A Study on Storage allocation problem based on clustering algorithms for the improvement of warehouse efficiency

Mohammad Reza Nafar

A Thesis
in
John Molson School of Business

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Supply Chain
Management at Concordia University
Montreal, Quebec, Canada

August 2022

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Mohammad Reza Nafar

Entitled: A Study on Storage allocation problem based on clustering algorithms for the improvement of warehouse efficiency

and submitted in partial fulfillment of the requirements for the degree of

Master of Supply Chain Management

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Danielle Morin Chair

Dr. Xiaodoan Pan Examiner

Dr. Satyaveer S.Chauhan Examiner

Dr. Satyaveer S.Chauhan Thesis Supervisor(s)

Dr. Salim Lahmiri Thesis Supervisor(s)

Approved by
Dr. Satyaveer S. Chauhan Chair of Department or Graduate Program Director

Dr. Anne-Marie Croteau Dean October 2022

Abstract

A Study on Storage allocation problem based on clustering algorithms for the improvement of warehouse efficiency

Mohammad Reza Nafar

The operation of warehouses has long been a focus of industry research. Faced with rapidly growing business needs, improving storage efficiency, and reducing customer response times have become crucial issues for improving the operational efficiency of a warehouse. Given a fixed area of space, optimizing the storage strategy can reduce the cost of goods handling, improve the efficiency of storage and delivery, accelerate the overall operational efficiency of the warehouse, and reduce logistical costs. In this paper we study the improvement of a real-life company's storage location strategy using cluster and association analysis. Two different clustering techniques namely pairwise comparison clustering and K-means clustering are used, and their performances are compared with the current random storage policy used by the company. Both clustering algorithms consider item association and classify items into groups based on how frequently they appear with each other in customer's orders. The next stage applies assignment techniques to locate the clustered group in each aisle so as to minimize the total number of aisle visits and ultimately picking distance. By emphasizing the item association, our model is suitable for orders with multiple items in the modern retailing sector. It also more effectively shortens the picking distance compared with random assignment storage method. In our case, Warehouse studied herein, both models prove more effective as it reduces over 35% and 25 % of the picking distances versus the current set-up. However, when compared with each other the K-means clustering method outperforms the pairwise comparison.

Acknowledgments

I would like to express my gratitude and appreciation to all those who gave me the possibility to complete this thesis. Special thanks is due to my supervisors Dr Satyaveer Chauhan and Dr Salim Lahmiri whose help, suggestions and encouragements helped me in writing this thesis. It would have never been possible for me to achieve this degree without their incredible support and encouragement.

I would also like to thank my family for their constant guidance support and motivations. Lasty I would like to acknowledge me classmates and friends who not only made this Master's degree and enjoyable one but also helped me with their knowledge and advice throughout this journey.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

A warehouse is a facility that is designed to store material in bulk for future distribution or sale. Warehouses form an important connection among suppliers, distributors, and customers in the supply chain. For supply chain operation to be efficient, warehousing management is necessary because it acts as an intermediary connecting upstream suppliers with downstream customers along the supply chain. Currently, retailing systems require fast inventory turnover and heavily rely on a quick and accurate replenishment system. With electronic ordering systems, retailers and customers are able to order more frequently and in smaller quantities. Therefore, From the operational standpoint, it is necessary that warehouses increase their efficiency to meet the rapid and timely demand from retailers/customers (Yi-Fei Chaung et al., 2012). According to the literature, (Kudelska & Pawłowski, 2019) stated that warehouse operations are labor intensive and, in most cases, more than 50% of warehouse operating costs belong to order-picking. Which pinpoints the great potential of efficient order picking for warehouse improvements. Figure.1 illustrates each main warehouse activity as a percentage of cost, emphasizing the importance of order picking efficiency. Furthermore, Tompkins, White, Bozer, Frazelle, and Tanchoco (2003) also indicated that among all warehouse operations, order picking is the most important one. They divided the order picking process into five major activities: travel, search, pick, set-up, and others, and the percentages of time for the five activities are 50%, 20%, 15%, 10%, and 5%, respectively.

Figure.2 demonstrates the subtasks of order picking in their study. Therefore, it is evident, that travel is the most time-consuming activity and is the most common subject academically and practically for improving warehouse efficiency.

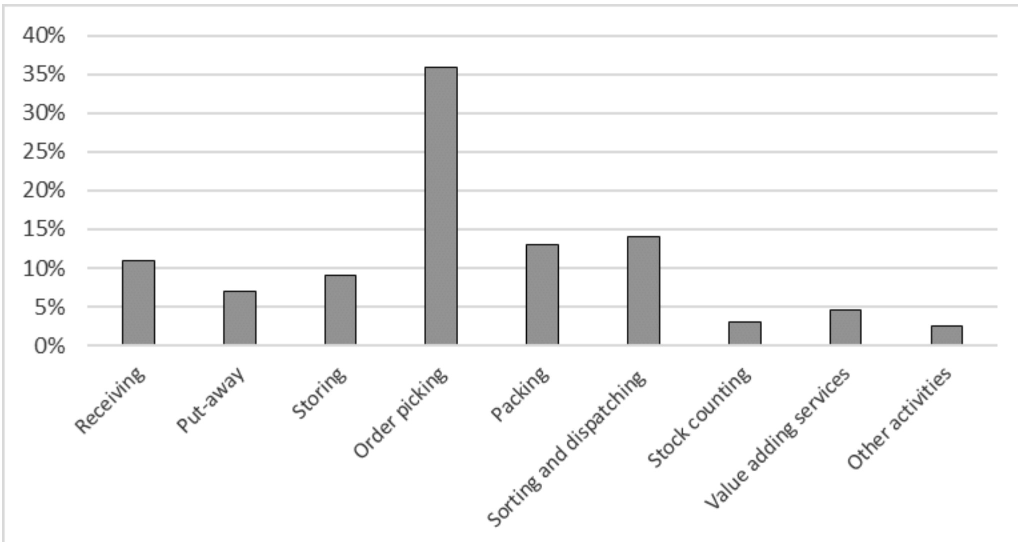

Figure 1. Warehouse activities as a percentage of total costs (Škerlič et al., 2017)

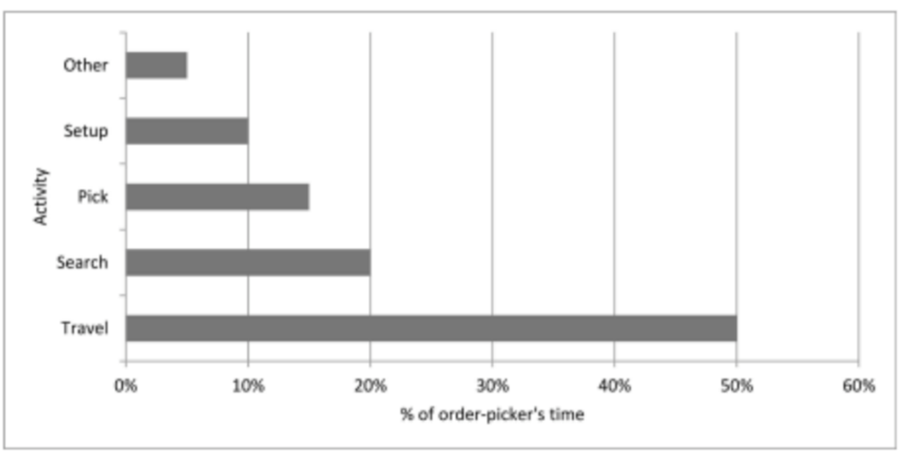

Figure 2. Order picker's time distribution (Tompkins et al., 2003).

The travel time of order picking in a warehouse can be reduced in a number of different ways, such as altering the layout of racks, assigning better storage places, selecting shorter ordering picking routes, and accumulating orders into batches to decrease picking frequency (Chuang et al., 2012). For most pick-by-order warehouses, layout modification may not be realistic as it is expensive and time consuming. Optimizing the sequence in which an order is picked, or order batching can significantly reduce the time needed to fulfil an order. However, aside from order picking strategies, another critical decision is on storage location assignment prior to the order picking decision (Chan, H. L., Pang, A & Li, K. W., 2011). The decision on the storage location of items can be expressed as a storage location assignment problem (SLAP) which involves the decision on assigning the incoming item, often defined as a Stock Keeping Unit (SKU), to a particular location in a warehouse, with the objective of minimizing the distance traveled for order picking or maximizing the space utilization of a warehouse. Thus, assigning appropriate storage locations is a feasible approach for reducing travel time and distance (Chuang et al., 2012).

1.2 Outline

The remainder of this paper is organized as follows. Chapter 2, reviews relevant literature in the area of warehouse slotting. Chapter 3 will explain the formulation of both models for the SLAP followed by a demonstrative example. The experimental procedure and results of our study is displayed in chapter 4. Chapter 5 presents the case study on the company's warehouse and showcase the improvements. Chapter 6 will summarize the implications of our research and how it will benefit the company. Future potential extensions of this study are also argued in this last chapter.

Chapter 2:

Literature Review

This Chapter provides an overview of the most relevant studies in order to identify and highlight the recent methods in the research areas relevant to the problem studied in this thesis. The chapter is divided into the following three sections.

(1) Section 2.1 – Storage Location Assignment Problem

(2) Section 2.2 – Typical Storage Policies

(3) Section 2.3 – Correlated Storage Policy

2.1 Storage Location assignment problem

SLAP involves assigning incoming SKU's into the best storage locations based on certain criteria while achieving predefined objectives. According to literature, the most common objectives are as follows. The first is to improve the overall operating efficiency (Heragu, Du, Mantel, & Schuur, 2005; Hsu, Chen, & Chen, 2005; De Koster et al., 2007; Jane & Laih, 2005), next is to minimize storage space costs (Muppani & Adil, 2008) and finally to minimize picking distance (Hwang & Lee, 1988; Rosenwein, 1994; Liu, 2004). In the sections to come we discuss how these objectives are achieved by different storage policies, the benefits, and drawbacks.

2.2 Typical Storage Policies

Generally, in a warehouse, product storage approach for incoming items in SLAP can be categorized as randomized, dedicated and class-based storage (Gu et al., 2007). Randomized storage approach assigns the items arbitrarily to storage locations without taking into account the products popularity and interdependency. The incoming items are generally stored at the first available location closest to the I/O entry point. In their paper Chloe, K., and Sharp, G.P., (1991) showed that the randomized storage method results in better space utilization of the warehouse at the cost of increased travel distance for order picking operations.

In dedicated storage approach, each storage location is fixed to a specific product based on demand and therefore is sometimes referred to as volume-based storage. Item locations are reserved, even when some products are out of stock, other products are not allowed to be allocated in these spaces. According to (Linn and Wysk 1987; Malmborg 1996), the dedicated storage approach gives better result in terms of order picking travel distance when compared with randomized storage approach because order pickers become familiar with product locations, so they take shorter time to locate the products. However, its drawback is low space utilization (De Koster et al., 2007).

The class-based storage is a combination of the aforementioned storage methods and classifies products into several categories. Each classification has a fixed zone, in which products in the same zone are assigned to storage locations (De Koster et al., 2007). The assigning criteria are closely related to the objectives of SLAP. Some of the most common criteria used in the literature are turnover, popularity, volume pick density and cube per order index (COI). For picking distance minimization, the most logical way to assign a storage location is based on item turnover (Chuang et al., 2012). By assigning the highest turnover item to the location closest to the outlet of the depot can reduce picking distance. To improve operating efficiency, the most common measure used in the literature is the COI. Which is the ratio of an item's volume to the number of trips required to fulfil the item's order demand per period. It considers the product's popularity and its storage space requirement simultaneously. The typical storage strategies are summarized in figure 3. Some studies on the aforementioned storage policies are as follows.

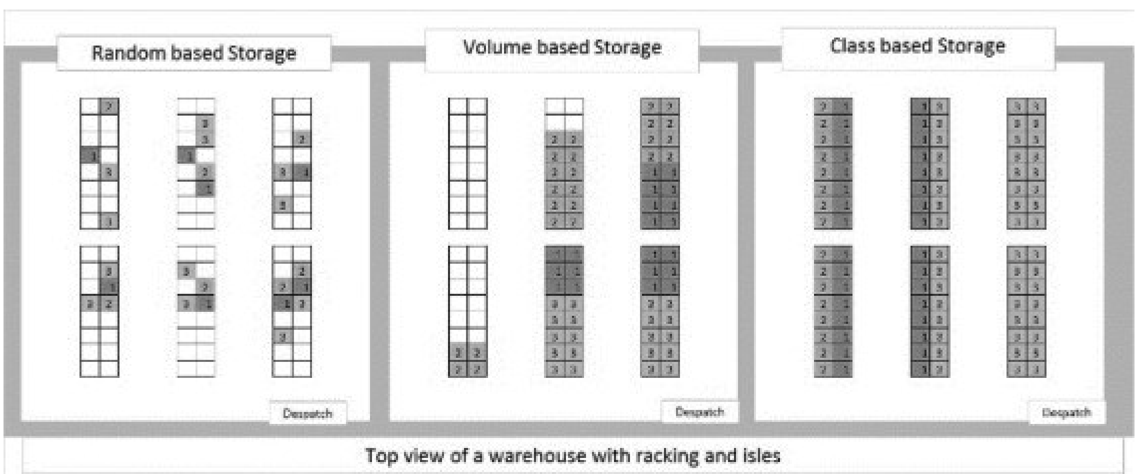

Figure 3. Typical storage Policies (Mickleson, G., Thai, V. V., & Halim, Z. 2019)

The study of Glock and Grosse (2012) offered a U-shaped shelving system in order to compare different storage policies and order picking strategies. Rao & Adil's (2013) study focused on a low-level picker to part warehouse, and they demonstrated a three-class storage policy for a S-shaped traversal routing system. The study of Quintanilla et al. (2015) involved a storage location assignment problem with dynamic storage and practical restrictions in a unit-load warehouse. They presented several heuristic algorithms to improve the space availability in a warehouse by determining the strategies for relocating the stored items. COI is one of the earliest metrics developed and it is widely used technique for dedicated storage strategy (Khullar, Chirag 2021). In his study Petersen (1999), showed that volume-based storage provides better performance compared to random storage. The study of Petersen, et al. (2005) show that COI slotting measure provides good results and is well known to minimize order picking travel time for single command order picking, however for multi command order picking this is not true (Schuur, 2015). Frazelle and Sharp (1989) pointed out that when a customer's needs involve multiple items on the same order, especially when the order of one item is associated with the order of another, a correlated storage assignment strategy may improve the efficiency of order selection. As COI is based on picking just one SKU in each trip it becomes unrealistic for most warehouses that pick multiple SKU picks in a single order. Therefore, COI is not a suitable metric for the scope of our study as we deal with orders of multiple items. Hence, besides the turnover and frequency of items, how SKUs are to be arranged based on their combination in an order also becomes essential. Items in customer orders could be correlated therefore, it may not be optimal to assign the premium locations to products with higher turnover. The aforementioned storage policies studied, do not consider interdependence among the products in a picking order. Yener & Yazgan (2019) mentioned that in the retailing industry, customers frequently purchase similar products together. This is directly linked to the warehousing operation, and they stated that a similar scenario can be observed in the warehouse. When several items are frequently requested in the same picking order these products are called correlated products. It is advantageous to store these items in adjacent locations to reduce travel time during the order picking activity, given that the product dependency can be predicted.

2.3 Correlated Storage Policy

Correlated storage policy is founded on the estimation of an appropriate index of correlation among the items in a product mix. More specifically, cluster analysis classifies groups of items that customers frequently order together, products with high index value of correlation are then placed close to each other as it makes order picking more efficient by reducing the travel distance. The strategy for the storage location assignment with the consideration of correlated products has also been studied extensively.

Kress et al. (2017) divided items into different groups using mathematical models to minimize the total number of products that have to be reached in order to pick customer orders. Research like, (Frazelle, 1989; Sadiq et al., 1996) studied family-group assignment policies, where certain products are grouped and allocated to a subsection of the warehouse based on shared characteristics. A strategy commonly used in the retailing industry with the objective to minimize shelf replenishment time. Amirhossein and sharp (1996) introduced several measures to find the association between products. They discovered that an optimal strategy is possible by taking in account all combinations pairs, but they mentioned that it only works for warehouses with limited number of products. Frazelle (1989) applied a two-stage heuristic approach for SLAP that minimizes the order picking travel time by looking at the correlation between items. In the first stage products are clustered, starting with the most popular products and adding the highest correlated products until the capacity of the cluster is reached. The second stage allocates the clusters with the highest total popularity to the closest available locations. A similar approach was taken by Amirhossein and Sharp (1996), in which a pair of clusters with the greatest correlation repeatedly merge until they reach the maximum allowed size.

Zhang et al. (2019) introduce the concept of the demand correlation pattern to assign the items ordered together to storage locations and to determine order picking routes. They use an S-shape routing and solve their mathematical model using simulated annealing. Accorsi et al. (2012) make use of the nearest and farthest neighborhood algorithm to cluster correlated products. They then, assign clusters to locations based on COI and other ranking indices calculated for each cluster. Sharp et al. (1998) propose a heuristic that improves existing hierarchical clustering algorithms. It is used to assign an assortment up to 700 products. Garfinkel (2005) considered correlated storage strategy for zone picking. The goal was to minimize the number of zones visited for all orders by assigning products to specific zones based on their correlation. Jane and Laih (2005) measure the similarity degree of two products by their coappearance in an order set and then store similar items over different zones. They formulate an integer programming model to maximize the utilization of a synchronized zone and solve it using a heuristic. Xiao and Zheng (2012) use the bill of materials to calculate item correlation and then use a mathematical model to minimize zone visits. They use heuristics to solve the model. Jewkes et al. (2004) use a stochastic model for product assignment and picker allocation to minimize the cycle time of picking random orders from storage bins that store multiple products. Hansen and Jaumard (1997) formulate a mixed-integer program to form clusters, with the objective of maximizing the total affinity in the clusters. Bindi et al.

(2009) use data mining techniques to define a similarity measure that is used in a clustering algorithm and assignment rules. Their paper includes a case study, which shows that similarity-based strategy outperforms both class-based and random strategies. Wang et al. (2020) present a storage location assignment model to minimizes the total travel distance for fulfilling customer orders. They use a heuristic that switches item pairs to improve the travel distance, based on the information in the order data. Although the literature on cluster-based storage location assignment policies have been extensively studied, comparing different clustering strategies remains rare. Therefore, this research differs from previous ones and extends the literature on correlated storage policy as it compares two different cluster-based strategies in real life warehouse that is currently implementing a random storage policy. We adopt ideas from previous studies and adjust them to fit the scope of our research and to extend the literature.

Chapter 3

Problem Statement

3.1 Notations

P(i) Number of orders for item *i*

P(j) Number of orders for item *j*

K Number of clusters

X_{ik} If item *i* is assigned to cluster *k* the value is 1; 0 otherwise (binary value)

X_{jk} If item *j* is assigned to cluster *k* the value is 1; 0 otherwise (binary value)

N total number of items

3.2 Pairwise Clustering Methodology

The pairwise clustering model can be described in the following two-stage procedure.

Stage 1: clustering items into groups

Step 1: Calculating the correlation between items: The support between items is calculated by the appearance of the item in orders. We implement an order item association rule that defines the item-index between items *i* and *j* as follows:

$$S_{ij} = \frac{P(i \cap j)}{[P(i)+P(j)]} \tag{1}$$

In this formula the numerator, $P(i \cap j)$ represents the number of orders containing both items *i* and *j*. The denominator represents the number of orders that contain item *i* and/or the number of orders that contain item *j*. In other words, the value of S_{ij} ranges from ½ to 0. S_{ij} = ½ means that items *i* and *j* are always order together, whereas S_{ij} = 0 means that item *i* and *j* are never ordered together.

Step 2: Clustering items: In this step the items are assigned into clusters. The purpose of this step is to achieve the highest between-item-support (maximum item index value) so that items with higher associations can placed in adjacent locations. This objective is achieved, by solving the following mathematical programming problem.

$$Maximize \sum_{K=1}^{K}\sum_{j>1}^{N}\sum_{i=1}^{N-1} S_{ij}X_{ik}X_{jk} \tag{2}$$

$$Subject\ to\colon \sum_{K=1}^{K} X_{ik} = 1\,; \qquad i\ \in \{1, 2, 3, \ldots, N\}$$

$$X_{ik} \in \{0{,}1\}; \quad X_{jk} \in \{0{,}1\}$$

$$i\ \in \{1, 2, 3, \ldots, N\}; \quad j > i \tag{3}$$

The pair-wise selection of the $X_{ik}X_{jk}$ term in the objective function (Eq. 2) guarantees that the support-index accumulates only when items *i* and *j* are in the same cluster *K*. The constraints in (Eq. 3) ensures that each item can only be clustered in one group. We adopt the linearization method proposed by Yi-Fei Chaung, Hsu-tang Lee and, Yi-Chaun Li (2012) to transfer the 0-1 quadratic programing into MIP. The Assigning of different clusters into storage locations awaits the assignment process in the next stage.

3.3 K-Means Clustering Methodology

Clustering items into groups:

The K-Means algorithm uses iterative improvement to produce a final result. The inputs needed for the algorithm are the number of clusters K and the data set. In our case we set the number of clusters K to be equal to the number of aisles in the warehouse. The algorithm starts with initial estimates for the K centroids, which can either be randomly generated or randomly selected from the data set. The algorithm then iterates over two steps (Rahman, S. M. M. 2019):

Step 1: Data Assignment: Each centroid is representation of one of the clusters. In this step, each data point (item) is assigned to its nearest centroid, based on the squared Euclidean distance. More specifically, if c_i is the collection of centroids in set C_i then each data point p is assigned to a cluster based on:

$$\arg\max_{c \in C} dist\ (c_{i,}p)^2$$

$dist()$ is the standard (L_2) Euclidean distance

Step 2: Updating the centroid: Let the set of data point assignments for each i^{th} cluster centroid be S_i. The centroids are recomputed by taking the mean of all data points assigned to the centroid's cluster.

$$c_i = \frac{1}{|S_i|} \sum_{p_i \in S_i} p_i$$

The aforementioned two steps are continuously iterated by the algorithm until no data points change clusters and the sum of the Euclidian distance is minimized.

Since each cluster is assigned to an aisle, then the maximum items allowed in each cluster must be the capacity of the items a given isle can hold. The K-means algorithm does not account for this and therefore a capacity issue arises. In order to implement this in our model, when the maximum capacity of a cluster is met, we add the new item in the Cluster but immediately reevaluate the cluster (find new centroid) and remove one item which is farthest from the centroid. The capacity of the cluster is given by the following inequality.

$$\sum_{i=1}^{n} x_{ik} < C \qquad k \in (1, 2, 3, \ldots.)$$

Where C is the capacity of the cluster.

3.3.1 K-means Pseudo code

The K-Means clustering algorithm used is summarized in the form of pseudo code below:

χ: Set of points p $\{p_1, p_2, \dots, p_k\}$
See table 1 for defining $p_1, p_2, ..$

Input: Data points *D,* number of clusters k

Step 1: Randomly choose an initial k center $\varsigma = \{c_1, c_2, \dots, c_k\}$

Step 2: For each $i \in \{1, \dots, k\}$, set the cluster C_i to be the set of points in χ that are closer to c_i than they are to c_j for all $j \neq i$

Step 3: For each $i \in \{1, \dots, k\}$, set c_i to be the center of the mass of all points in C_i : $c_i = \frac{1}{|S_i|} \sum_{p_i \in S_i} p_i$

Step 4: Repeat Steps 2 and 3 until ς no longer changes.

Output: Data points with cluster members.

3.3.2 Determining the value of k

The algorithm described above finds the clusters and data set labels for a particular pre-chosen K. The parameter K in K-Means clustering denotes the number of clusters that the data will be divided into. It is a parameter that needs to be set before the algorithm is started. In general, there is no method to determine the exact value of K but in the case of our problem, since we are trying to assign items into storage location, we set K to the number of aisles present in the warehouse. In other words, each cluster will be assigned to an aisle. With that being said the performance of the algorithm does not depend on the value of K.

3.3.3 K-means Clustering Intuition

K-means is an algorithm that allows you to cluster your data and it is a convenient tool for discovering categories of groups in your data set. In this section we describe the intuition behind the algorithm and also provide a small example to help explain how the K-Means algorithm was implemented in this thesis.

The data set that is used is displayed on a scatterplot based on the variables associated with each data point. In our case, each data point represents and items in the warehouse. Therefore, they are plotted on a scatterplot based on how they appear with each other in customer orders. In other words, each item is given a coordinate on a scatterplot depending on their combination in an order. In fact, each data point represent an SKU. We define *n* dimensional Cartesian space (n= number of SKU's). In order to define the location of each SKU in this space, we simply compute the affinity of the SKU with other SKUs. Affinity in our case simply means the frequency i.e. if an item is ordered 20 times with item X, then the X-coordinate of this item will be 20.Since we are considering n-dimensional space, we compute this frequency for all the items. Table 1 illustrates an example of the combination of n items in customer orders.

Table 1. Defining a data point

Items\coordinates	1	2	3	……………………………………..					n
1		2	10						
2	2		10						
3	10	2							
.									
.									
.									
.									
.									
n									

The table can be understood in the following way. The top row represent the coordinates (n). Each cell within the outermost column and row represents the quantity that the corresponding items appear together. For example, item 1 is purchased together with item 2, two times and purchased with items 3, ten times. Based on these variables all the items are plotted on a scatter plot, with the closely associated item plotted closer to each other. The data set is then cluster into groups in following way:

Step 1: Choose the number of K clusters (in our case this is number of aisles in the warehouse)

Step 2: K centroids are selected

Step 3: Each data point is assigned to the closest centroid based on Euclidean distances (that forms K clusters)

Step 4: The new centroid of each cluster is computed

Step 5: each data point is reassigned to the new closest centroid. If reassignment took place, step 4 is repeated otherwise the algorithm ends.

3.4 Demonstrative example

We now provide a demonstrative example to examine the viability of our model. The orders in the example are constructed in a way to test the ability of the model in grouping items that appear together frequently in the same order in one cluster. Ten orders and their demands are randomly generated. Table 2 Illustrates the simulation of these orders as follows. The ten orders are labelled O1 to O10 and consist of twelve items labelled I1 to I12 in total. A cell containing the number 1 indicates that the item corresponding to that cell was purchased. An empty cell indicates that the item corresponding to that cell was not purchased. For example, order O1 contains three items I1, I2 and I3. The purpose of this example is to validate the models that we are proposing in this study. The example is purposely constructed in manner so that the solution is obvious and there is no need of a model to cluster the items. For example, visually it can be noted that I1, I2, and 3 should be placed in one cluster, I4, I5, and I6 in another cluster and I8, I9, I10 in another cluster. We run both models and compare it to the aforementioned cluster predictions, If the results from our model provide us with the same solution, then we can validate our model. Table 3 Lists the grouping results from both the pairwise and the K-means model. The grouping of the models differs slightly but they seem to have captured the result we were expecting. Therefore, our model is proved to be suitable for the context of the problem.

Table 2. Simulation Orders

Orders	*Items*											
	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12
O1	1	1	1									
O2	1	1	1									
O3				1	1	1						
O4				1	1	1						
O5				1		1	1	1	1	1		
O6	1		1					1	1	1		
O7											1	1
O8											1	1
O9							1				1	1
O10	1	1			1	1				1		1

Table 3. Result of Grouping

Result of Clustering	
Pairwise Clustering	{1, 2, 3}{4, 5, 6,}{7, 10}{8, 9, 11, 12}
K-Means Clustering	{1, 2, 3}{4, 5, 6}{7, 11, 12}{8, 9, 10}

Chapter 4

Experimental Procedure

4.1 Experimental setup

To investigate the performance of the heuristics, experiments of different scenarios were conducted. Therefore, in this chapter we describe the steps taken to complete our experiments, the simulation technique used to minimize picking distance and we provide an example to summarize the whole procedure.

For the purpose of this thesis, we run experiments for three different warehouse scenarios. A small warehouse, medium sized warehouse, and a large warehouse. We consider a warehouse to be small if it contains a range of 50 to 100 items, medium if it contains a range of 150 to 200 items and large if it contains 250 items or more.

For each warehouse scenario we consider three different cases of maximum items. For example, for a small warehouse we consider 50, 80 and 100 items. For simplicity, we assume that in all the cases, each aisle in the warehouse can hold 10 items. Therefore, the value of K (Which represents the number of aisles/clusters) can be found by dividing the items in the warehouse by the maximum number of items a given aisle can hold. For example, the value of K for a small warehouse with 50 items will be 5, a warehouse with 80 items will be 8 and so on.

For each case, we use Microsoft excel to randomly generate customer orders. These customer orders are represented similar to the orders illustrated in table 2 of the demonstrative example section of this thesis. Where each column represents an item, and each row represent an order. Once the orders are generated, we use our clustering algorithms to cluster the items into K clusters depending on the number of items and number of spaces available in a given aisle.

4.2 Order picking efficiency Simulation

The purpose of the simulation is to assess the performance of item associated cluster assignment models. A decrease in picking distance indicates an improvement in picking efficiency, meaning that the proposed models are superior to the current strategy. Since the number of aisles visited is directly proportional to the picking distance, we use the number of aisles visited to satisfy the customer orders in a given period as our performance metric. We calculate the number of aisles visited in the following way.

Using Microsoft excel we develop a macro to count the number of times a single aisle is visited for each customer order. The customer orders are represented in a table where each column represent an item and each row represent an order (Similar to table 2). Since each cluster represents an aisle in the warehouse, our macro searches and counts any of the items in a given cluster that is also present in a certain order. This is done for all clusters and orders. We assume that if an aisle is visited the whole length of the aisle is traveled. We then sum the number of items picked up from each aisle. Based on this we assign storage location of the aisles, with the aisle containing the highest number of ordered items placed closest to the I/O entry point. This procedure will be clearer in context of an example which is provided in the next section.

4.3 Demonstrative example

In the section, we demonstrate an example to summarize the experimental procedure of this thesis. We use the same data provided in table 2 of the demonstrative example section of chapter 3. Given the explanation of the simulation technique in section 4.2, the results of the macro are as follows:

Table 4. Summation of items picked from each aisle

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Order 1	3	0	0	0
Order 2	3	0	0	0
Order 3	0	3	0	0
Order4	0	3	0	0
Order 5	0	2	1	3
Order 6	2	0	0	3
Order 7	0	0	2	0
Order 8	0	0	2	0
Order 9	0	0	3	0
Order 10	2	2	1	1

Table 4 contains four columns which represent the number of clusters/aisles and 10 rows which represent the number of customer orders in our example. Each cell represents the number of items that are selected from the corresponding order and cluster. For example, the cell corresponding to order 1 and cluster 1 is 3 which indicates that in the first order, 3 items that are in cluster/aisle 1 is ordered. Any none zero integer in table 4 represent one aisle visit. With this explanation we can conclude that in the first order only the first cluster/aisle is visited once. To find the total number of aisle visits to satisfy the customer orders we take the sum of all the nonzero digits in table 4. In this case the sum is 16. The goal is to minimize this number. Figure 4 to 13 shows a warehouse countaining 4 aisles and summarizes the customer orders in our demonstartive example. The shaded items imdicate that, that item was present in an order and that specific aisle is visisited. In order to assign a cluster to an aisle we sum each of the columns of table 4. This way we are able to identify the most popular aisle. We assign the cluster containg the highest number to the aisle closest to the I/O entry point.

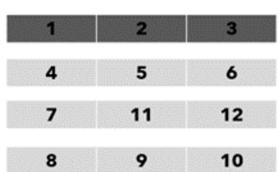

Figure 4. Order 1

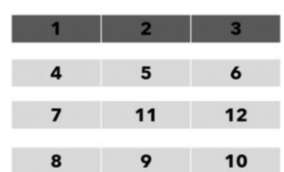

Figure 5. Order 2

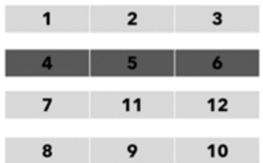

Figure 6. Order 3

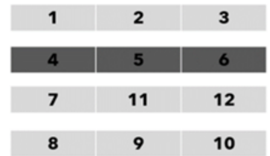

Figure 7. Order 4

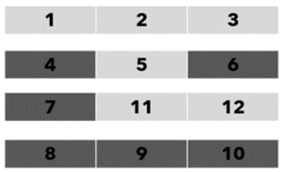

Figure 8. Order 5

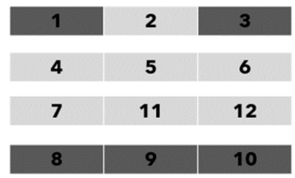

Figure 9. Order 6

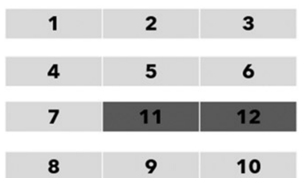

Figure 10. Order 7

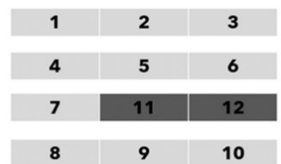

Figure 11. Order 8

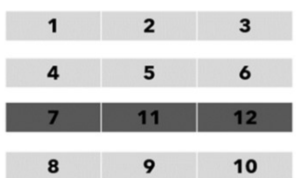

Figure 12. Order 9

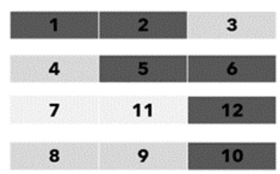

Figure 13. Order 10

4.4 Experimental Results and discussion

We now look at the results according to the setup described in the previous section. Table 5 to 7 illustrate the results of the experiment on small, medium, and large size warehouse. Each table consists of the following column headings: Number of items, Number of clusters, Visits and Run time of the heuristic in seconds. To get the number of visits and run time of each scenario the experiment is run ten times with different orders generated randomly. The visits and run time for each experiment is noted and the average is found. This is done to ensure that the results are reliable and unattained by bias.

Table 5. Small Warehouse

		Visits		Time	
Number of items	Number of clusters	K means	LP Clusters	K means	LP Clusters
50	5	178	181	4.2	15
80	8	315	344	7	22.4
100	10	473	481	9.3	40

Table 6. Medium Warehouse

		Visits		Time	
Number of items	Number of clusters	K means	LP Clusters	K means	LP Clusters
150	15	556	583	12.5	475.3
180	18	575	610	18.8	965.9
200	20	1384	1465	19.5	2008.9

Table 7. Large Warehouse

		Visits		Time	
Number of items	Number of clusters	K means	LP Clusters	K means	LP Clusters
250	25	1208	1228	35	2756

4.4.1 Aisle Visits

Figure 14 to 16 shows the number of visits for both heuristics plotted on the same graph for each warehouse scenario with different number of items. This is done to visually compare both heuristics. Figure 14 shows a small warehouse, Figure 15 shows a medium size warehouse and Figure 16 shows a large warehouse.

For each figure, we see that the general trend is an increase in number of visits as number of items increase. This is because as the number of items increase there is bigger range of items to choose and naturally will require more visits. Furthermore, there is greater chance of different combination of items in appearing in different orders. This becomes more evident as the warehouse size increases.

The next point to note is that the K-means Heuristic outperforms the Pair wise heuristic in every experiment except one. The pairwise optimization problem looks to maximize an objective function (equation 2 in section 3.2) that contains the item index S_{ij} (formula 1 in section 3.2). In this formula the numerator, $P(i \cap j)$ represents the number of orders containing both items i and

j. The denominator represents the number of orders that contain item i and the number of orders that contain item j. S_{ij} = ½ means that items i and j are always order together, whereas $S_{ij} = 0$ means that item i and j are never ordered together. This could be misleading because item i and j could appear together only once in a large number of orders and still be assigned a value of ½. In other words, the algorithm will assume the items have high association, but in reality, it may not be true as they only appear together once. It is also evident that as number of items and the size of the warehouse increases the difference between the performance of the two heuristics gets larger.

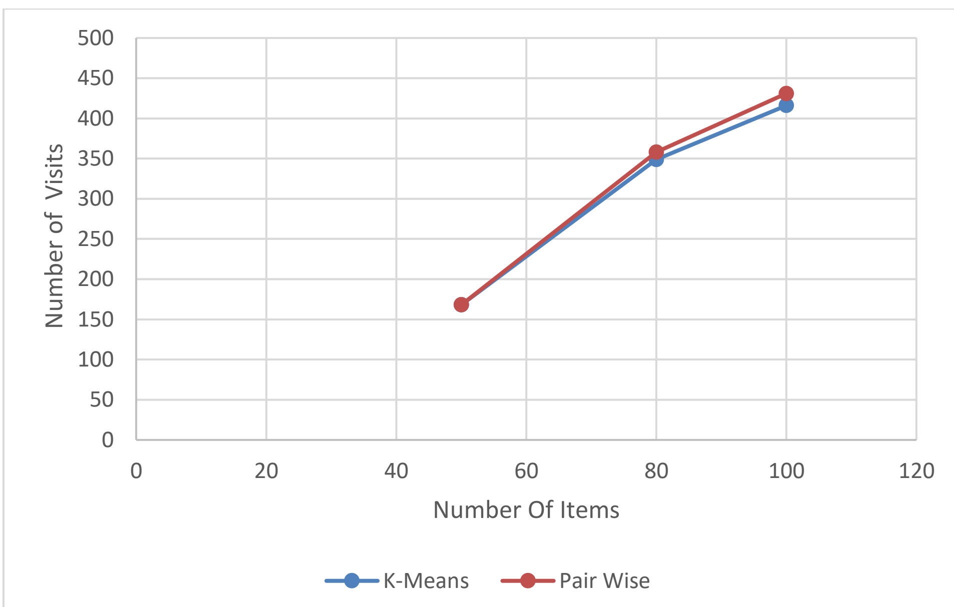

Figure 14. Number of visits for small warehouse

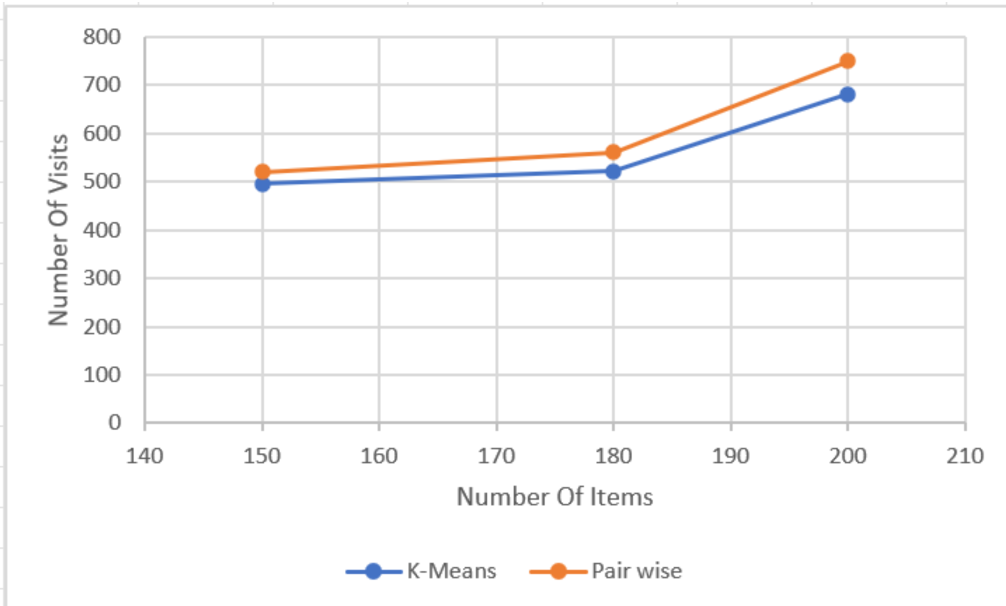

Figure 15. Number Of visits for medium warehouse

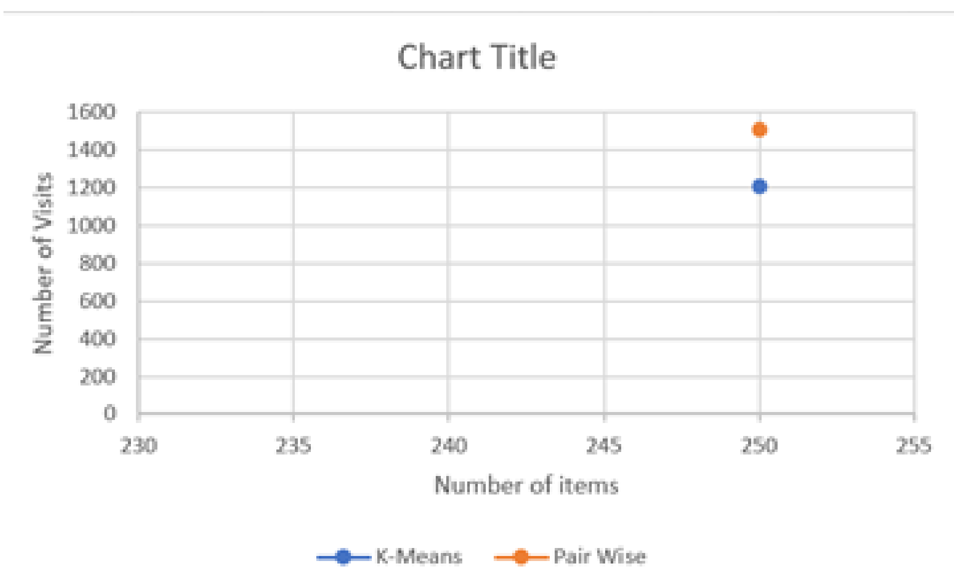

Figure 16. Number of visits for small warehouse

4.4.2 Run Time

Aside from the total number visits needed for both heuristics another important metric to considered is the time taken to run the models. Therefore, the next parameter we compute is the runtime for both heuristics. Figure 17 to 19 shows the run time plots, plotted on the same graph. Figure 17 shows a small warehouse, Figure 18 shows a medium size warehouse and Figure 19 shows a large warehouse. There are several notes to point out from the results.

As the number of items and the size of the warehouse increases more time is taken for the algorithms to run. This is logical because, the algorithm is dealing with a greater number of items and therefore requires more time to compute the associations between items. It can also be noted that the K-Means heuristic outperforms the pairwise heuristic, and we can easily conclude that the K-Means is more time efficient. The pairwise algorithm clusters items by calculating the item index between every possible combination of pairs of items. In other words, it computes the association of every single item with each other, whereas K-means algorithm considers the whole data set and classifies the data points into K clusters according to their characteristics similarity by using distance measures such as Euclidean distance and therefore is much quicker than the pairwise algorithm.

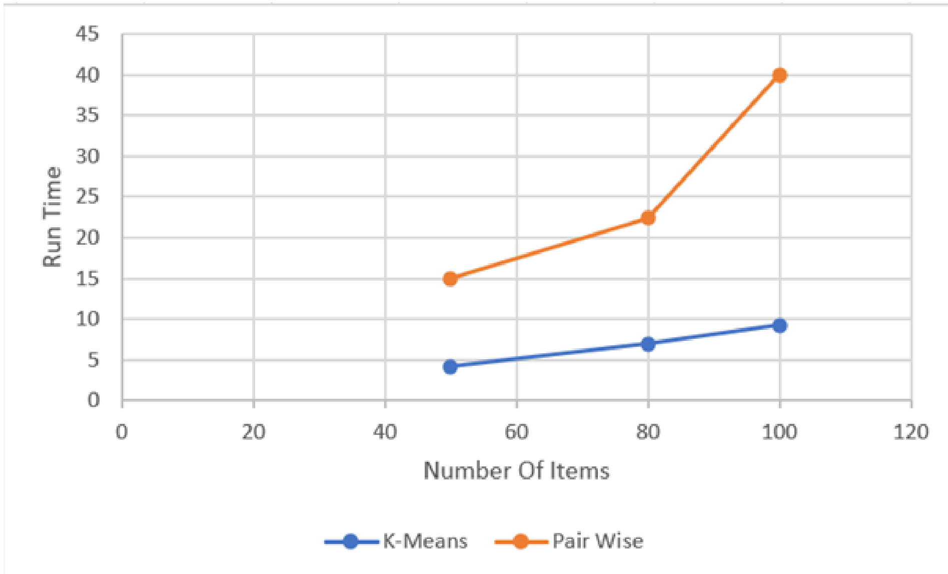

Figure 17. Run Time for small warehouse

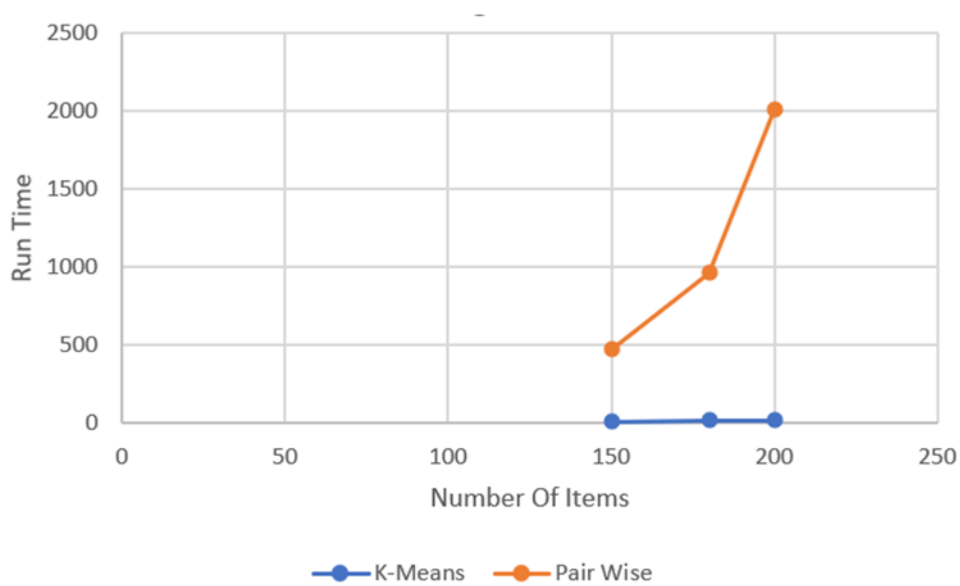

Figure 18. Run time for medium warehouse

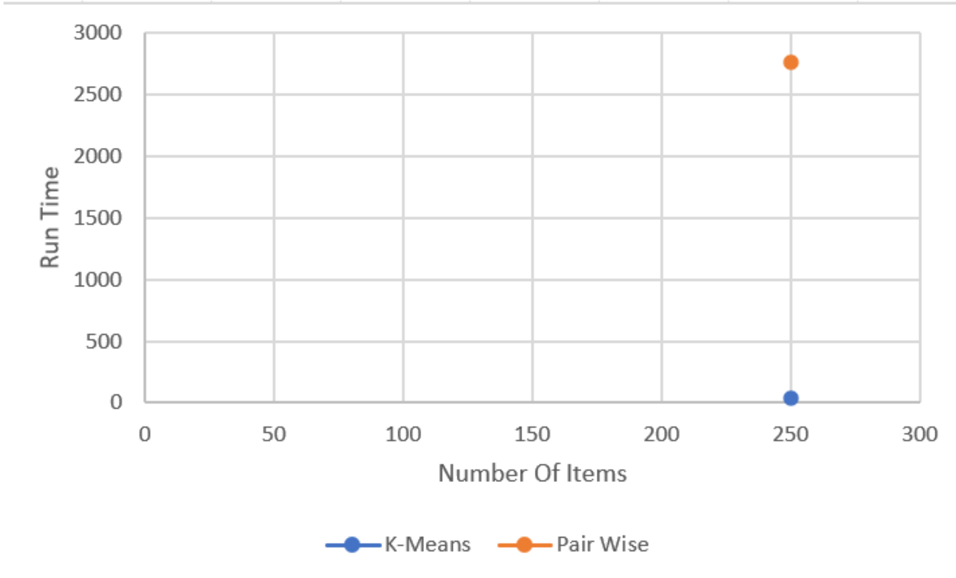

Figure 19. Run time for large warehouse

Looking at the experiments, it is quite evident that the K-means algorithm is superior to the Pairwise, as it proved to require a smaller number of visits to complete customer orders and be more time efficient. Therefore, we expect that the K-Means algorithm to be more effective in our case study, which is provided in the next section.

Chapter 5

Case study

5.1 Introduction of case

The company studied in this thesis is one of the largest importers and distributors of dry and frozen foods in Canada founded in 1952. The company serves premium imported products from around the world across North America's foodservice and retail network. Therefore, the company sells its products B2B (business to business). Its distribution center, located in Dorval, Quebec is established to effectively implement logistics management and improve customer service quality. Its storage layout is shown in figure 20. Currently, the random storage assignment method is used based on the need to utilize the storage space effectively. However, this results in employees becoming unfamiliar with storage locations and increases selection time and distance. To resolve this problem, order information should be analyzed first and then a viable storage assignment program is developed. Based on 1- year order information, the top 10 customers are selected. These customers are well-known, accounting for 5% of overall customers. The period has 132 orders and 178 items. Representing 90% of all stored items.

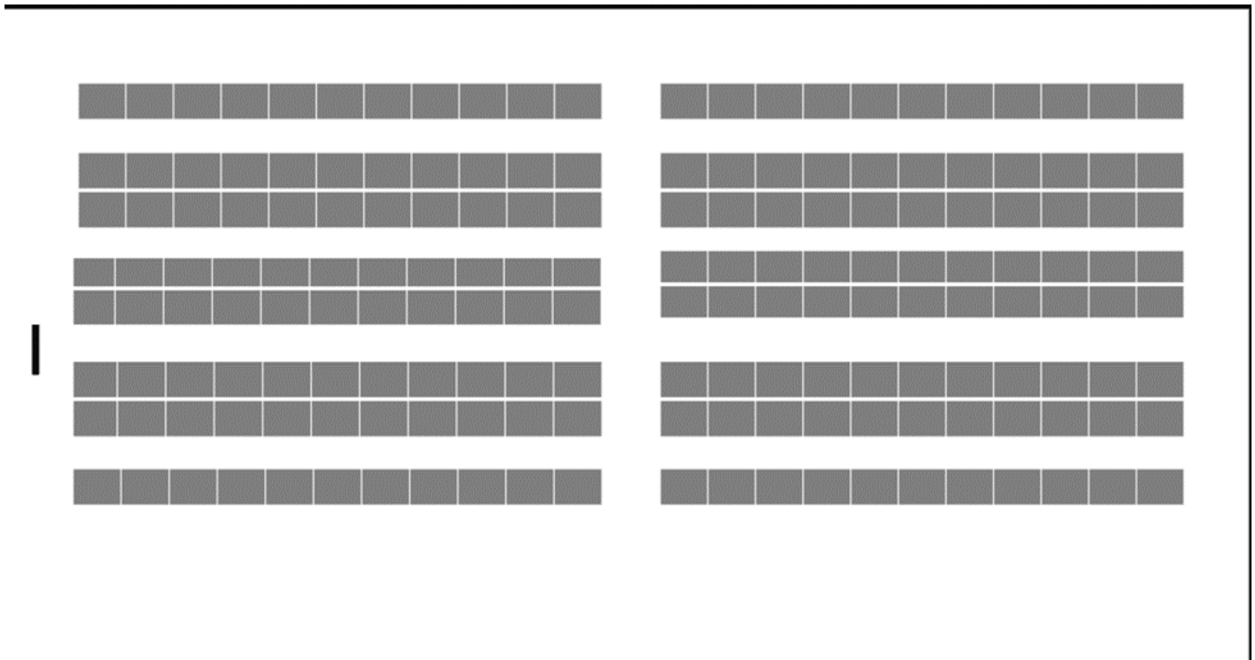

Figure 20. Company's Storage Lay-out

5.2 Order picking efficiency simulation

In this research article on an improved storage strategy, a limited range of commodities are considered. As shown in Figure 20, the warehouse has the following features: a horizontal shelf layout, with two single rows of shelves next to the walls and double rows of back-to-back shelves in the remaining space, one I/O point for the entrance and exit of goods, one cross-aisle with access for 16 aisle, and eleven picking points on each side of each picking aisle, for a total of 176 picking points. The simulation model was developed using Excel. The purpose of the simulation is to assess whether the item associated cluster assignment models is more efficient than the current storage strategy of the company's warehouse and also to discover which of the two models is the best: a decrease in total number of aisle visits indicates an improvement in picking efficiency, meaning that the proposed models are superior to the current set up.

5.3 Results and discussion

To propose the best storage assignment program, we used both algorithms based on 1- year order information. Table 8 and 9 illustrate the clustering results of the Pairwise and K-means algorithms respectively. Once the clusters were made, we calculate the number of visits required by both models and compare them with the number of visits required to fulfill the customer orders using the current set-up. The results are displayed in table 10. Furthermore, we also calculated the total number of items selected from each cluster/aisle which is displayed in table 11. This was done to in order to assigning each cluster to an aisle in the warehouse. The most popular cluster (the greatest number of selected items) assigned to the aisle closest to the I/O entry and the least popular cluster (least number of selected items) assigned to the aisle furthest to the I/O entry. Figures 21 and 22 show the layout of storage location according to the Pair wise and the K-mean model respectively. The aisles with the greater shade of red indicate the most popular clusters and therefore are assigned closest to the I/O entry point and the aisles with the lighter shade of red indicate the least popular clusters and they are placed further away. The results show that both the K-means and Pair wise models are more effective as they reduce over 35% and 25% of the picking distance compared to the random storage assignment that is being currently implemented. Similar to the experimental cases of this thesis, the K- Means model outperforms the Pair wise model and is the recommended model.

Table 8. Result of clustering

Cluster	***Result of clustering from Pairwise model***									
1	**5**	**13**	**32**	**42**	**55**	**60**	**77**	**78**	**113**	**116**	**164**
2	**14**	**16**	**25**	**44**	**46**	**64**	**67**	**69**	**112**	**129**	**176**
3	**6**	**23**	**35**	**50**	**82**	**95**	**101**	**104**	**110**	**115**	**122**
4	**1**	**49**	**98**	**107**	**120**	**121**	**131**	**132**	**144**	**152**	**163**
5	**7**	**52**	**61**	**85**	**90**	**92**	**97**	**114**	**134**	**143**	**150**
6	**9**	**17**	**22**	**33**	**63**	**70**	**123**	**130**	**149**	**158**	**174**
7	**11**	**19**	**24**	**28**	**29**	**56**	**76**	**125**	**140**	**155**	**160**
8	**10**	**12**	**34**	**39**	**59**	**62**	**68**	**73**	**91**	**94**	**173**
9	**15**	**37**	**57**	**66**	**86**	**89**	**100**	**108**	**111**	**118**	**127**
10	**26**	**38**	**48**	**99**	**103**	**109**	**117**	**119**	**141**	**148**	**169**
11	**2**	**8**	**51**	**65**	**75**	**81**	**84**	**106**	**146**	**171**	**175**
12	**18**	**27**	**36**	**58**	**79**	**88**	**105**	**137**	**142**	**153**	**154**
13	**4**	**20**	**41**	**43**	**87**	**136**	**145**	**151**	**156**	**166**	**172**
14	**31**	**71**	**74**	**80**	**83**	**102**	**139**	**161**	**165**	**167**	**170**
15	**3**	**40**	**45**	**72**	**93**	**124**	**128**	**135**	**159**	**162**	**168**
16	**21**	**30**	**47**	**53**	**54**	**96**	**126**	**133**	**138**	**147**	**157**

Table 9. Result of clustering

Cluster	*Result of clustering from the K-mean model*										
1	1	4	38	41	48	52	61	85	92	97	159
2	82	60	65	119	116	54	80	84	26	125	74
3	15	20	21	22	23	25	96	117	156	157	158
4	64	94	91	75	89	32	173	56	76	71	132
5	39	40	45	49	62	176	10	34	122	167	16
6	5	29	55	120	130	17	111	136	11	7	69
7	67	72	141	142	143	138	137	140	165	163	164
8	33	42	63	78	81	98	102	123	149	175	36
9	90	99	114	115	118	50	59	30	68	154	107
10	2	6	37	57	93	100	124	127	134	150	161
11	95	104	105	43	83	139	18	151	73	31	129
12	77	12	109	27	35	103	131	3	162	58	160
13	133	147	112	19	108	135	155	110	128	14	88
14	8	24	51	79	101	106	121	144	146	152	153
15	53	86	87	166	168	169	170	171	172	148	174
16	9	13	44	46	66	113	126	145	47	28	70

Table 10. Results for the Case Study

		Visits			Time		Improvement	
Number of items	Number of clusters	Current	K-Means	Pair Wise	K Means	LP Clusters	K Means	LP Clusters
176	16	1110	676	783	19	277	39%	29%

Table 11. Number of items selected from each aisle

Cluster	**Number Of Items Selected**	
	K-means	Pair Wise
1	11	69
2	96	113
3	11	149
4	211	144
5	506	44
6	31	93
7	19	75
8	206	418
9	87	90
10	101	63
11	65	116
12	124	64
13	27	32
14	135	72
15	21	145
16	60	24

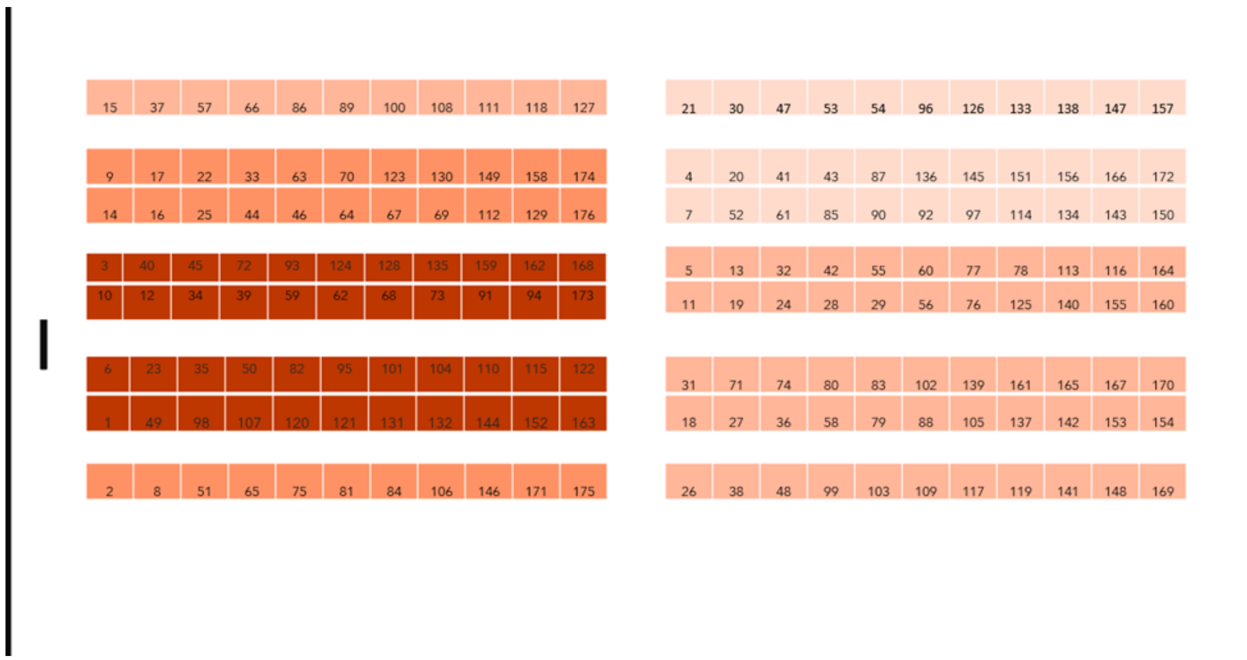

Figure 21. Storage Lay-out using the Pairwise algorithm

Figure 22. Storage layout using the K-Means algorithm

Chapter 6:

Conclusion

6.1 Concluding Remarks

With an effective order picking operation, a warehouse is able to promptly respond to customers orders. Order Picking has become more and more critical in the fast-response, small quantity, and diversified-needs business era. This thesis has proposed two different formulations for the storage location assignment problem and compared them. By doing this we have extended the previous literature as there has been a lack of studies comparing different models in this area. The company studied implements the random storage assignment method in order to utilize the storage space effectively. This results in employees becoming unfamiliar with storage locations and increases order picking time and distance. Hence, we developed a heuristic approach using association and clustering analysis to minimize the order picking time. We showed that in the retailing industry, where customers commonly buy similar products together, the correlated storage policy is a better alternative to random storage policy and that it is advantageous to place these items close to each other and in premium locations. By emphasizing the item association, our model is suitable for orders with multiple items as it effectively shortens the picking distance by minimizing the number of aisle visits compared with random assignment storage method. We successfully showed that both heuristics are a good alternative, but the K-means clustering outperforms the pairwise.

6.2 Limitations

There are several limitations to this study. Although The storage location assignment strategy developed in this thesis was experimented on different orders and layout data, there might be variation in effectiveness if tested with order data and layout of different companies in different industries such as defense, automobile, agriculture, basic metal production etc. The warehouse studied in this thesis is a symmetrical warehouse and contains pallets of the same shape and size, therefore the storage space requirement of items was not a factor that was considered in the storage allocation process. Whereas in other warehouses this may not be the case and the dimensions of the item may play a role in the storage assignment. The pairwise optimization problem looks to maximize an objective function that contains the item index S_{ij}. In this formula the numerator, $P(i \cap j)$ represents the number of orders containing both items *i* and *j*. The denominator represents the number of orders that contain item *i* and the number of orders that contain item *j*. $S_{ij} = \frac{1}{2}$ means that items *i* and *j* are always order together, whereas $S_{ij} = 0$ means that item *i* and *j* are never ordered together. In some cases this could be misleading because item *i* and *j* could appear together in one customer order and never appear again in any order and still be assigned a value of ½. In other words, the algorithm will assume the items have high association, but in reality, it may not be true as they only appear together once in numerous customer orders. In our case this situation is rare, however in other industries some item pairs may only appear once amongst customer orders.

6.3 Future Research directions

Many potential directions for improvement remain to be explored in future research. Firstly, many factors affect the warehouse operations in a company such as picking path, order batching warehouse layout and the storage strategy. However, the focus was only on the storage strategy. All the factors should be considered in order to get the most optimal operations in a warehouse.

Second, the data considered in this paper was only based on a 1- year customer information. In order to get better and more accurate results more customer information could be considered. Furthermore, with the advance in technology, more factors could also be considered such as shape size and the ratio the items are purchased together to help with better and more accurate storage allocation assignment.

Reference

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

Amirhosseini, M. M., and G. P. Sharp. "Simultaneous analysis of products and orders in storage assignment." *Manufacturing Science and Engineering ASME* (1996): 803-811.

Bindi, F., et al. "Similarity coefficients and clustering techniques for the correlated assignment problem in warehousing systems." *Proceedings of 19th International Conference on Production Research*. 2007.

Chan, H. L., Pang, K. W., & Li, K. W. (2011). Association rule based approach for improving operation efficiency in a randomized warehouse. In Proc. Int. Conf. Ind. Eng. Oper. Manage. (pp. 22-24).

Khullar, C. (2021). Development of a warehouse slotting model to improve picking performance (Doctoral dissertation, Concordia University).

Chloe, K., and Sharp, G.P., 1991, "Small parts order picking: design and operation," Material Handling Research Center Technical Report. Georgia Tech. August.

Chuang, Yi-Fei, et al. "Item-Associated Cluster Assignment Model on Storage Allocation Problems." *Computers & Industrial Engineering*, vol. 63, no. 4, 2012, pp. 1171–1177., https://doi.org/10.1016/j.cie.2012.06.021.

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.

Frazelle, E. A., & Sharp, G. P. (1989). Correlated assignment strategy can improve any order-picking operation. Industrial Engineering, 21(4), 33–37.

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

Garfinkel, M. (2005). *Minimizing multi-zone orders in the correlated storage assignment problem*. Georgia Institute of Technology.

Glock, C. H., and E. H. Grosse. 2012. "Storage Policies and Order Picking Strategies in U-shaped Order-picking Systems with a Movable Base." International Journal of Production Research 50 (16): 4344–4357

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

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

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

Hsu, C. M., Chen, K. Y., & Chen, M. C. (2005). Batching orders in warehouses by minimizing travel distance with genetic algorithms. Computers in Industry, 56(2), 169–178.

Hwang, H., & Lee, M. (1988). Cluster algorithms for order picking in an automated storage and retrieval system. International Journal of Production Research, 26(2), 189–201.

Jane, C. C., & Laih, Y. W. (2005). A clustering algorithm for item assignment in a synchronized zone order picking system. European Journal of Operational Research, 166(2), 489–496.

Jewkes, E., Lee, C., Vickson, R., 2004. Product location, allocation and server home base location for an order picking line with multiple servers. Comput. Oper. Res.

31 (4), 623–636. https://doi.org/10.1016/S0305-0548(03)00035-2.

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

Kudelska, I., & Pawłowski, G. (2019). Influence of assortment allocation management in the warehouse on the human workload. Central European Journal of Operations Research, 28(2), 779–795. https://doi.org/10.1007/s10100-019-00623-2

Linn, R.J., and Wysk, R.A., 1987, “An analysis of control strategies for automated storage and retrieval systems,” INFOR, 25, 66-83.

Liu, C. M. (2004). Optimal storage layout and order picking for warehousing. International Journal of Operations Research, 1(1), 37–46.

Malmborg, C.J., 1996, “Storage assignment policy tradeoffs,” International Journal of Production Research, 34, 363–378

Mickleson, G., Thai, V. V., & Halim, Z. (2019). The influence of responsibility shift on warehousing performance: The case of Australia. *The Asian Journal of Shipping and Logistics*, *35*(1), 3-12.

Muppani, V. R., & Adil, G. K. (2008). Efficient formation of storage classes for warehouse storage location assignment: A simulated annealing approach. Omega, 36(4), 609–618.

Petersen, C. G. (1999). The impact of routing and storage policies on warehouse efficiency. International Journal of Operations & Production Management, 19(10), 1053-1064.

Petersen, C. G., Siu, C., & Heiser, D. R. (2005). Improving order picking performance utilizing slotting and golden zone storage. International Journal of Operations & Production Management, 25(10), 997–1012.

Quintanilla, S., Á. Pérez, F. Ballestín, and P. Lino. 2015. "Heuristic Algorithms for a Storage Location Assignment Problem in a Chaotic Warehouse." Engineering Optimization 47 (10): 1405–1422.

Rahman, S. M. M. (2019). Forklift Routing Optimization in a Warehouse using a Clustering-based Approach (Doctoral dissertation, Concordia University).

Rao, S. S., and G. K. Adil. 2013. "Class-based Storage with Exact S-shaped Traversal Routeing in Low-level Picker-to-part Systems." International Journal of Production Research 51 (16): 4979–4996.

Rosenwein, M. B. (1994). An application of cluster analysis to the problem of locating items within a warehouse. IIE Transactions, 26(1), 101–103.

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

Schuur, P. C. (2015). The worst-case performance of the cube per order index slotting strategy is infinitely bad – a technical note. *International Journal of Production Economics*, *170*, 801–804. https://doi.org/10.1016/j.ijpe.2015.05.027

Škerlič, S., & Muha, R. (2017). Reducing errors in the company's warehouse process. *Transport problems*, *12*.

Tompkins, J. A., White, J. A., Bozer, Y. A., Frazelle, E. H., & Tanchoco, J. M. A. (2003). Facilities planning. NJ: John Wiley & Sons.

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

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

Yener, F., & Yazgan, H. R. (2019). Optimal Warehouse Design: Literature Review and Case Study Application. Computers & Industrial Engineering, 129, 1–13. https://doi.org/10.1016/j.cie.2019.01.006

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