useful experience for me. And I am glad that I was given the chance to participate in a very interesting research on the field of warehouse. Improving the order picking efficiency through storage assignment strategy in my opinion is very essential for the companies that want to reduce supply chain costs and improve productivity. Though, it would be more interesting if I have got a practical project as in my future career I would like to focus on this sphere. Also, I might had the chance to enlarge my contacts.

I have the feeling that I learned a lot through this master thesis and internship. I carried out a variety of tasks and was able to gather quite some experience with, in particular, the general use of visual basic for applications. And also, for instance, organizing a Skype meeting with my supervisors and summarize the results of the session. In general, I can say that I really liked to work at the LoDI and the colleagues I had there.

However, I also made some mistakes during the course of this project which I have to learn from for future projects. Firstly, I should always write down every assumption I made, every steps of the model development, small and big discoveries and even simple notes on minor issues as an electronic version, including the date of the creation in the file name. I wrote down mostly on paper, which made it difficult to fit together all the pieces that were needed for the report of this research.

Furthermore, I had a challenging task of developing the Monte Carlo simulation model in visual basic for applications. That was because I had no sufficient background in programming. However, through some reference materials and help of my friends I made it through. And I am thankful for those who helped me out.

Except for the above mentioned facts, which were the cause for a delayed submission of the report, I think I did a quite good job and hope that the project will be adapted for practical case with some changes and that it will help to reduce costs and improve productivity in a particular warehouse.

REFERENCES

Bartholdi, J. J., & Hackman, S. T. (2011). WAREHOUSE & DISTRIBUTION SCIENCE Release 0.94. *Retrieved July, 28*, 2011.

Brynzér, H., & Johansson, M. I. (1996). Storage location assignment: Using the product structure to reduce order picking times. *International Journal of Production Economics, 46*, 595-603.

Caron, F., Marchet, G., & Perego, A. (2000). Optimal layout in low-level picker-to-part systems. *International Journal of Production Research, 38*(1), 101-117.

Clarke, G., & Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568-581.

De Koster, R. (2004). How to assess a warehouse operation in a single tour. *Report, RSM Erasmus University, the Netherlands*.

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

De Koster, R., Van der Poort, E. S., & Wolters, M. (1999). Efficient orderbatching methods in warehouses. *International Journal of Production Research, 37*(7), 1479-1504.

De Ruijter, H., Mantel, R. J., Schuur, P. C., & Heragu, S. S. (2009). Order Oriented Slotting and the Effect of Order Batching for the Practical Case of a Book Distribution Center.

Eisenstein, D. D. (2008). Analysis and optimal design of discrete order picking technologies along a line. *Naval Research Logistics (NRL), 55*(4), 350-362.

Gue, K. R., Ivanović, G., & Meller, R. D. (2012). A unit-load warehouse with multiple pickup and deposit points and non-traditional aisles. *Transportation Research Part E: Logistics and Transportation Review, 48*(4), 795-806.

Gue, K. R., & Meller, R. D. (2009). Aisle configurations for unit-load warehouses. *IIE Transactions, 41*(3), 171-182.

Heragu, S. S. (2008). *Facilities design*: Iuniverse Inc.

Heragu, S. S., Du, L., Mantel, R. J., & Schuur, P. C. (2005). Mathematical model for warehouse design and product allocation. *International Journal of Production Research, 43*(2), 327-338.

Heskett, J. L. (1963). Cube-per-order index-a key to warehouse stock location. *Transportation and distribution Management, 3*(1), 27-31.

Johnson, M. E., & Brandeau, M. L. (1996). Stochastic modeling for automated material handling system design and control. *Transportation science, 30*(4), 330-350.

Lerher, T., Potrč, I., Šraml, M., & Tollazzi, T. (2010). Travel time models for automated warehouses with aisle transferring storage and retrieval machine. *European Journal of Operational Research, 205*(3), 571-583.

Mantel, R. J., Schuur, P. C., & Heragu, S. S. (2007). Order oriented slotting: a new assignment strategy for warehouses. *European Journal of Industrial Engineering, 1*(3), 301-316.

Petersen, C. G. (1997). An evaluation of order picking routeing policies. *International Journal of Operations & Production Management, 17*(11), 1098-1111.

Petersen, C. G., & Aase, G. (2004). A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics, 92*(1), 11-19.

Ratliff, H. D., & Rosenthal, A. S. (1983). Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. *Operations Research, 31*(3), 507-521.

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

Roodbergen, K. J., & De Koster, R. (2001b). Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research, 133*(1), 32-43.

Rouwenhorst, B., Reuter, B., Stockrahm, V., Van Houtum, G., Mantel, R., & Zijm, W. (2000). Warehouse design and control: Framework and literature review. *European Journal of Operational Research, 122*(3), 515-533.

Sarker, B. R., & Babu, P. S. (1995). Travel time models in automated storage/retrieval systems: A critical review. *International Journal of Production Economics, 40*(2), 173-184.

Tompkins, J., White, J., Bozer, Y., & Tanchoco, J. (2003). Facilities planning: Wiley, New Jersey.

APPENDIX 1 - GENERAL ORDER PICKING PROCESS FLOWCHART

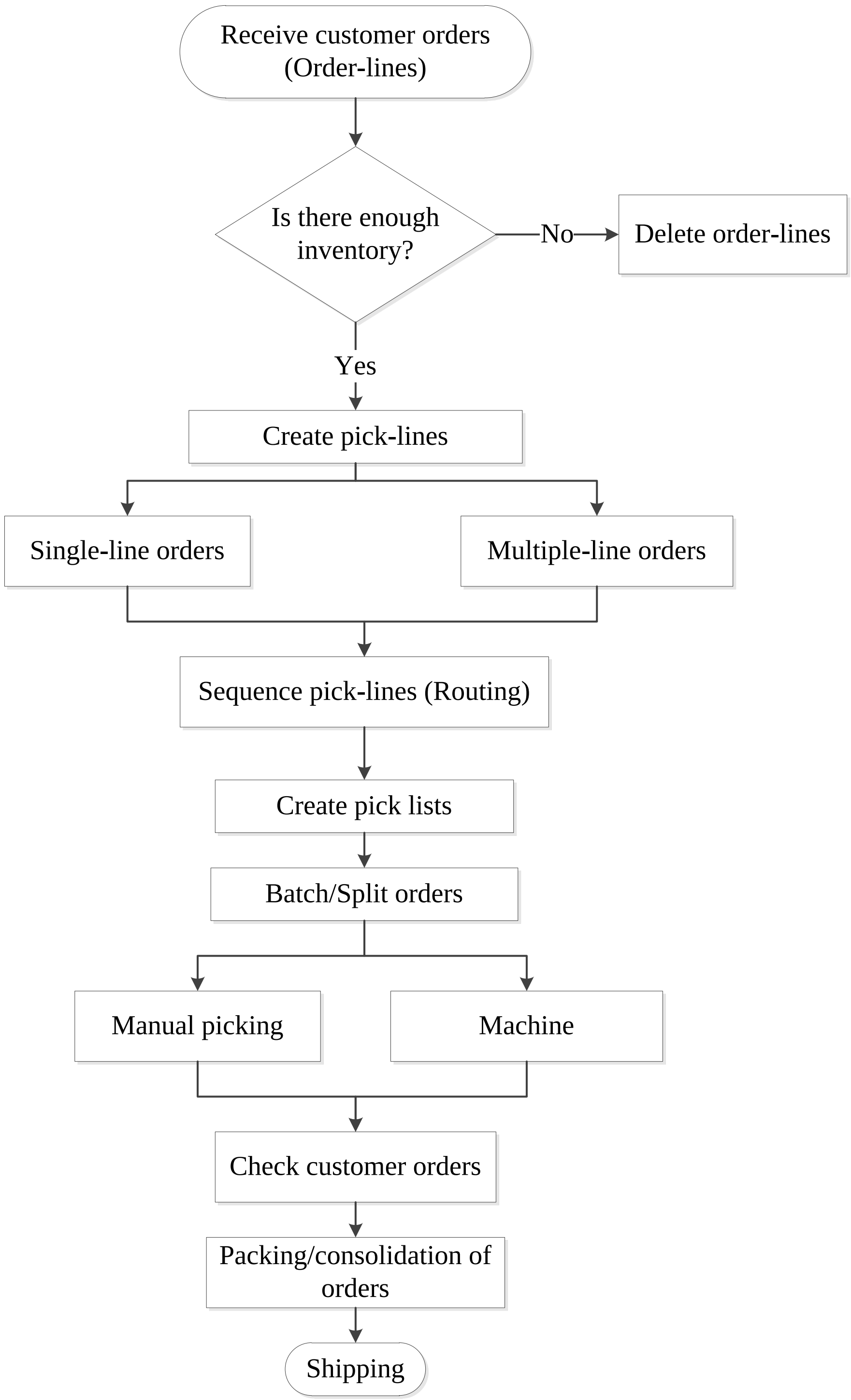


Figure 13 General order picking process flowchart (Bartholdi & Hackman, 2011)

APPENDIX 2 – LAYOUT OF THE WAREHOUSE

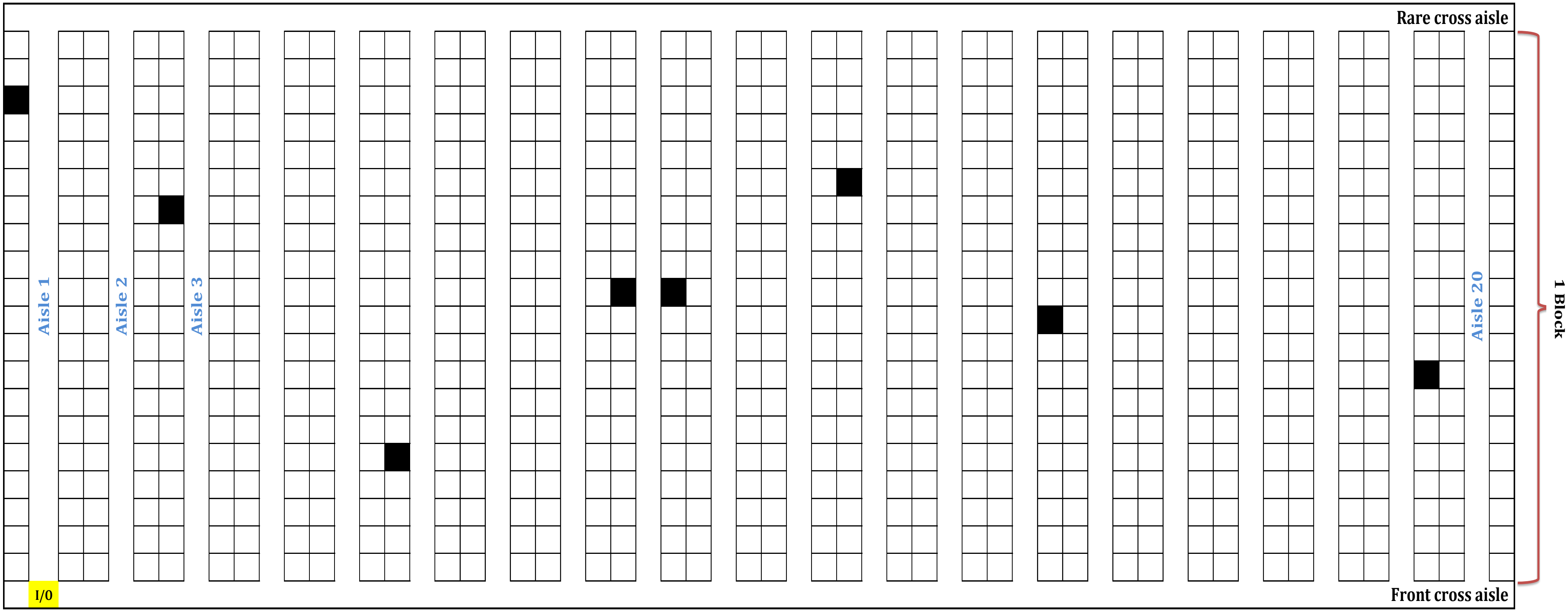


Figure 14 Layout of the warehouse

Input parameters:

- Number of aisles (2-sided) – 20
- Storage locations per aisle – 40
- Total pick locations – 800
- I/O-point location – lower left corner
- End-aisles – 2 (Rare and Front)
- Level of racks – 1
- Number of blocks - 1