12-2016

Dynamic warehouse optimization using predictive analytics.

Parvaneh Jahani
University of Louisville

DYNAMIC WAREHOUSE OPTIMIZATION USING PREDICTIVE ANALYTICS

By

Parvaneh Jahani
M.S. in Industrial Engineering, Sharif University of Technology, 2012
B.S in Textile Engineering, Tehran Polytechnic, 2009

A Dissertation
Submitted to the Faculty of the
J. B. Speed School of Engineering of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy in Industrial Engineering

Department of Industrial Engineering
University of Louisville
Louisville, Kentucky

December 2016

DYNAMIC WAREHOUSE OPTIMIZATION USING
PREDICTIVE ANALYTICS

By

Parvaneh Jahani
M.S. in Industrial Engineering, Sharif University of Technology, 2012
B.S in Textile Engineering, Tehran Polytechnic, 2009

A Dissertation Approved On

November 30, 2016

by the following Dissertation Committee:

Professor Kevin R. Gue, Dissertation Director

Professor John S. Usher

Professor Gail W. Depuy

Professor Olfa Nasraoui

ACKNOWLEDGEMENTS

First and foremost, I would like to express my special appreciation and thanks to my advisor Dr. Kevin Gue. He has always been a tremendous mentor and a model of a critical, elegant thinker for me. I have been very fortunate to be trained, instructed, and grounded in his thorough knowledge of warehousing and logistics. I wish I could have absorbed a fraction of his knowledge in these years. I owe him an enormous debt of gratitude. His advice on both research as well as on my career have been invaluable. I also would like to thank his PhD advisor Dr. John Bartholdi who took the time to listen to our dynamic slotting idea in the early stages and provide valuable advice.

I am greatly appreciative of my committee members: Dr. John Usher, Dr. Gail Depuy, and Dr. Olfa Nasraoui. They have provided their insights and suggestions, which are precious to me. I am grateful to Dr. Suraj Alexander, the chair of the Industrial Engineering Department, for his help in the demand forecasting chapter.

My sincere thanks also goes to my industrial supporters at Intelligrated, Luther Webb, Chris Arnold and Mark Steinkamp, who kept me inspired. I was continually amazed by working with their brilliant team, Operation and Solution Development.

Finally, I would like to thank and dedicate this dissertation to my compassionate and supportive parents Assadollah Jahani and Zahra Karamirad, who experienced all of the ups and downs of my research via Skype. I undoubtedly could not have done this without you, your love, and encouragement.

ABSTRACT

DYNAMIC WAREHOUSE OPTIMIZATION USING PREDICTIVE ANALYTICS

Parvaneh Jahani

November 30, 2016

A *warehouse* is a key component of a logistics system that provides a central location for receiving, storing, and distributing raw materials or manufactured goods. While the objective of a logistics system is reducing the overall inventories and cycle times (the average time between successive deliveries), warehouses are concerned with having the right items, available at the right place, at the right time.

As e-commerce continues to expand and order shipments become smaller, more diverse, and frequent, warehouses must adjust proactive approaches for order fulfillment. Efficient replenishment of the right products into the forward picking areas becomes a more challenging problem in this dynamic environment. The set of items ordered in one month might be completely different from next month's orders. Historical time-based demand data provides valuable information for the models, which have demand as an input. Disregarding the knowledge about the order data behavior over time is costly. One warehousing problem that is highly dependent on product demand and picks is the Forward-Reserve Problem (FRP).

The forward area is a small area of a warehouse with a low picking cost. Therefore, the items of a warehouse compete to be located in this area rather than the reserve area, which has a higher picking cost. Two approaches that are investigated for selecting the SKUs of the fast picking area and the allocated space are the static and the dynamic approaches.

In the case that decisions about the forward area are made periodically (e.g. yearly) and the products' demand patterns are completely ignored, the FRP is *static*. Due to the NP-hard nature of the product assignment to the forward area, we developed two heuristics that solve the large discrete assignment, allocation, and sizing problem simultaneously. We also developed a heuristic that determines the best sizes of the different pick modes within the forward area.

Using a fixed number for the "demand per year" in the static approach does not accurately capture the characteristics of the demand pattern. Replenishing the same product in the same place of the forward area brings about a "Locked" layout of the fast picking area during the planning horizon. By using a dynamic slotting approach, the product pick locations within the warehouse are allowed to change and pick operations can accommodate the variability in the product demand pattern. A dynamic approach can introduce the latest fast movers to the forward area, as an opportunity arises, and stop the replenishment of the products with decreasing turnover rates in this area at the right time. The allocated space to the items in the forward area can also vary over time. We show that on average 39% of the candidate SKUs for the forward area experience the flexibility that the dynamic slotting approach provides. However, updating the forward area periodically in the static approach affect on only 6% of the SKUs.

The primary objective of this dissertation is to formally define the dynamic

FRP. Although real-time order picking and replenishment systems are becoming a pivotal component of today's order fulfillment systems, no consensus in the literature has been made regarding a definition for *dynamic* slotting optimization. One main mission of this research is to define a generic dynamic slotting problem while also demonstrating the strengths of this approach over the static model.

Dynamic slotting continuously adjusts the current state of the forward area with real-time decisions in conjunction with demand predictive analytics. Therefore, the layout of the fast picking area is updated over time with replenishment of the appropriate SKUs, as opposed to traditional methods that periodically reslot the forward area to reach a predefined target map. A powerful slotting methodology not only considers seasonality, but also other types of demand shifts, trends, and frequencies. We explored the methods for demand pattern detection and demand forecasting as well as proposed MIP mathematical model for the dynamic forward-reserve problem for the first time. This model relaxes the major implicit assumptions of the static model and quantifies the effects of the static strategy versus the dynamic strategy.

Extensive numerical experiments are conducted to compare the static FRP solutions, optimal solutions of the dynamic slotting model, and the developed threshold policy, a faster method that performs almost as well as the dynamic MIP model. The results show that the threshold policy solution is always very close to the optimal solution in terms of both the total cost of picking and replenishment and the forward area assignment and allocation. Applying different order data with different demand volatility, we show that the dynamic model always outperforms the static model. The benefits attained from the dynamic model over the static model are greater for more volatile warehouses.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

A Warehousing in logistics systems

A *warehouse* is a building used for the storage of goods, such as manufactured parts, raw materials, spare parts and more. This building has both receiving and shipping areas, in which goods are unloaded from the trucks in the receiving docks and are loaded to the trucks on a smaller scale in the shipping docks. The level of automation differs in different warehouses. While the products are completely picked, packed, and transported automatically in some warehouses, others utilize labor for those activities.

Material flow within the warehouse varies in terms of both type of Stock Keeping Units (SKUs) and the volume. SKUs and demand growth are two subjects that jeopardize any warehouse space management system. These growths will also affect warehouse functions. In some cases, managers must accommodate by adding new products to the already strained capacity of the distribution center. They may also need to apportion available space to those SKUs that have experienced growth in demand.

Every warehouse requires labor, capital, land, and an information system, but providing these resources is costly. One important reason to have a warehouse is to address a highly volatile and changing demand environment. Warehouses provide a buffer for these unpredictable changes. They can also reduce transportation costs by product consolidation before shipping the aggregate volume. Several value added services, such as packaging, returned product services, repairs, testing, inspection, and assembly, are provided by warehouses.

B Warehouse operations

To accomplish the broad scope of warehousing functions (e.g. receiving, storing, picking, sorting, packing, shipping), a warehouse is commonly divided into several functional areas. Figure 1 illustrates the basic flows in a warehouse, starting from the receiving area and ending in the shipping area. After products are received, they are sent to other functional area(s) or directly to the shipping area. The process of unloading the receiving trucks and directly loading the shipping trucks is called cross-docking.

Warehouse operations are labor intensive. Bartholdi and Hackman (2010) report that 55% of warehouse operating costs belong to order-picking. This shows the high potential of order picking and replenishment for warehouse improvements. Not all of the areas of a warehouse have the same picking cost, however, the larger areas and also the farther areas from the Input/Output (I/O) point have a larger picking cost because pickers have to travel longer distances to pick items.

Slot and slotting are two common terms in warehouse studies. A *slot* is the place allotted to the products on the shelf (see Figure 2). The front side view of Figure 2 shows three bays, each having three shelves with four slots per shelf. *Slotting* is the process of determining the item location in a warehouse.

Regarding SKU units, we use the same terms applied in Walden (2005). Figure 3 illustrates the unit levels that describe an SKU in a warehouse. The levels are

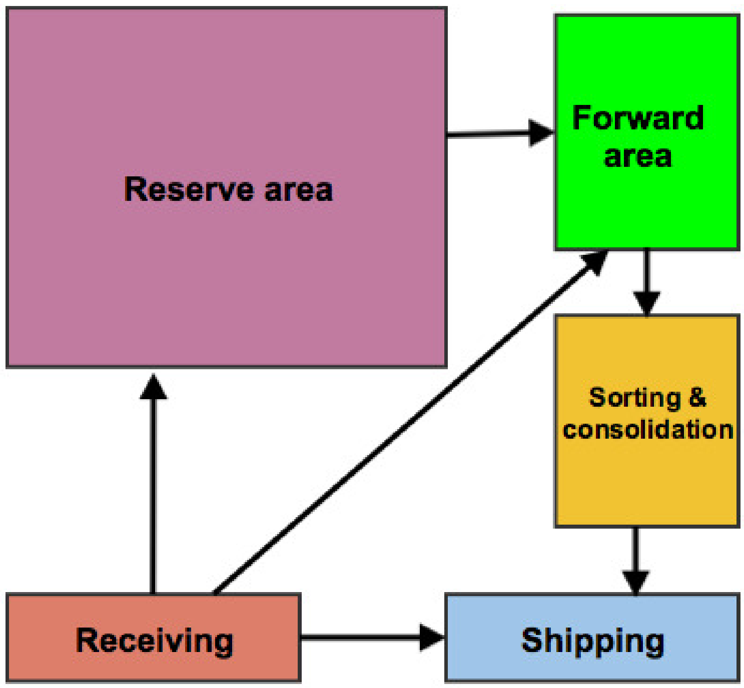

Figure 1. Basic flows in a warehouse

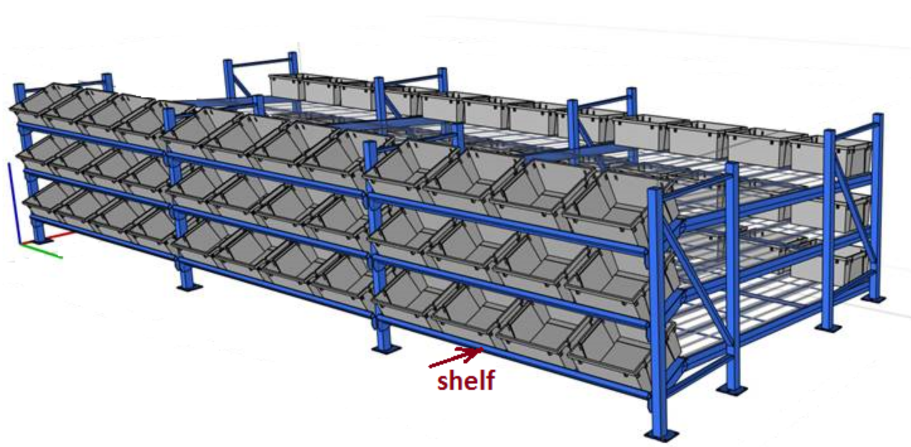

Figure 2. Bay configuration

pallet, tier (level), case (carton), inner, and each (piece), respectively. As Figure 3 demonstrates, a pallet includes layers of cartons. In logistics industry, the number of cartons on a layer is called TI. The number of layers that are stacked on the pallet is called HI. The TI and HI values in Figure 3 are 15 and 6, respectively. Case refers to the carton or box. A quantity per pick is usually less than a full case. The smallest unit of the SKU, which is picked from inside of the case, is called an each or piece.

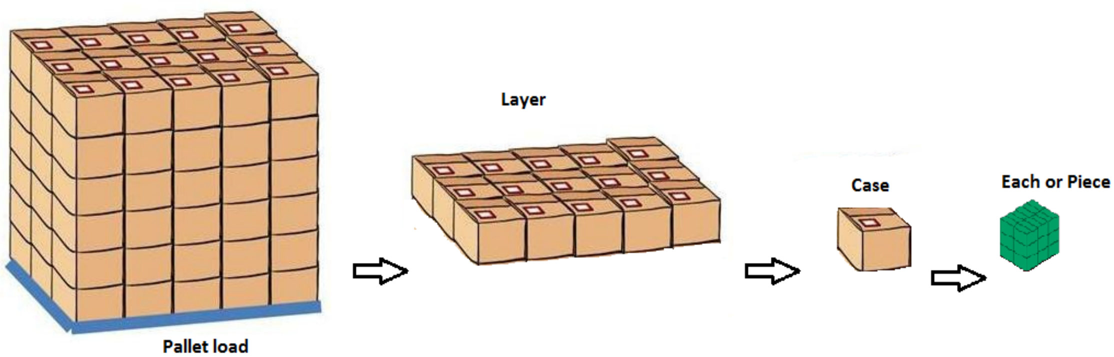

Figure 3. Structure of an SKU

While some zones of the warehouse are replenished by the SKU cases, others can be replenished by the pallets. Likewise, the SKUs can be picked by units or cases. The information about the SKUs and cases such as the length, the width, the height, the weight, the case pack (the number units per case), and the order data affect the warehouse operation decisions. An item in the warehouse can have a single or multiple pick location(s). The products are scanned in the different functional areas for tracking and visibility purposes. Determining the best pick location(s) of the products in the warehouse is challenging. Searching and extracting the SKUs located in the smaller areas need less travel distance. However, the picker should travel more distance to find and pick an item from the larger areas.

C Forward-Reserve Operations

The forward area, or the fast picking area, is a small valuable section of the warehouse with low picking cost. It is expected that the distance the picker traverses in the forward area to pick an order is less than the distance traversed in the reserve area because the forward area is smaller than the reserve area. In addition, the physical nature of rack types in the forward area that we discussed earlier make the pick operation more convenient in the forward area.

The items that go into the forward area are replenished (restocked) from the reserve or bulk area, which is a large area with a high picking cost, to be picked more efficiently. The SKUs are scheduled to be replenished from the reserve area to the forward area. The Warehouse Management System (WMS) keeps track of real-time inventory and schedules the replenishments. The best utilization of the areas with low picking costs plays a significant role in having a more productive warehouse.

The total picking and replenishment costs will increase considerably if we choose a wrong set of SKUs for the forward area. The reason is that inappropriate SKU assignment results in less saving opportunities that the forward area can provide. In addition, the number of replenishments will rise if the allocated slot(s) to the SKUs in the forward area is less than optimal. Allocating more slots than optimal reduces the chance of having a larger set of SKUs in the fast picking area. A clever approach to detect the best SKU for the fast picking area and also the optimal slot allocation enhances warehouse productivity and reduces operational costs.

Since the cost of picking from the forward area is low, one may be inclined to have more products in the forward area due to the low picking cost. Two strategies lead to having more items in the forward area: enlarging the forward area, and allocating less space to each item. The first approach often increases the picking cost, since the picker has to travel a longer distance to pick. The second approach not only involves more items in the restocking process, but also increases the number of replenishments from the reserve area to satisfy the demand. The optimal size of the forward area reduces the total cost of picking and replenishments.

1 Pick mode equipment

To present more details about the pick modules in the forward area, we compare different types of pick modes in this section. The term pick mode refers to a region of the forward area with similar rack characteristics. Typical examples of pick modes in the forward area include five categories: pallet flow racks, carton flow racks, decked racks, steel shelving, and bin shelving. Table 1 compares these different types of racks shown in Figures 4, 5, 6, 7, and 8.

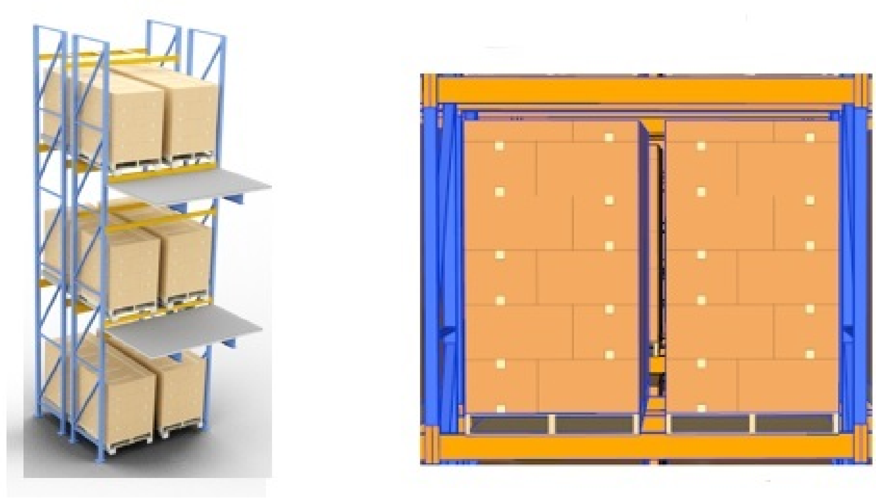

Figure 4. Pallet flow rack

TABLE 1

Comparisons of the different types of pick mode racks

Pallet flow rack	Application	Used for the fastest movers.
	Advantage	Can hold a substantial amount of inventory for a single SKU. Fast replenishment. Replenishment does not interfere with picking.
	Disadvantage	Low SKU density. Can pass few SKU's in a long distance. Low space utilization.
Carton flow rack	Application	Used for fast to medium movers.
	Advantage	Can be replenished by the behind reserve racks. Can hold a substantial amount of inventory and minimize the linear travel. Replenishment does not interfere with picking.
	Disadvantage	Low cube utilization. Density is lower than steel shelving and more expensive than it. Smaller product falls through the skate wheels or rollers.
Decked rack	Application	Used for medium to slow movers.
	Advantage	Can be utilized on the floor level with reserve pallets above. Medium SKU density.
	Disadvantage	Higher cost. Decked rack has a thick support beam compared to shelving.
Steel shelving	Application	Used for slow movers.
	Advantage	High SKU density. Can pass many SKUs in a short distance. High space utilization
	Disadvantage	Not ideal for larger SKUs. Replenishment is cumbersome.
Bin shelving	Application	Used for small slow movers.
	Advantage	Low cost. High space utilization.
	Disadvantage	Can result in excessive travel for a picker. Difficult to pick from the top shelf. Replenishment can interfere with picking.

Figure 5. Carton flow rack

Figure 6. Decked rack

Figure 7. Steel shelving

Figure 8. Bin shelving

D Forward-Reserve Problem

The literature on the forward-reserve problem so far assumes that a warehouse has one small and one large section, named fast and slow picking areas, respectively. In practice, the fast picking area may refer to a shelving area, a section of the carton flow rack, or even an automated system, such as carousels or a miniload system. Our research is not concerned with the specific type of system as long as the picking cost within the area is lower than in the reserve area.

To clarify the configuration of the forward-reserve area in this dissertation, we describe our system as:

The warehouse has a forward and reserve area, where the picking cost from the forward area is less in the reserve area, and the restocking cost from the reserve area to the forward area is more than the cost of picking from the reserve area.

Assuming that the item is available in the reserve area, we perform concurrent replenishment, in which the replenishments can happen during the order picking process.

As opposed to a storage/retrieval machine that travels along the aisle to bring part(s) to the picker, our picking policy in both the forward and reserve areas is picker-to-parts, where the order picker walks or drives along the aisles to pick order lines.

Only one SKU can be stored in a particular slot.

Optimizing the decisions about the forward area is addressed in a well-known warehousing problem called the Forward-Reserve Problem (FRP). Two important decisions of this problem are the selection and the quantities of SKUs in the forward area. The size of the forward area is another critical decision. All previous studies assume that the set of SKUs assigned to the forward area should be known to determine the appropriate size of the forward area. The research in this dissertation solves these three problems simultaneously.

The decisions about the forward and reserve areas are critical. Selecting the wrong products for the forward area is costly. If the slow movers are stored in the forward area, the chance of having more fast movers in the fast picking area is reduced. If the fast movers with high volume movement per year are selected for this area, the slots of the fast picking area will be depleted frequently and having enough inventory for pick operations will require more restocks.

In addition, if the allocated slots to the SKUs in the forward area are higher than the optimal, we can store less products there and so less savings by picks will be achieved. If the allocated slots are less than optimal, it will result in more replenishments. The picking and replenishment costs in the forward-reserve problem can significantly increase with improper SKU assignment and slot allocation. The mathematical models for the traditional forward-reserve problem will be presented in section B of Chapter II.

E Literature Review

The static forward-reserve problem with a continuous allocation of space has been widely researched. Hackman et al. (1990) were the first to develop a mathematical model for the problem. They proposed a greedy heuristic to solve the model. A further contribution is Frazelle et al. (1994), which considers the size of forward area as the decision variable.

Gu et al. (2010) applies a *branch and bound* algorithm to solve the forward-reserve problem. They assert that the heuristic and optimal assignment of SKUs, as well the total cost, are very close together. The optimal stocking strategy is analyzed by Bartholdi and Hackman (2008) and Bartholdi and Hackman (2010) in detail. They compare the optimal strategy with two practical real world strategies: equal space and equal time allocations. Equal space allocation strategy allocates the same amount of space to each SKU. Equal time allocation strategy allocates an equal time supply of each SKU in the forward area.

Hackman and Platzman (1990) extend the fluid model by proposing a generic discrete model based on non-smooth convex relaxation for determining the SKUs and their volume, in an automated forward area. They develop an algorithm with near-optimal solution for the problems, where each allocation is a fraction of standard size bin. One deficiency of the greedy heuristic is that it provides no posterior bound on the performance of the solution (Hackman and Platzman, 1990).

Walter et al. (2013) relax the assumption of continuous space forward area and solve the discrete assignment, allocation, and sizing of the fast picking area. However, they do not solve these three problems simultaneously. They propose four heuristics for solving the discrete forward-reserve problem, which allocates SKUs to shelves (in contrast with slots). Their method is applicable for small size problems. We address this study in chapter II in detail.

Van den Berg et al. (1998) investigate the prior to picking replenishments to minimize the expected labor during the pick period, assuming that prior to picking period there is sufficient time for replenishing the products. They consider the replenishments and demands of forward area SKUs as random variables. Through this method, the number of restocks in objective function of problem Continuous Assignment-Allocation Problem (CAA) is no longer nonlinear. They perform a concurrent replenishment of unit load of SKU i. In other words, their model determines whether the unit load of an SKU is replenished prior to picking period or not. The number of restocks in their model is defined as the sum of the multiplication of binary decision variables (x_{ij}: if the jth unit-load of SKU i is replenished in advanced or not) by the probabilities of having more demand of SKU i than j allocated unit loads. They solve the linear programming relaxation of the discrete model and obtained limited number of fractional solution.

Bozer (1985) discusses the optimal inventory and unit load size in the picking area. He also compares the results of considering the entire warehouse as picking area and separating the picking and reserve areas.

Heragu et al. (2005) investigate the proportion of continuous available space allocated to forward, reserve, and cross-docking areas. They consider five operation areas in the warehouse: receiving, shipping, forward area, reserve area, and cross-docking operation. The authors define four material flows based on these configurations, where all originate from the receiving area and end in the shipping area. The first flow involves a cross-docking operation. Order picking is performed directly from the reserve area in the second flow. The third flow is similar to the forward-reserve problem environment. The last flow stores the product directly in the forward area to perform order consolidation. This research assumes that the product assignment of each area is known. In addition, the average distance traveled to store and retrieve a product in an area is constant and also known. Finally, the model decides whether or not a product should be assigned to a flow, and how much space is to be allocated to each functional area.

Hollingsworth (2003) reduces the number of restocks by performing replenishments directly from the receiving area. It minimizes the replenishment cost containing three components: the number of trips from receiving area to reserve area, the number of trips from reserve area to forward area, and the number of trips from receiving area to forward area. In the domain of restocking cost reduction, Liu et al. (2011) develop a non-linear mathematical fluid model for allocation of storage resources in the forward area. Their order picking system, called the Complex Automated Order Picking System (CAOPS), is automated with multiple dispenser types. They consider four storage modes and safety stocks for each mode.

Some researchers have studied *replenishment prioritization* of the forward area. Gagliardi et al. (2008) propose four heuristics for replenishment policies, two for long-term demand, and two for short-term demand. Unlike Gagliardi et al. (2008) that study the replenishment of the next product, de Vries et al. (2012) consider wave-picking and set replenishment priorities for several workers. The latter develops two replenishment strategies. The first one, Stock-Needs Rule, prioritizes the replenishments based on a ratio dividing the available inventory. The second strategy, the Order-Quantity Based Rule, minimizes the total expected number of zero-picks. The authors extend their study further by comparing three policies for prioritizing replenishments and considering the number of stockouts (de Vries et al., 2014).

In addition to exact optimization techniques and heuristics, the forward reserve problem can be investigated by simulation methods. Venkatadri et al. (2015) propose a simulation model to evaluate the queueing of a given product placement policy in the forward area. This study aims to reduce the congestion in the fast picking area.

We will review the relevant studies to the dynamic slotting optimization problem in chapter V.

F Purpose of the dissertation

To have the best set of SKUs in the forward area continuously, warehouses apply the static FRP periodically. This approach prompts inevitable assumptions. The forward area will have a fixed set of SKUs during a certain period. The products have only one pick location in the warehouse if they are assigned to the forward area. In other words, they should be picked only from the forward area, not the reserve area. However, when the order quantity is occasionally high, it is more efficient to pick the item from the reserve area rather than the forward area. This assumption originates from choosing a fixed number as an annual demand of SKU. Furthermore, refilling the same SKU in the same location with the same replenishment quantity is not the best way to address the SKUs' order fluctuations over time. To combat this, we develop a dynamic model to update the layout of the forward area over time.

We have heard from warehouse managers that they want to avoid the long list of moves generated after running the FRP. The moves are designated for transferring the slow movers to the reserve area. They may only need to update specific areas within their picking area more frequently to keep up with changing demand like seasonality. The dynamic slotting proposed in this research addresses that need. Dynamic approach performs the reslotting of the forward area —updating the forward area layout— on a frequent basis by using the replenishment of empty slots with the correct SKUs without any moves.

Besides, there are critical questions that warehouse managers are challenging with:

1. Which SKUs go into forward area?

2. How many days of inventory should a restocker store in the forward area?

3. How often should a facility reconsider the set of items that go into the forward area and allocated slots?

4. If an SKU is stored in the forward area, are there any cases that it can more efficiently be picked from the reserve area rather than the forward area?

The first two questions have been extensively studied with an assumption of continuous space of the forward area. The last two questions have not been answered in literature. The problem addressing the integral solution of assignment, allocation, and sizing simultaneously, which consider the slot and SKU geometries, have not been answered yet.

There are two major weaknesses in previous studies on the FRP. First, they assume that the space of the forward area is continuous, when most often it is discrete. Assuming cubic product movement per year and disregarding slot and SKU dimensions, they allocate cubic space of the forward area to the selected items for this area. In addition, current approaches assume decisions about the forward area are one-time decisions during the planning horizon. As a result the fast picking area is replenished with the same products for a long time. These approaches miss the opportunity of updating the layout of the forward area based on the SKUs' demand patterns over time. SKU assignment and allocation in the fast picking area are not long term decisions because of the ever-changing demand environment.

The first shortcoming of previous studies creates a gap between the "state of art" and the "state of practice" in the forward-reserve problem. The state of practice does not allow allocation of the continuous space of the forward area to SKUs, while the state of art is based on this assumption. The space wasted while allocating the SKUs to the slots is unavoidable. Geometry considerations for both slots and SKUs are necessary. We develop a discrete FRP model, which relaxes these continuous model assumptions.

The contributions of this dissertation are:

Contribution 1: For the static forward-reserve problem, we develop two heuristics that address the discrete assignment, allocation and the sizing of the forward-reserve problem for large size problem. As opposed to the first heuristic, the second heuristic takes the slots and SKUs' dimensions into account. The algorithms are fairly simple, fast, and applicable for a real world warehouse. The solutions are quite close to the optimal.

Contribution 2: We propose an algorithm for both profiling and slotting optimization simultaneously. This algorithm determines the best size of each pick mode within the forward area, as well as respecting the different rack configurations, pick technology specifications and replenishment policies of the pick modes. The SKU and demand growths, the active period of the fast movers based on their order date, and the optimal case orientation in each slot are the subjects that have also been taken into consideration.

Contribution 3: To the best of our knowledge, we are the first to propose the dynamic forward-reserve problem. We developed the first MIP formulation for the dynamic assignment and allocation of the forward area. The main contribution of this research is quantifying the effects of the traditional wisdom of running the static model in certain intervals assuming constant demand. We elaborate on demand trend analysis prior to the development of the dynamic forward-reserve model. We first propose an Artificial Neural Network (ANN) based model for pattern recognition of the different types of demand trends. Further, we develop an algorithm for forecasting the demand quantity. The method of forecasting is dependent of the class of demand trend recognized in the previous stage. The algorithm is the combination of the AutoRegressive Integrated Moving Average (ARIMA) model for predicting smooth demand trends and the Markov-based bootstrapping method for predicting intermittent demand pattern.

The remainder of the dissertation has been organized as follows. Chapter II focuses on static forward-reserve problem and presents two intuitive simple heuristics for discrete FRP. We propose an algorithm in Chapter III for determining the best sizes of pick modes within the forward area. A model for predictive analytics of products' demand is developed in Chapter IV. We propose the MIP formulation for the dynamic forward-reserve problem and compare the static and dynamic model with experimental design in chapter V. Chapter VI concludes the dissertation.

CHAPTER II

THE STATIC FORWARD-RESERVE PROBLEM

A Introduction

This chapter addresses the *static discrete* assignment, allocation, and sizing problems of the forward area. The term static suggests that the decisions about the forward area are made periodically (e.g. yearly). This approach disregards the SKUs' demand trends during the planning horizon. Thus, in the static forward-reserve problem, the demand term represents the total demand of an SKU during the past year or in a forecast year.

The term discrete suggests that discrete units of the SKUs can be stored in discrete slots. This concept avoids allocating a portion of a slot to an SKU, which is allowed in the continuous space model but not in practice. Previous research in this area has focused on the continuous forward-reserve problem. No more than one type of SKU can be kept in the discrete model. The discrete model considers lost space resulting from differences in slots and SKU dimensions. Solving the allocation problem in a continuous space model causes many SKUs having allocated space of less than one slot, which is impractical.

Rounding down the solution of continuous space model threatens the optimal solution. It has the risk of removing SKUs with less than one allocated space, from the forward area. Further, if the case width is larger than the allocated slot(s) width, the stored unit will no longer fit the allocated slot(s).

Rounding up the solution of continuous space model may also assign the ineligible slow movers with very small space (close to zero), to the forward area. Consequently, the eligible ones will have to leave the set of SKUs of forward area or get fewer slots. Allocating few slots will increase the number of replenishments. To address the aforementioned shortcomings, this chapter tackles the discrete forward-reserve problem considering both slot and SKU dimensions. There is also the need for solving the assignment, allocation and sizing problems, simultaneously, for a large number of SKUs.

B The continuous model for space allocation

The fluid model for space allocation assumes that the forward area can be continuously subdivided. In other words, each SKU is considered as an incompressible fluid rather than discrete units that are packed in cartons. Since the solution of the continuous space model is the basis of our proposed algorithms for the discrete model, we first review Hackman et al. (1990)'s model for allocation and assignment of SKUs to the forward area in this section.

The flow rate of SKU i f_i is the demand of SKU i per year expressed as volume per year, e.g. cubic feet per year. Variable f_i can be computed as follows (Bartholdi and Hackman, 2010):

$$f_i = \frac{d_i}{b_i} o_i, \tag{1}$$

where d_i is the demand of SKU i per year (units per year), b_i is the number of selling units within a storage unit (case), and o_i is the volume per storage unit of SKU i.

Hackman et al. (1990) assume that the pick quantity for SKU i in the forward area is always less than the full allocation of an SKU in the forward pick area. A restock is scheduled when the inventory level of slots drop to a certain threshold. The number of restocks per year is defined as follows:

$$r_i = \left\lceil \frac{f_i}{v_i} \right\rceil \tag{2}$$

where v_i is the volume of SKU i stored in forward area. In the fluid model, it is also assumed that the replenishment can be fully satisfied in one trip. In other words, the entire restock quantity is always less than the restocker capacity. The restocking cost includes the following costs (Bartholdi and Hackman, 2008):

1. The travel between forward and reserve areas, which depends on the warehouse layout.

2. The average travel within the reserve area, which is based on "random storage" in this area.

3. The negligible cost of traveling within the forward area, since the size of forward area is a small fraction of the warehouse.

4. The fixed cost of handling storage units while restocking.

Due to the fixed and small nature of these cost components, the number of restocks multiplied by the associated restocking cost fully represents the total restocking cost. Hackman et al. (1990) develop a heuristic solution algorithm, with a priori and posteriori tests for optimality, to determine which SKUs go into the forward area. In their model, the space allocated to each SKU is continuous. Their objective function maximizes the benefits (pick savings, less restock costs) as below.

Problem CA (Continuous Allocation Problem):

$$Maximize \quad \sum_{i \in A} sp_i - c\frac{f_i}{v_i} \tag{3}$$

$$\sum_{i \in P} v_i \leq V \tag{4}$$

$$v_i > 0 \tag{5}$$

where V is the volume of the forward area, p_i is number of picks of SKU i per unit time, A is the set of SKUs that go into forward area, s is the savings per pick if stored in the forward area (s is equal to the difference between cost of picking from the reserve area and forward area), and c is restocking cost.

The capacity constraint refers to the maximum inventory of each SKU selected to be assigned in forward area. Given the set of items allocated to the forward area and setting optimal Lagrange multiplier for constraint 4, the space allocation vector $v = \{v_1, ..., v_i\}$ can be computed as below (Hackman et al., 1990):

$$v_i^* = \frac{\sqrt{f_i}}{\sum_{i \in A} \sqrt{f_j}} V. \tag{6}$$

The following knapsack problem considers the allocation and assignment of SKU i to forward area together.

Problem CAA (Continuous Assignment-Allocation Problem):

$$Maximize \quad \sum_{i=1}^{n} \left(c_1 p_i + c\frac{f_i}{v_i} \right) x_i + \sum_{i=1}^{n} (c_2 p_i)(1 - x_i) \tag{7}$$

$$\sum_{i=1}^{n} v_i x_i \leq V \tag{8}$$

$$x_i \in \{0, 1\} \tag{9}$$

The reformulation of problem CAA based on $c_i = c_2 - c_1$ is

$$Maximize \quad \sum_{i=1}^{n} \left(sp_i - c\frac{f_i}{v_i} \right) x_i \tag{10}$$

$$\sum_{i=1}^{n} v_i x_i \leq V \tag{11}$$

$$x \in \{0,1\} \tag{12}$$

where c_1 and c_2 are the cost of picking from the forward and reserve areas, respectively, and x_i is the binary decision variable determining if item i is assigned to the forward ($x_i = 1$) area or not ($x_i = 0$). Problem CAA is NP-complete. A well known heuristic for solving this category of problem is to rank the SKUs according to their "bang-for-buck," which in our case is

$$\frac{\text{benefit}_i}{v_i} = \frac{sp_i - c\frac{f_i}{v_i}}{v_i}. \tag{13}$$

We fill the knapsack until adding the additional SKU exceeds the capacity of the forward area. Since the set of SKUs assigned to the forward area is unknown, the labor efficiency ratio le_i is used to sort the SKUs. This ratio is equivalent to bang-for-buck. Substituting equation 6 in CAA, the labor efficiency of SKU i is defined as:

$$\frac{p_i}{\sqrt{f_i}}. \tag{14}$$

However, this method cannot be implemented directly because we do not know the v_i's a priori.

The Hackman et al. (1990) heuristic for solving problem CA is summarized as below:

1. Rank all SKUs in order of non-increasing $\frac{p_i}{\sqrt{f_i}}$

2. **for** i=1:No. of SKUs(N) **do**

   a) Use equation 6 to compute the space allocation vector $v = \{v_1, ..., v_i\}$ corresponding to the set of $S_i = \{1, ..., i\}$ SKUs in forward area.

   b) Use Equation 3 to compute the total benefit for each set of S_i.

   end for

3. Select the set of S_i with maximum value of total profit satisfying constraint 4.

Both a priori and a postriori tests are checked for the optimality of the heuristic algorithm in Hackman et al. (1990). Using a numerical example, the authors show that their proposed algorithm outperforms a conventional method of ranking based on number of picks per unit time.

Gu et al. (2010) evaluate the gap between the Hackman et al. (1990) greedy heuristic and optimal solution. They conclude that this gap is negligible for real world problems, where the number of SKUs is large enough.

Frazelle et al. (1994) extends the Hackman et al. (1990) study by treating the capacity of the forward area as a decision variable. Determining the optimal size of the forward area, they first solve the assignment-allocation subproblem with fixed size of forward area. They show how the forward area sizing problem can decrease the picking costs. In their numerical example, they reduce the picking costs from \$0.25 to \$0.14 by decreasing the size of the forward area to 32% of its original size. The order picking and replenishment costs are discussed in detail in their work. In addition to storage-volume capacity, they consider congestion constraint.

The congestion constraint plays a more important role in AS/RS systems than cart picking system. Cart picking systems with wide aisles allow more than one order picker to travel among the same aisle (Frazelle et al., 1994). The congestion constraint is also a function of time spent in the forward area. The authors develop two algorithms for solving the assignment, allocation, and sizing forward reserve problem. Another extension of the fluid model is the case when several forward areas exist (Subramanian, 2013).

C A discrete model for space allocation

In the following section, we review the methodology of Walter et al. (2013) to solve the discrete forward-reserve problem. Then, we propose a greedy heuristic solution procedure for solving the discrete forward-reserve problem. We also relax the assumption of one SKU in each shelf by considering a variety of SKUs in one shelf. As a result, more than one SKU may be assigned to a shelf with a certain number of slots.

SKUs are slotted into carton flow rack with both SKUs and slots' dimensions considerations. All details, including dimensions of storage containers and shelves are accounted for. Generally, two kinds of SKUs cannot be located to one slot. The SKUs wider than the slots' width have different lower bounds for the allocated number of slots decision variable. For example, if the width of SKU i is 18 inch and the slot width is 12 inches, the discrete solution should assign at least two slots to SKU i. The minimum number of allocated slots to the SKU in the forward area varies based on SKU and slot dimensions. Previous studies assume that the width of an SKU is always less than the opening location.

After defining the discrete FRP, we propose two heuristics to find a discrete solution. The optimal size of forward area is also investigated through the proposed algorithms.

Walter et al. (2013) investigate the discrete forward-reserve area with equal size shelves, each containing only one SKU. They assume that SKU i can be stored with a certain number of units (e.g. $1a_i, 2a_i, 3a_i, ..., ja_i, ..., na_i$) in the forward area, where a_i is the number of units of SKU i that can be stored in one shelf. The storage mode j is the number of allocated shelves. Therefore, $a_{ij} = ja_i$ is the number of units of SKU i associated with each storage mode j and w_{ij} is the space required. They developed three discrete forward-reserve problems:

1. The discrete forward-reserve allocation model.
2. The discrete forward-reserve assignment and allocation model.
3. The discrete forward-reserve allocation and sizing model.

The authors then compare four repair heuristics with an optimal discrete solution using the Bitran and Hax (1981) algorithm. In what follows, we use the notation of Walter et al. (2013).

Among four heuristics (R_1, R_2, R_3, R_4) described in Walter et al. (2013), R_4 outperforms the others. The gap between fluid models and their discrete counterparts for the defined instances are negligible from the practitioners' point of view. They leave the solution procedure for large sized problems open for future research.

The discrete version of problem CAA is mathematically equivalent to the *multiple choice* knapsack problem (See problem DAA, Discrete Assignment-Allocation Problem.) In this type of knapsack problem, the items are categorized into k classes, and exactly one item must be taken from each class. In discrete problem, the storage modes are same as the items in multiple choice knapsack problem. If an SKU is selected for the forward area, it must take exactly one type of storage mode.

Binary variable x_{ij} not only decides about the assignment of an SKU to the forward areas, but also determines which storage mode j is optimal for SKU i. Again, the storage mode is the number of shelves allocated to the SKU. If SKU i is assigned to the forward area, then $x_{i0} = 0$, and otherwise $x_{i0} = 1$. The discrete problem can be formulated as below.

Problem DAA **(Discrete Assignment-Allocation Problem)**:

$$Minimize \quad \sum_{i \in P} \sum_{j=1}^{n_i} \left(c_{1i} p_i + c_i \frac{d_i}{a_{ij}} \right) x_{ij} + \sum_{i \in P} (c_{2i} p_i) x_{i0} \tag{15}$$

$$\sum_{j=0}^{n_i} x_{ij} = 1 \quad \forall i \in P \tag{16}$$

$$\sum_{i \in P} \sum_{j=1}^{n_i} w_{ij} x_{ij} \leq S \tag{17}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in P; j = 0, ..., n_i \tag{18}$$

where n_i is the upper bound of storage mode for SKU i and P is the set of all SKUs. c_{1i} and c_{2i} are the costs of picking of SKU i from forward and reserve areas, respectively, c_i is the restocking cost for SKU i, S is the size of forward area in terms of number of shelves, and p_i is the number of picks for SKU i. Assume $C_i^{add} = c_{2i} - c_{1i}$. The reformulation of objective function of problem DAA is:

$$C(x) = \sum_{i \in P} \sum_{j=1}^{n_i} \left(c_{1i} p_i + c_i \frac{d_i}{a_{ij}} \right) x_{ij} + \sum_{i \in P} (c_{2i} p_i) x_{i0} \tag{19}$$

$$= \sum_{i \in P} \sum_{j=1}^{n_i} c_i \frac{d_i}{a_{ij}} x_{ij} + \sum_{i \in P} C_i^{add} p_i x_{i0} + \sum_{i \in P} c_{1i} p_i \tag{20}$$

Minimizing $C(x)$ is equivalent to minimizing

$$C'(x) = \sum_{i \in P} \sum_{j=1}^{n_i} c_i \frac{d_i}{a_{ij}} x_{ij} + \sum_{i \in P} C_i^{add} p_i x_{i0} \tag{21}$$

If only one shelf is allocated to SKU i, the aggregate restock cost of SKU i is $q_i = c_i \frac{d_i}{a_i}$. The additional cost generated by SKU i if picked from the reserve area is

$q_i^R = C_i^{add} p_i$. The minimum storage is defined as j_i^{min}, which leads to $c_i \frac{d_i}{a_{ij_i^{min}}} < C_i^{add} p_i$ for the first time. If the shelves are equally sized, then problem DAA becomes:

$$Minimize \quad C(x) = \sum_{i \in P} f_i(x_i) \tag{22}$$

$$S.t. \sum_{i \in P} x_i \leq S \tag{23}$$

$$x_i \in \{0, 1, ..., n_i\} \quad \forall i \in P; \tag{24}$$

where:

$$f_i(x_i) = \begin{cases} \frac{q_i}{x_i} & \text{if } x_i^* \geq 1 \\ q_i^R & \text{otherwise} \end{cases}$$

Walter et al. (2013) compare the discrete optimal solution of assignment and allocation problems with two repair heuristics R_2 and R_4. Both the discrete optimal solution and the repair heuristics use the Bitran and Hax (1981) algorithm and enumerate all possible SKU selections $A \subseteq P$, where A is an alternative set of SKUs going into the forward area. The authors substitute the total number of shelves (slots) S for the volume of the forward area V in equation 6:

$$v_i'^* = \frac{\sqrt{f_i}}{\sum_{i \in A} \sqrt{f_j}} S. \tag{25}$$

Then, the allocated spaces obtained from equation 25 are forced to have at least j_i^{min} slots to all SKU $i \in A$ via Bitran and Hax (1981) algorithm, and all $A \subseteq P$ fulfilling $\sum_{i \in P} j_i^{min} \leq S$ are considered a reasonable selection. Their procedure requires checking all possible SKU-selection $A \subseteq P$ to enumerate the reasonable SKU selections, which is equal to $2^{|P|} - 1$ selection of SKUs. This means an exponential number of instances should be solved, which is computationally expensive.

Walter et al. (2013) study four heuristics for solving the equally sized shelves DAA problem. They refer to this problem as the discrete forward-reserve assignment and allocation problem ($DFRAAP_{ES}$). Their best repair heuristic for altering non-integral solution obtained from the fluid model to an integer solution is R_4. They consider z_i^* as the optimal non-integral solution of fluid model, and x_i^R is the round-down non-integral solution elements of the fluid model. Walter et al. (2013)'s procedure to find the discrete assignment and allocation of SKUs in forward area, $DFRAAP_{ES}$, using heuristic R_4 is as below:

for all possible SKU-selections **do**

1. Obtain the fluid model solution according to $z_i^* = \frac{S\sqrt{q_i}}{\sum_{k \in A} \sqrt{q_k}}$ via the Bitran and Hax algorithm, forcing $z_i^* \geq j_i^{min}$ for all $i \in A$.

end for

2. Determine the continuous optimal SKU assignment.

3. Round down the non-integral solution elements (allocations) of the fluid model ($x_i^R = \lfloor z_i^* \rfloor$).

4. Compute $d_i(x_i^R) = f_i(x_i) - f_i(x_i + 1)$ for all non-integral solution elements of fluid model.

5. Sort all SKUs in forward area in order of non-increasing $d_i(x_i)$.

6. Increase the number of allocated shelves for each of the δ SKUs by one until $\sum_{i \in P} x_i^R = S$ (δ is the difference between S and the number shelves allocated in step 2).

7. Compute $C(x) = \sum_{i \in P} f_i(x_i)$.

Replicating the Walter et al. (2013) model, we implemented Bitran and Hax (1981) algorithm, which is a recursive procedure that repeatedly allocates shelves according to equation 25 until all SKUs i received at least j_i^{min} shelves. Those SKUs in A that have received the number of shelves less than the lower bound ($z_i^* < j_i^{min}$) are shown with P_t^-, and those SKUs in A that have received the number of shelves more than upper bound n_i are shown with P_t^+. The upper bound to find the optimal solution of DAA in Walter et al. (2013) is $n_i = S - |A| + 1$.

In each iteration t, the total gap with lower bound (Δ^-) and upper bound (Δ^+) corresponding to P_t^- and P_t^+ are determined, respectively. If $\Delta_t^+ \geq \Delta_t^-$, then $z_i^* = n_i, \forall i \in P_t^+$. Otherwise, $z_i^* = j_i^{min}, \forall i \in P_t^-$. The remained space at iteration t of this algorithm is shown with S_t. Let assume the SKUs with $z_i^* < j_i^{min}$ reach their lower bound. The remained space $S_t = S - \sum_{i \in P_t^-} j_i^{min}$ should be allocated to the rest of SKUs and resolve the equation 25 with the new total space S_t. Then the next iteration (t+1) of this recursive algorithm is performed. The procedure will stop when all SKUs receive at least j_i^{min} and at most n_i shelves. At the end of this procedure, the non-integral solution of the fluid model is found.

$DFRAAP_{ES} - R_4$ is only solvable in reasonable time for small problems. Walter et al. (2013) solved the assignment-allocation problem for a warehouse with 12 and 24 SKUs using their heuristics R_2 and R_4. Two implicit limitations of their work are:

1. The first assumption is related to the size of the problem. Since all SKU assignments are generated in both discrete optimum and repaired heuristics, their methodology is not applicable for real size problems.

2. None of their problems considers the joint assignment, allocation and sizing of discrete forward-reserve problem.

They also implicitly assume that the SKUs are stored in the shelf not slot. Note that a shelf consists multiple slots. This assumption fails to address the geometric considerations of both slots and SKUs. Specifically, the case that SKU width exceeds the slot's width are not addressed. Situations in which there is more than one SKU per shelf are not addressed.

1 Heuristic G_1

Heuristic G_1 solves the assignment and allocation problems simultaneously. The labor efficiency in G_1 is obtained from following formula:

$$\frac{c_i^{add} p_i}{\sqrt{q_i}} = \frac{\lambda_i q_i}{\sqrt{q_i}} = \lambda_i \sqrt{q_i} \tag{26}$$

Heuristic G_1 is as follow:

1. Sort all SKUs in order of non-increasing labor efficiency.

Alternative set of SKUs for the forward area = [ ]

for i=1:No. of SKUs(N) **do**

2. Add one SKU to the alternative set of SKUs.

3. Compute $z_i^* = \frac{S\sqrt{q_i}}{\sum_{k \in A} \sqrt{q_k}}$

4. Let $x_i^R = \lfloor z_i^* \rfloor$.

5. Compute $d_i(x_i) = f_i(x_i) - f_i(x_i + 1)$ for all SKUs in the alternative set for the forward area ($i \in A$).

6. Sort all SKU in order of non-increasing $d_i(x_i)$.

7. Having δ as the difference between S and the number shelves allocated in step 3, Increase the number of allocated shelves for each of the δ SKUs by one until $\sum_{i \in P} x_i^R = S$.

8. Compute $C(x) = \sum_{i \in P} f_i(x_i)$

end for

9. Select the minimum $C(x)$.

Walter et al. (2013)'s procedure for finding DAA solution distributes δ among a fixed set of SKUs, which is called the continuous optimal SKU assignment. Afterward, they discretize the respective fluid model allocations according to R_4. Their solution guarantees that each SKU receives at least j_i^{min} shelves.

On the other hand, G_1 may delete some SKUs from the fluid model assignment solution, if the round down allocation is zero. The SKUs with no allocated shelf go into the reserve area. In summary, the optimal assignment of the continuous model may or may not be same as the assignment generated.

2 Testing the model

We now elaborate on a test instance generation for DAA problem with respect to Walter et al. (2013)'s tests. The three changing parameters in instances are as below:

1. ***Set of SKUs (P)***. We choose $|P| \in \{12, 15, 18\}$ as the total number of SKUs.

2. ***Total number of available shelves (S)***. This parameter corresponds to coefficient r, where $r \in \{1/2, 1/2, 2/3\}$. The number of shelves is set to $S = rP$. So the number of SKUs assigned to the forward area is a portion of total SKUs.

3. ***The aggregate restock cost (q_i)***. q_i is the aggregate number of restocks if minimum number of slots is given to the SKUs in the forward area. For the our numerical examples, we assume that this parameter is independent uniformly distributed in the range $[0.1, 0.2)$ for product category 1, $[0.2, 0.4)$ for product category 2 and $[0.4, 0.8)$ for product category 3. For total additional costs, we assume $q_i^R = \lambda_i q_i$, where λ_i is distributed in the intervals (0.1,.5) with probability p_λ and (1.5,2) with converse probability. The values of p_λ are selected as $p_\lambda \in \{0.2, 0.5, 0.8\}$.

Similar to Walter et al. (2013) study, we consider three product categories with a distribution D_2. The probability of product belonging to category 1, 2, and 3 are .3, .4 and .3, respectively. We generate 100 independent instances for $3^3 = 27$ experiments (three varying parameters P, S and p_λ each with three choices). Four performance measures are:

- ACI: Average percentage of cost improvement of G_1 over $DFRAAP_{ES} - R_4$.
- ACD: Average percentage of cost difference between G_1 and its continuous model counterpart.
- SA: Same Assignment (but different allocation) in $DFRAAP_{ES} - R_4$ and G_1 via 100 replications.
- SAA: Same Assignment and allocation in $DFRAAP_{ES} - R_4$ and G_1 via 100 replications.

TABLE 2

Heuristics $DFRAAP_{ES} - R_4$, G_1 and continuous space model comparisons

		$\lvert P\rvert = 12$				$\lvert P\rvert = 15$				$\lvert P\rvert = 18$			
	P_λ	ACI	ACD	SA	SAA	ACI	ACD	SA	SAA	ACI	ACD	SA	SAA
r=1/3	0.2	0.12	2.15	83	74	0.13	2.07	69	68	0.10	1.99	66	65
	0.5	0.18	2.90	76	69	0.17	3.04	70	70	0.07	2.82	82	82
	0.8	6.84	0.21	100	58	0.00	3.74	100	100	0.16	4.29	84	80
r=1/2	0.2	0.22	2.12	65	64	0.28	2.29	54	54	0.11	2.72	61	61
	0.5	0.63	2.59	56	51	0.40	3.01	68	65	0.16	3.93	71	71
	0.8	2.21	0.15	100	58	6.13	0.73	38	16	2.56	0.90	100	64
r=2/3	0.2	0.78	2.18	60	54	0.44	2.23	54	52	0.30	2.88	48	46
	0.5	1.42	2.57	79	68	2.99	2.55	66	42	1.57	3.04	39	33
	0.8	1.38	0.11	100	42	0.86	0.33	100	74	3.28	0.29	100	18

Table 2 lists the results of our computational study on the $DFRAAP_{ES} - R_4$, G_1, and continuous model. The numbers of this table represent the cost improvement of G_1 over $DFRAAP_{ES} - R_4$ or ACI is considerable for larger problems. The largest cost improvement (6.13%) happens for the moderate size of the forward area ($r = 1/2$), when $P = 15$. The assignment and allocation solutions of $DFRAAP_{ES} - R_4$ and G_1 are respectively similar in 38 and 16 out of 100 generated instances in this case. The assignment solutions of these two heuristics is more similar for lower additional costs resulted by picking more orders from the reserve area rather than the forward area (larger P_λ). As expected, the cost difference between G_1 and its continuous counterpart becomes smaller for larger size forward areas. The adverse effect of the continuous model is more tangible for smaller forward area.

Figure 9 confirms G_1 outperforms $DFRAAP_{ES} - R_4$ in all experiments. This figure also shows large probability of lower additional cost of picking from the reserve area (large P_λ), specifically when the number is SKUs is low, results in smaller gap between G_1 and its continuous counterpart. Note that our heuristics are based on greedy algorithm, while $DFRAAP_{ES} - R_4$ is based on Bitran and Hax (1981) algorithm.

Regarding the solution time, algorithm $DFRAAP_{ES} - R_4$ applies Bitran and Hax (1981) algorithm with running time $o(|p|^2)$. Every generated combination set $A \subseteq P$ (with at most $o(2^{|p|})$ running time) applies Bitran and Hax (1981) algorithm. Therefore, $DFRAAP_{ES} - R_4$ solution time is $o(|p|^2 2^{|p|})$ and it cannot be solved within reasonable time for large size problems. However, G_1 delivers the assignment and allocation solution of $P = 1000$ in 10.42 seconds.

In following, we address discrete assignment, allocation, and sizing problems together with no restriction on the width of SKUs that go into the forward area.

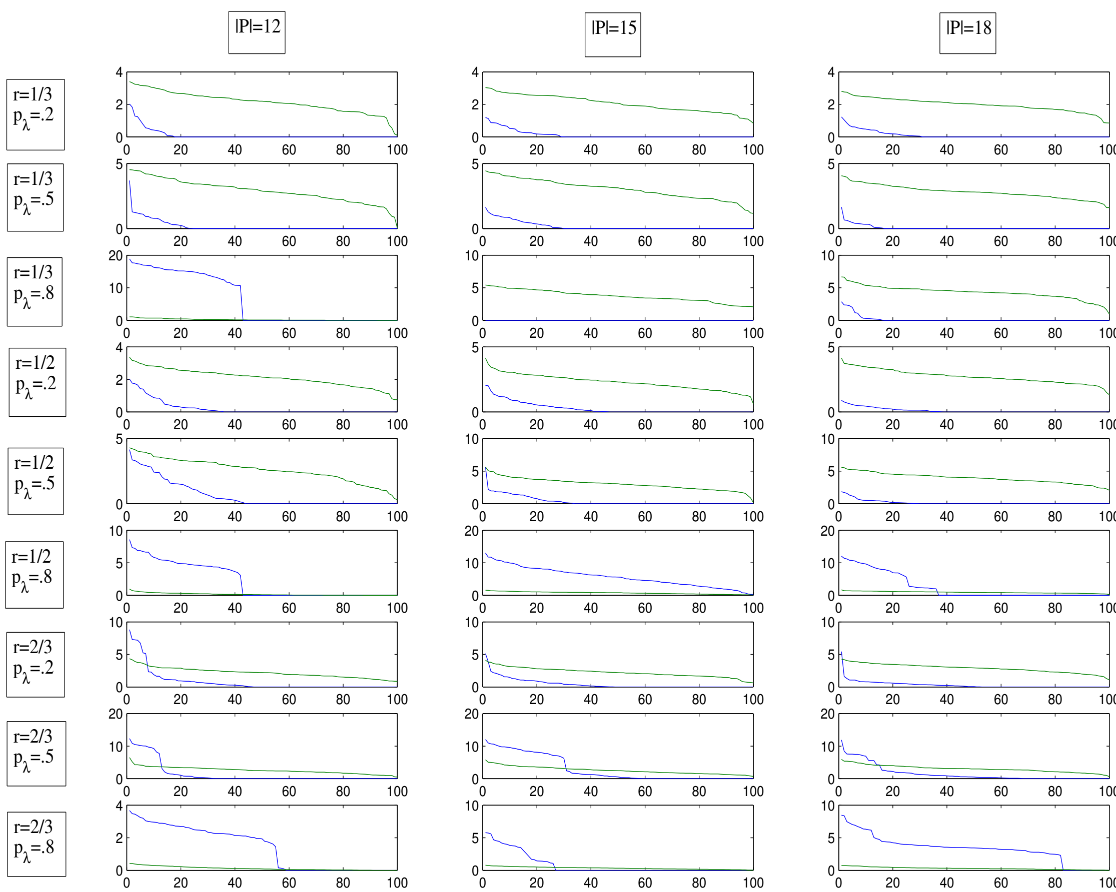

Figure 9. The sorted ACI and ACD for 100 replications

φ_i:

3 Heuristic G_2

Before explaining the heuristic G_2, we represent the parameters used in the remainder of this chapter as following.

Notation:

1. Rack information:

   W: Slot width

   L: Slot Depth

   H: Slot height

   O: Volume of slot $(= WLH)$

   N^{SL}: Number of slots per shelf

   W^{SH}: Shelf width

   N^{SH}: Number of shelves per bay

   N^{B}: Number of bays

   S: Total number of slots

   V: The volume of forward area $(= N^{B}N^{SH}W^{SH}LH)$

2. SKU information:

   w_i: Case width for SKU i

   l_i: Case length for SKU i

   h_i: Case height for SKU i

   o_i: Volume of carton of SKU i $(= w_i l_i h_i)$

   b_i: Eaches per case for SKU i

   d_i: Demand for SKU i per year

   p_i: No. of picks for SKU i per year

   f_i: Flow of SKU i in ft^3 per year

   φ_i: Maximum possible stack for SKU i in slot $(= \lceil \frac{H}{h_i} \rceil)$.

θ_i: Maximum possible No. of cartons of SKU i in depth of slot $(= \lceil \frac{L}{l_i} \rceil)$.

Input data: Rack information, SKU information, as above.

Result: Optimally slotted SKUs into the carton flow rack (determines which SKUs should be stored in the forward area and number of slots given to SKU i, n_i, in the forward area).

Two important questions come up in discrete assignment-allocation problem with greedy algorithm perspective:

1. How to rank the SKUs in the discrete problem?
2. How many slots are given to the set of $A \subseteq P$ SKUs selected for the forward area?

The answer of these two questions in the continuous fluid model were addressed before. However, we need to apply a different approach for the discrete problem because of SKU and slot dimensions considerations and the resulted lost space.

As previously defined, the flow rate of SKU i (f_i) is the demand of SKU i per year translated to the volume per year. We need to revise the concept of *flow* in the discrete version of forward reserve problem to account for unavoidable wasted empty space due to the case(s) not completely occupying the slot(s) (see Figure 10). The practical flow f_i^p is:

f_i^p= Number of slots required for SKU i per year $\times$ slot volume

Therefore:

$$f_i^p = \left\lceil \frac{f_i}{(\varphi_i h_i)(\theta_i l_i)W} \right\rceil O \tag{27}$$

$$= \left\lceil \frac{\frac{d_i}{b_i} o_i}{(\varphi_i h_i)(\theta_i l_i)W} \right\rceil O \tag{28}$$

$$= \left\lceil \frac{\frac{d_i}{b_i} w_i l_i h_i}{(\varphi_i h_i)(\theta_i l_i)W} \right\rceil O \tag{29}$$

$$= \left\lceil \frac{\frac{d_i}{b_i} w_i}{\varphi_i \theta_i W} \right\rceil O \tag{30}$$

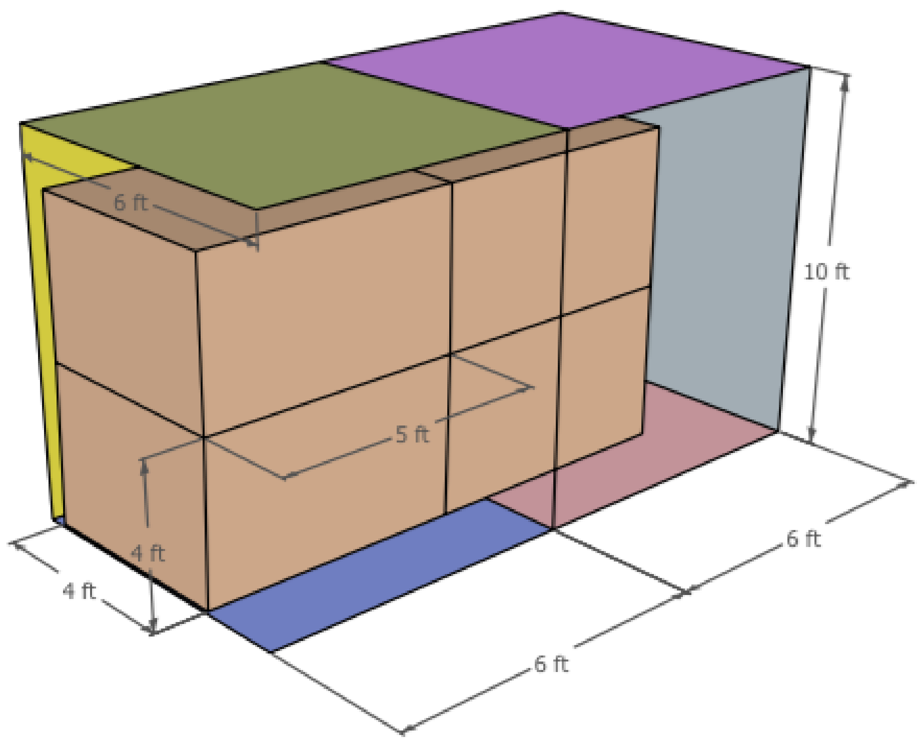

Figure 10. The unavoidable wasted empty space due to the difference between the cases and the allocated slots dimensions.

Example. Assume that the demand of SKU i per year is 320 units (eaches) and each case (carton) of SKU i has the capacity of 80 eaches ($d_i = 320$ and $b_i = 80$).

The dimension of the case for SKU i is:

$w_i = 5$ ft, $l_i = 4$ ft, $h_i = 4$ ft.

$o_i = w_i l_i h_i = 80 ft^3$

In the continuous model, where the slot dimensions are ignored, the flow of SKU i is equal to:

$f_i = \frac{d_i}{b_i} o_i = \frac{320}{80} 80 = 320 ft^3/year$

However, the slots' dimensions are considered in discrete model. Assume:

$W = 6$ ft, $L = 6$ ft, $H = 10$ ft

$O = WLH = 360$

Practical flow f_i^p in discrete model is:

$$f_i^p = \left\lceil \frac{\frac{d_i}{b_i} w_i}{\varphi \theta W} \right\rceil O \tag{31}$$

$$= \left\lceil \frac{(\frac{320}{80})5}{(2)(1)(6)} \right\rceil 360 \tag{32}$$

$$= 720 ft^3/year \tag{33}$$

The difference between f_i and f_i^p (400 ft^3), is the volume of empty space around the cases stored in two slots of the forward area as shown in Figure 10. If we generalize this wasted space to all slots in forward area, the amount of lost space by discretizing the problem is non-trivial. The heuristic will inherently tend to reduce the lost space as much as possible. Consequently, we introduce parameter e in our heuristics in order not to exceed the capacity of the forward area and generate feasible solutions. As mentioned, only a fraction of the forward area space can be practically allocated to SKUs and the rest is wasted. This fraction depends on the selected set of SKUs for the forward area. So we search the best solution by examining different amounts of e in the range $0 < e \leq 1$. Finally, the best coefficient of space is found.

We develop four procedures for ranking and fraction searching. y_i in heuristic

G_2 is used for calculating the number of slots allocated to SKU i n_i and is corresponded to the optimal space allocation in fluid model. For the first two procedures, where the SKU flows f_is are cubic feet per year, y_is are equal to the optimal cubic space given to SKU i in fluid model. However, the y_is in the last two procedures are based on case movements per year. q'_i is the aggregate number of restocks of SKU i for the planning horizon period, if only a single slot (or minimum number of feasible slots for $w_i > W$) is allocated to SKU i ($q'_i = \frac{d_i}{b_i a_i}$.) a_i is the units of SKU i that can be stored in minimum number of feasible slots allocated to the SKU and is defined as:

$$a_i = \begin{cases} \theta\varphi\lfloor\frac{W}{w_i}\rfloor & \text{if } w_i \leq W \\ \theta\varphi & \text{otherwise} \end{cases}$$

The procedures are:

$$\begin{array}{llll}
A_1: & f_i^1 = \frac{d_i}{b_i} o_i & le_{1i} = \frac{p_i}{\sqrt{f_{1i}}} & y_i^1 = \frac{\sqrt{f_{1i}}}{\sum_{j \in A} \sqrt{f_{1j}}} S \\
A_2: & f_i^2 = \frac{d_i}{b_i} o_i & le_{2i} = \frac{p_i}{\sqrt{f_i^p}} & y_i^2 = \frac{\sqrt{f_{2i}}}{\sum_{j \in A} \sqrt{f_{2j}}} S \\
A_3: & f_i^3 = \frac{d_i}{b_i} & le_i = \frac{p_i}{\sqrt{f_{3i}}} & y_i^3 = \frac{\sqrt{f_{3i}}}{\sum_{j \in A} \sqrt{f_{3j}}} S \\
A_4: & f_i^4 = \frac{d_i}{b_i} & le_i = \frac{p_i}{\sqrt{f_{4i}}} & y_i^4 = \frac{\sqrt{q'_i}}{\sum_{j \in A} \sqrt{q'_j}} S
\end{array}$$

While A_1 and A_2 rank the SKUs based on cubic feet movement of SKU, A_3 and A_4 use the number of cases needed during the planning horizon, instead of volume, for ranking SKUs. Note that the fraction given to SKU i in A_4 corresponds to parameter q'_i, not f_i. Of the four procedures, only A_2 uses the practical flow f_i^p for labor efficiency computation.

Heuristic G_2 is a greedy algorithm based on rounding up the continuous model solution. After discretizing the non-integral solution, it applies a post processing step, called *bottom-up deletion*, for removing undesirable SKUs from the forward area. After assignment of SKUs and allocation of slots, it sorts the SKUs in order of non-increasing total number of restocks. Then, it deletes the SKU with minimum number of restocks (say 1 restock), and allocates its space to the SKU in the forward area with maximum number of restocks. We call this method bottom-up deletion, since the bottom SKU in number of restocks ranking will be deleted and its slot is added to the upper SKU in the ranking. We iterate this procedure until achieving no cost improvements. Heuristic G_2 is as follow:

1. Rank all SKUs in order of non-increasing le_i (different definitions for le_i will be discussed)

for i=1:No. of SKUs(N) **do**

2. Define

$$n_{1i} = \left\lceil \frac{w_i}{W} \right\rceil \qquad n_{2i} = \left\lceil \frac{ey_i}{O} \right\rceil \qquad n_i = max(n_{1i}, n_{2i})$$

3. Get the assignment and allocation solutions and number of restocks (r_i)

end for

4. Rank all elements of SKUs assignment solution given by step (4) in order of non-increasing r_i.

5. Apply bottom-up deletion approach:

for j=1:No. of selected SKUs for forward area **do**

7. Remove the SKU with minimum r_i and add its slot to the SKU with maximum r_i

end for

8. Repeat steps (1) to (7) for different $0 < e \leq 1$ with interval .1 to choose the one gives the minimum cost.

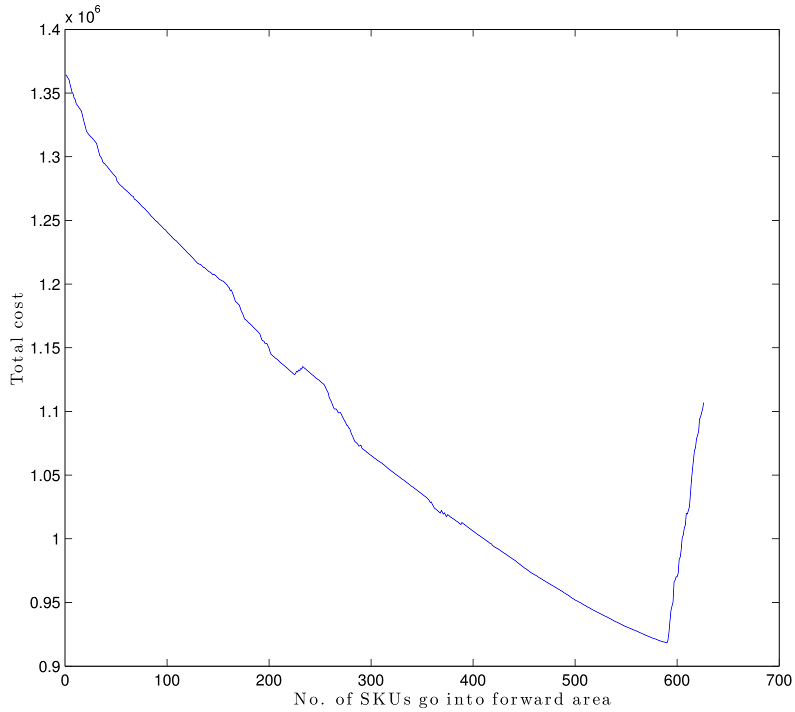

Figure 11. The cost reduction representation with increasing the number of SKUs in forward area.

We consider a warehouse with 700 SKUs to determine the assignment, allocation, and size of the forward area. The SKUs' dimension data belongs to a real world warehouse. The best size of the forward area as suggested by G_2 is 626 slots with 590 SKUs (see Figure 11). The minimum cost in Figure 11 occurs when to start adding those SKUs to the forward area that could be picked more efficiently from the reserve area. However, we have only 400 slots available in the forward area.

Before applying the bottom-up deletion approach, the set of 374 SKUs leads

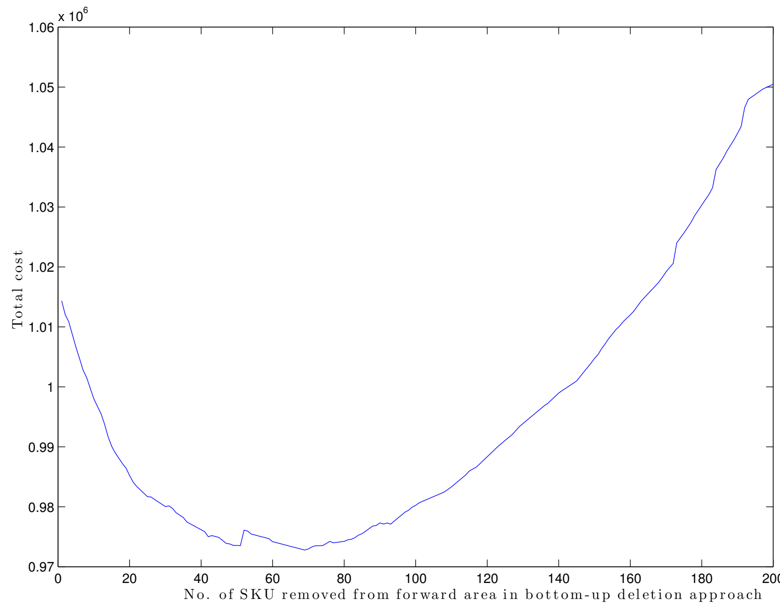

Figure 12. The bottom-up deletion approach.

to minimum cost of picking and replenishment. As Figure 12 shows, the bottom-up deletion approach in heuristic G_2 recommends deleting 69 out of 374 SKUs from the forward area to devote their slots to those SKUs of the forward area with a higher number of restocks. The bottom-up deletion approach motivates having the uniform number of restocks among SKUs because the SKUs, which have had high number of restocks, no longer be replenished very frequently. Using this approach, they have more slots and more cases in the forward area because of allocating the slots of the deleted SKUs to them. The cost increment in iteration 52 of Figure 12 is associated with the situation, where the bottom-up deletion approach deletes one SKU with one slot from the forward area, but the candidate SKU for this slot from top of the list, needs more than one slot to be able to have one more lane in the forward area (wider SKU than slot width.) Therefore, we have one deleted SKU from the forward area without any value added and the total cost slightly increases. In our example, this approach reduces the total cost by 4.4%.

D Comparisons

We apply a data set from a telecommunications provider warehouse studied in Bartholdi and Hackman (2008) to compare the procedures. The warehouse that we addressed in our numerical example has 3049 SKUs, 30 bays, each having four shelves. In this warehouse, each shelf has eight equal standard size slots and therefore 960 slots. The algorithm suggested the best coefficient of space for this warehouse $e = 0.17$.

Table 4 compares the costs of G_2 for different procedures A_1 to A_4. The costs of A_1, A_3 and A_4 are very close. However, A_4 outperforms others in this example.

TABLE 4

Cost comparisons between the procedures A_1 to A_4 using heuristic G_2

	A_1	A_2	A_3	A_4
No. of SKUs in forward area	805	783	721	722
Total cost	16237323	17035435	16190930	16182930
Solution time (seconds)	2.321091	1.903433	2.083014	2.123233
Cost Imp. of G_2-A_4 (%)	0.334988	5.004304	0.049408	-

Using the same data set, Figure 13 shows the total cost of picking and restocking for heuristic G_2 using A_1 to A_4. In this figure, the horizontal axis shows iteration i, when we add an SKU into the forward area in each step. We avoid naming this axis "Number of SKUs that go into the forward area," since some SKUs selected for the forward area were removed in bottom-up deletion approach in heuristic G_2. Figure 13 confirms the result of table 4 regarding the minimum cost of G_2 using A_4 scenario.

The cost associated with $G_2 - A_4$ lies below the others.

Using Figure 13, we can find the iteration that leads to the minimum cost, and then the best size of the forward area corresponding to that iteration can be obtained for each scenario. This figure provides practitioners a hint to decide about the size of forward area.

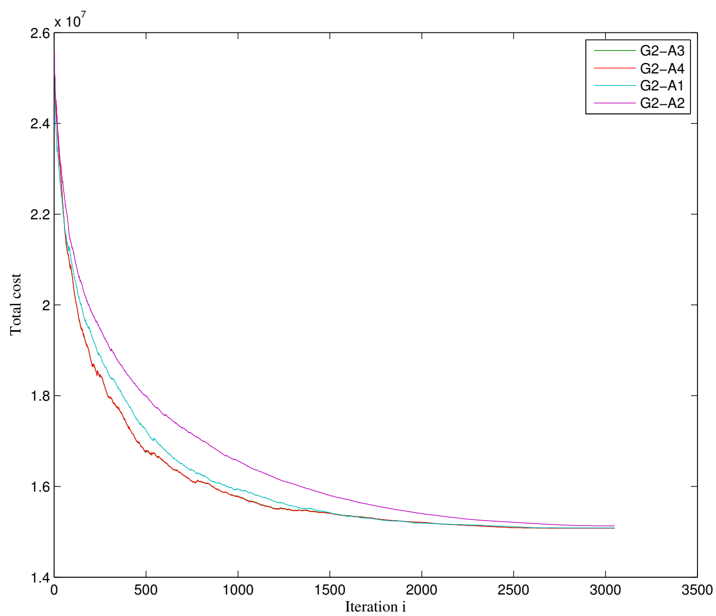

Figure 13. The cost comparisons of G_2 using different scenarios A_1 to A_4.

In a second numerical example, we apply heuristic $G_2 - A_1$ to a real data set to compare the capability of the offered heuristics in assignment, allocation, and sizing of forward area with an available online software (http://www.warehouse-science.com/) by Bartholdi. In this example, the warehouse has 6498 SKUs and the slots of the forward area are assumed to be identical with given dimensions. The SKUs' dimensions, demands, and picks data are given. The total cost is:

$$\sum_{i \in P} \sum_{j=1}^{n_i} (c_{1i} p_i + c_i \frac{d_i}{a_{ij}}) x_{ij} + \sum_{i \in P} (c_{2i} p_i) x_{i0}. \tag{34}$$

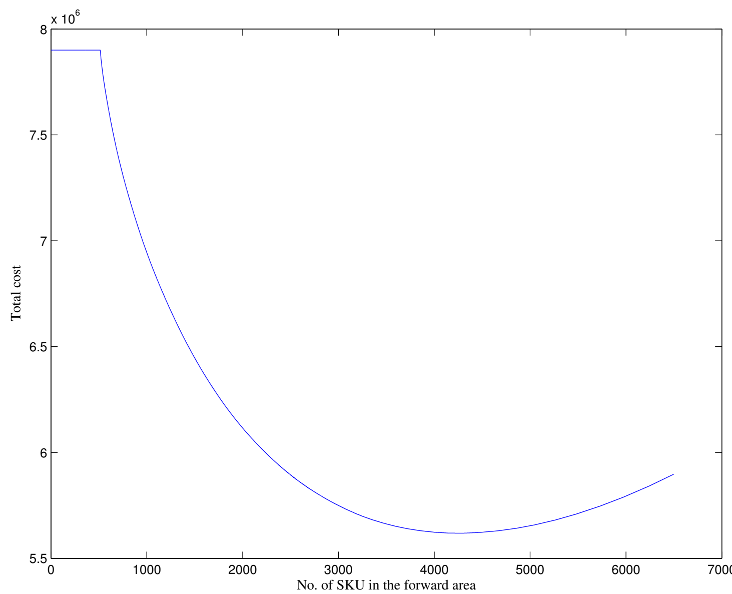

Figure 14. Heuristic $G_2 - A_1$ for real world data set with 6498 SKUs.

TABLE 5

Comparison of two methodologies for the discrete forward-reserve problem

	No. of SKUs in FW area	No. of used slots	Cost	No. of replens.
$G_2 - A_1$	4294	4318	5618853	3779

As observed in Figure 14 and Table 5, the optimal size of the forward area is 4318 slots. Giving more slots to the SKUs selected for the forward area than their allocated slots does not reduce their number of replenishment. So, the algorithm suggests to leave some slots unfilled.

As Figure 14 shows, the total cost starts to increase after the minimum point of the curve because the algorithm assigns the SKUs with the low rank labor efficiency to the forward area. These SKUs can be picked more efficiently from the reserve area. According to the results, if we assign more than 4294 SKUs to the forward area, the total cost will grow due to the increase of replenishments. The solution time for $G_2 - A_1$ is 2.83 seconds.

E Conclusion

Unlike previous studies conducted on the forward reserve problem, which consider the continuous space of the forward area, we addressed the assigning and allocation of the discrete units of SKUs to their discrete slots. The gap between the continuous space model and its discrete counterpart has been quantified for different test problems. Our numerical results showed that the cost difference between the heuristic and its continuous counterpart decreases for larger size forward areas.

First, we investigated the assignment, allocation, and the sizing problems simultaneously. We determined the optimal size of the forward area while deciding the optimal set of SKUs for the forward area. Second, the proposed heuristics delivered a solution in seconds for the large size discrete forward-reserve problem with thousands of SKUs. Finally, we relaxed the assumption of assigning the SKUs to the shelves versus the slots, since this hypothesis limits the problem to the condition whereby the width of the opening is always greater than the SKU width.

In all of the experiments, the cost comparison results showed that G_1 always outperforms $DFRAAP_{ES} - R_4$ in terms of the total cost of picking and replenishment. This cost improvement will become greater for larger size problems.

While G_1 works for the situations where the case and slot dimensions are not available, G_2 considers these dimensions. The last two procedures for ranking the SKUs in the greedy algorithm, A_3 and A_4, which involve the discrete f_i, outperform the first two procedures, A_1 and A_2, which contain a portion of the forward area's continuous space, o_i. Procedure A_4 outperforms other procedures. It allocates the slots based on the aggregate number of restocks, if only a single (or minimum feasible number of slots) is allocated.

CHAPTER III

THE AREA SIZING AND SLOTTING OF A MULTI-MODE FORWARD AREA

The picking and replenishment costs of the SKUs selected for the forward area can be reduced in multi-mode forward areas. While the number of cases stored in some pick modes can be more flexible, others have a fixed storage capacity. For example, the number of slots given to the SKUs in the carton flow rack are optimized by the allocation problem. The allocated slots can be one or multiple slots. However, the SKUs selected to be picked from the pallet flow rack are stored by a definite number of cases in the pallet. Different pick modes result in different number of replenishments for the SKUs selected for the forward area.

Selecting the best types of racks (e.g. pallet flow rack, carton flow rack, bin shelving, etc.) and their effective size along with the best assignment of the SKUs to the pick mode within the forward area, considerably affects the total picking and replenishment cost. For instance, although pallet flow racks can hold many cases of an SKU on one pallet, a small quantity of that SKU can fit in one bin of bin shelving. As a result, the former has lower replenishment costs, but higher picking costs because of the lower pick density of items. On the other hand, the latter has lower picking cost inside the bay, but higher number of replenishments because of the smaller allocated space. After finding the best SKU assignment and slot allocation for each pick mode, the best size of the mode and so the overall size of the forward area are found.

The classic Forward-Reserve Problem selects the best set of SKUs for the fast picking area of the warehouse and allocates the best number of slots to them having the size of the forward area. However, we will address the problem of determining the best number of bays/slots for each pick mode (e.g. pallet flow rack, carton flow rack, bin shelving, etc.) within the forward area, and we do so while determining the best SKU assignment and slot allocation. Considering an available space, we develop an algorithm, namely Profiling and Slotting Optimization (PSO) algorithm, which can increment number of bays of each pick mode, until adding more bays in the forward area increases the travel distance and costs (see Appendix.)

Although expanding the forward area decreases the total number of replenishments, the large forward area has larger fixed picking and replenishment costs because of larger travel distance. Determining the best size of each pick mode, we calculate the cost of every possible combination of bays quantities corresponding to each pick mode, while not exceeding the available space. In each iteration, the best SKU assignment and slot allocation are found as well.

In this chapter, one iteration of the PSO algorithm refers to generating one alternative for the forward area. The alternatives differ in their number of bays of each pick mode. The average travel distance for picking or replenishing of the items depends on the size of the pick mode. The average travel distance of a pick mode refers to the average horizontal distance that the labor traverses to pick or replenish an item (average aisle width) plus the average vertical distance (average aisle length). Therefore, our model accounts for the different picking and replenishment costs between the pick modes with different sizes within an alternative and also between the same pick mode of different alternatives. The sequence of the pick modes within the forward area is assumed known and is taken into consideration while calculating the average travel distance. Therefore, the farther pick modes to the Input/Output point have higher average pick/replenishment travel distance(See Figure 15.)

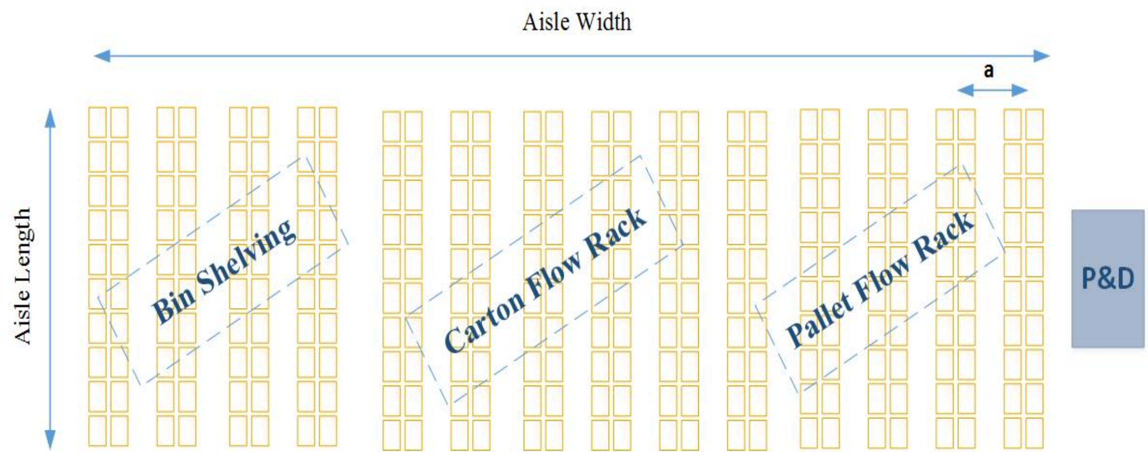

Figure 15. The multi-mode forward area

Besides the travel distance analysis, SKU (each), case and slot dimensions are taken into consideration in the proposed model to conduct the fitting test for the replenishment unit and the slot. Furthermore, the model suggests the best orientation of replenishment unit (case), which enhances space utilization. The space allocation to the SKUs is discrete, as opposed to the traditional method of allocating continuous space to the SKUs.

Jernigan (2004) develops a multi-tier inventory system. A multi-tier inventory system is an extension of the forward-reserve system with multiple forward modes. As opposed to our problem, which assumes all pick modes are replenished from the bulk storage, Jernigan (2004) establishes intermediate modes between the forward modes and the reserve area to reduce the restocking cost. The slotting of the inventory system in her study refers to finding the SKU assignment of the forward modes and the space allotted to the SKUs there, aiming to reduce the total picking and replenishment costs. While the volumes of the storage modes are known in Jernigan (2004)'s study, we optimize the size of the pick modes coupled with the slotting of the modes. Her study is also based on continuous space allocation, while we allocate discrete cases to the discrete number of slots slots, considering SKU and slot dimensions.

The PSO algorithm, which can be found in appendix, accounts for:

1. **Discrete case quantity movement**: The small size fast movers have a low cubic feet movements. The continuous space allocation has the risk of assigning the small size fast movers to the high cost areas within the forward area and large size slow movers to the low cost areas within the forward area. Algorithm PSO is based on discrete quantity of moves for each SKU.

2. **Discrete space allocation**: PSO is a discrete space allocation model, which considers the lost spaces resulted from differences in slots and SKUs dimensions. Solving the allocation problem in a continuous space model causes many SKUs to have allocated space of less than the volume of one slot, which is impractical. Rounding up the solution and allocating one slot to these category leads to exceed the size of the forward area, which is infeasible. The PSO algorithm considers discrete space allocation, as opposed to the continuous space allocation of the picking areas. So, discrete units of the SKUs (case quantities) are stored in discrete units of slots. This concept prevents allocating a portion of a slot to the SKUs.

3. **Replenishment unit fit test**: If we do not check the dimensional fitness of the SKU to the slots of the pick modes, there is the risk of assigning the large size items in small size rack types. The PSO algorithm will consider both SKU and slot dimensions.

4. **Optimizing case orientation**: Finding the best case orientation in the rack and maximum feasible stack level leads to better space allocation and so more cases can be replenished in the slot. So, the number of replenishments will be reduced. SKU rotation allows the SKUs to have more options for being assigned to the different pick modes. The PSO suggests the best case orientation.

Notation:

1. Rack information:

j:	Pick modes, $j \in \{0, 1, 2, 3\}$
W_j:	Slot width in pick mode j
L_j:	Slot Depth in pick mode j
H_j:	Slot height in pick mode j
O_j:	Volume of slot in pick mode j $(= W_j L_j H_j)$
N_j^{SH}:	Number of shelves per bay in pick mode j
N_j^{SL}:	Number of slots per shelf in pick mode j
N_j^{B}:	Number of bays in pick mode j
W_j^{SH}:	Shelf width in pick mode j
V_j:	The volume of the pick mode j $(= N_j^B N_j^{SH} W_j^{SH} L_j H_j)$

2. SKU information:

w_i^{sku}:	Width of SKU i
l_i^{sku}:	Length of SKU i
h_i^{sku}:	Height of SKU i
w_i:	Case width for SKU i
l_i:	Case length for SKU i
h_i:	Case height for SKU i
o_i:	Volume of case containing SKU i $(= w_i l_i h_i)$
b_i:	Eaches per case for SKU i

L_i: Find the number of lanes for SKU i in carton flow rack

θ_{ij}: Maximum No. of cases of SKU i in depth of slot in mode $j (= \lceil \frac{L_j}{l_i} \rceil)$.

φ_{ij}: Maximum calculated stack level for SKU i in slot of mode j $(= min(\rho_i, \lceil \frac{H_j}{h_i} \rceil)$.

ρ_i: Max given stack level of SKU i

3. Costs notation:

c_j^p: Average picking cost from pick mode j.

c_j^r: Average replenishment cost of pick mode $j > 0$ ($c_{i0}^r = 0$.)

r_{ij}: Number of replenishments if SKU i is assigned to mode $j > 0$ ($r_{i0} = 0$.)

C_{ij}^p: Picking cost if SKU i is assigned to mode j.

C_{ij}^r: Replenishment cost if SKU i is assigned to mode $j > 0$ ($C_{i0}^r = 0$.)

C_{ij}^T: Total picking and replenishment costs, if SKU i is picked from the mode j.

s_{ij}: Saving of SKU i, if it is picked from mode $j > 0$ rather than mode 0.

The steps of the proposed algorithm for Profiling and Slotting Optimization of the forward area are as follow. The algorithm has been coded with the Python programming language.

1. **Import data**: Four types of data are imported: SKU data, order data, rack data and facility data. The SKU data provides the information about the case and SKU (each) dimension. The order file contains the historical demand data. The rack information delivers the setting of the racks in different types of pick modes (see table 10). The available space for designing the forward area and number of bays in the reserve area are provided by the facility data to estimate the fixed picking and replenishment costs. Table 7 summarizes the inputs parameters.

TABLE 7

Inputs

SKU Data	Order Data	Rack Data	Facility Data
SKU Number	Time	Width of a shelf	Horizontal length of the forward area
SKU Length	SKU Number	Depth of a shelf	Vertical length of the forward area
SKU Width	Order quantity	Height of a shelf	No. of bays in one aisle of the reserve area
SKU Height		No. of level	No. of aisles in reserve area
Case Length		No. of Slots per Shelf	
Case Width			
Case Height			
Case Pack			
Max Stack Level			

We assume that the pallet flow rack, which is generally the best option for the fast movers, is closest zone to the Pick up and Deposit (P&D) point. Next, the carton flow rack and the bin shelving zones are designed (see Figure 15). We also consider three types of pick mode in our example warehouse.

2. **Fit test**: Based on SKU and slots dimensions, the SKUs that are not fitted to a particular type of slot will not be assigned to that rack type.

3. **Case orientation**: For all SKUs, the best case orientation in each pick mode that gives the maximum space utilization is determined in this section of the PSO algorithm.

4. **Finding other parameters**: The SKU demand d_i and picks p_i, which is the sum of order quantity and the order lines per SKU during the planning horizon, respectively, and the SKU flow f_i or the number of cases of SKU i during the planning horizon, are found in this step. f_i can be found as:

$$f_i = \frac{d_i}{b_i} \tag{35}$$

We define a_{ij} as the number of cases of SKU i that can be stored in one pallet of a pallet flow rack ($j = 1$), the number of cases can be stored in minimum number of feasible slots in a carton flow rack (j=2), or one for bin shelving ($j = 3$). Assuming that the case pack quantity can fit in one bin of bin shelving, a_{ij} is obtained from:

$$a_{ij} = \begin{cases} \theta_{ij}\varphi_{ij}\left\lfloor \frac{W_j}{w_i} \right\rfloor, & \text{if} \quad w_i \leq W_j \\ \theta_{ij}\varphi_{ij}, & otherwise. \end{cases}$$

For the pallet flow rack, a_{ij} is calculated from the first equation of a_{ij}, because the case width is always less than the pallet width. For bin shelving, a_{ij} is one case.

TABLE 8

6 possible case orientations in slot

Orientation	W_j	L_j	H_j
1	w_{ij}	l_{ij}	h_{ij}
2	w_{ij}	h_ij	l_{ij}
3	l_{ij}	h_{ij}	w_{ij}
4	l_{ij}	w_{ij}	h_{ij}
5	h_{ij}	l_{ij}	w_{ij}
6	h_{ij}	w_{ij}	l_{ij}

For each SKU, we check 6 possible case orientations to find the best orientation of the case in pick mode j that gives the maximum a_{ij} (see Table 8.)

We define q'_{ij} as the number of replenishments if the minimum number of slot(s) of mode j is allocated to the SKU and is calculated as below:

$$q'_{ij} = \frac{f_i}{a_{ij}} \tag{36}$$

5. **Cost analysis**: In this section, for every possible combination of the number of bays in each pallet flow rack, carton flow rack and bin shelving, the savings of picking from these areas versus picking from the reserve area are calculated for all SKUs. The slow movers with negative savings, however, are picked from the reserve area. The reserve area, pallet flow rack, carton flow rack, and bin shelving have the modes 0, 1, 2, and 3, respectively, in our example.

   To calculate the average picking and replenishment costs, we assume that the maximum horizontal length and vertical length of the available space for the forward area are given. The total costs and savings of an SKU by picking from the mode j are calculated as:

$$r_{ij} = \left\lceil \frac{d_i}{m_{ij}} \right\rceil \tag{37}$$

$$C^T_{ij} = C^p_{ij} + C^r_{ij} \tag{38}$$

$$= c^p_j p_i + c^r_j r_{ij} \tag{39}$$

$$s_{ij} = C^T_{i0} - C^T_{i,j>0}, \tag{40}$$

   where r_{ij} is the number of replenishment of SKU i in mode j. m_{ij} is the units (eaches) of SKU i in mode j. The algorithm for finding m_{ij} can be found in Appendix.

6. **Export file**: The algorithm will be terminated by exporting the outputs.

We have implemented the PSO algorithm for an example warehouse with 28 SKUs. The rack information data can be found in Table 10. Using PSO algorithm, the optimal number of bays of pallet flow racks, carton flow racks and bin shelving, are

TABLE 9

Outputs

Slotting and Cost Data		Size of each pick mode
SKU No. Pick mode Adjusted Width Adjusted Length Adjusted Height Each fit CF? Case fit CF? Each fit PF? Case fit PF? Each fit SH? Case fit SH?	Slots in CF Cases in CF Lanes in CF Stacks in CF Depth in CF Ordered QTY Order Lines SKU Flow Saving Picking Cost Replenishment Cost Total Cost	No. bays of carton flow rack No. bays of pallet flow rack No. of bays of bin shelving

PF = Pallet flow rack
CF = Carton flow rack
SH = Bin shelving

3,3,1, respectively ($[N_1^B, N_2^B, N_3^B]$=[3,3,1].) If we expand the forward area more, the picking and replenishment costs will increase. The algorithm then suggests the best number of bays of each pick mode, along with the maximum total savings.

TABLE 10

Rack information

Carton Flow Rack (CF)		**Pallet Flow Rack (PF)**		**Bin Shelving (SH)**	
Width of a shelf	96	Width of a shelf	96	Width of a shelf	48
Depth of a shelf	96	Depth of a shelf	48	Depth of a shelf	21
Height of a shelf	20	Height of a shelf	60	Height of a shelf	12
Number of levels	4	Number of levels	1	Number of levels	5
Number of slots per Shelf	6	Number of slots per Shelf	2	Number of bins per shelf	4

1 Extensions

The PSO algorithm (see Appendix) is a general model for sizing and slotting of the forward area's zones. Considering the broad spectrum of requirements in real world warehouses, we evaluate the opportunities to empower the PSO algorithm in many ways. The goal is to explore and validate multiple pick mode design options that meet the distribution center's requirements. The main features that have been added to the model are as follows:

Volume growth and SKU growth

Two types of growth affect the decisions made about the forward area: SKU growth and units growth. They should be considered while planning for the size of the pick modes. The units growth is the total growth in shipping units during the planning horizon, which includes the SKU growth as well. If no new SKU is introduced to the warehouse during the planning horizon (e.g. next 2 years), meaning that the SKU growth is equal to zero, we can account for the units growth by multiplying the historical SKUs demand and pick data by the units growth in the forward area sizing problem. The effect of the SKU growth is zero in this case. Thus, the SKUs demand will grow by the units grow factor itself. However, when the SKU growth is not zero, the SKU growth should be extracted from the units growth to provide a net growth factor. This prevents an over-expansion of the forward area. Applying the units growth and the SKU growth separately on the demand data executes the SKU growth twice because the units growth already includes the SKU growth. We are looking for the net growth factor to apply to the demand data, which will address both growths simultaneously. The goal is to change the historical demand data to account for both units and SKU growths.

We define factor γ (%) as the net growth factor, which can address both types of growth. Thus, planning for the size of the pick modes will be based on the demand data, which has grown by γ. We have $\alpha = \gamma + 1$ in the following equations for simplification. Assume:

- a_1: Units growth during the planning horizon (%)×0.01.
- a_2: SKU growth during the planning horizon (%)×0.01.
- n_1: Number of SKUs before the SKU growth.
- n_2: Number of added SKUs after the SKU growth ($n_2 = a_2 n_1$.)
- p_i: Historical picks (lines) for SKU i before any growth, $i \in \{1, ..., n_1\}$.
- $\bar{p}$: Average picks (lines) of new SKUs based on the historical pick data.

$$\alpha(\sum_{i=1}^{n_1} p_i + n_2\bar{p}) = (1 + a_1)\sum_{i=1}^{n_1} p_i \tag{41}$$

The term $n_2\bar{p}_i$ in equation 41 represents the expected additional picks in the future, which considers the SKU growth. It is assumed that the prospective SKUs have the mean picks of the historical pick data. Then, we apply α to the expanded data set, containing both prospective and current SKUs. This would be equivalent to applying the $(1 + a_1)$, which inherently includes both units and SKU growths, to the picks of the current SKUs in the warehouse. Then, by setting $\bar{p} = \frac{\sum_{i=1}^{n_1} p_i}{n_1}$, we obtain:

$$\alpha(\sum_{i=1}^{n_1} p_i + n_2 \frac{\sum_{i=1}^{n_1} p_i}{n_1}) = (1 + a_1) \sum_{i=1}^{n_1} p_i \tag{42}$$

For simplicity, we assume: $\beta = \sum_{i=1}^{n_1} p_i$

$$\alpha(\beta + n_2 \frac{\beta}{n_1}) = (1 + a_1)\beta \tag{43}$$

$$\alpha\beta(1 + \frac{n_2}{n_1}) = (1 + a_1)\beta \tag{44}$$

$$\alpha(1 + \frac{n_2}{n_1}) = (1 + a_1) \tag{45}$$

$$\alpha(1 + \frac{a_2 n_1}{n_1}) = (1 + a_1) \tag{46}$$

$$\alpha = \frac{1 + a_1}{1 + a_2} \tag{47}$$

$$\gamma = \frac{1 + a_1}{1 + a_2} - 1 \tag{48}$$

$$\gamma = \frac{a_1 - a_2}{1 + a_2} \tag{49}$$

We can observe from equation 49 that the actual growth factor γ is not affected by the order data. We can also see if the SKU growth, a_2, is larger than the volume growth, a_1; the decreasing γ factor should be applied to the historical order data.

From a practical viewpoint, if a high number of new SKUs are introduced to the warehouse (high SKU growth) but the total shipping units do not considerably grow (low units growth), the historical order data is experiencing a downward trend with a decreasing γ factor. Likewise, if a low number of new SKUs are introduced to the warehouse (low SKU growth) but the total shipping units are growing considerably (high units growth), the historical order data is experiencing an upward trend with an increasing γ factor. If the SKU and units growth are the same, $a_1 = a_2$, it means that the new SKUs' orders are following the order history. No change in order data is required, however, when the γ factor is zero. After applying the γ factor to the historical order data, we expand the size of the forward area based on SKU growth, a_2, accordingly, to stay up-to-date with the projected growths.

Time window analysis

Among those SKUs that have been selected to be picked from the forward area, many items are active only during a specific period of the year and are occasionally ordered (Halloween products, Christmas products, etc.). These seasonal products are fast movers in their active period and inactive during the rest of the planning horizon. Those seasonal products that their active periods have no overlap during the planning horizon can share their slots in the forward area. Considering an individual space for the seasonal items in the forward area expand the forward area unnecessarily. Storage sharing of the seasonal items prevents over expansion of the forward area. Considering that seasonal fast movers are not always active during the planning horizon, the storage share idea provides the opportunity of having *active* fast movers in the forward area.

We extract the "start and end order dates" of each SKU from the order file. Therefore, the active periods of all SKUs, which is the period between these two dates, are achieved. Those active SKUs, which are picked from the similar pick mode zone and whose active periods do not have any overlap, will share one location in that zone. For example, the Christmas products will replace the Halloween products, when their active periods are over. This method shrinks the forward area as required and shortens the travel distance for the rest of the fast movers, which are active all year. Numerical example will quantify the decrease in number of slots by storage share in each pick mode.

Although it can be assumed that the SKUs have multiple active periods during the planning horizon, we consider a single active period per SKU, beginning with the first order date and ending with the last order date. The reason is the opportunity of replacing the SKU, whenever it gets inactive, is delivered by the dynamic slotting of the forward area.

Other extensions of the PSO algorithm

Other features that have been applied to the model are stated below:

- **Reorder point.** This percentage triggers the replenishment process, whenever the inventory on hand for the SKUs in the forward area is diminished up to a certain threshold. Different pick modes present different replenishment trigger thresholds, but larger values for this factor leads to more frequent replenishments.

- **Space utilization factor.** This factor determines the space efficiency of the pick modes. Higher space utilization refers to the higher density of products in the bay. Since the model considers the discrete replenishment units, there is always some lost space due to the difference between the dimensions of the slots and the cases. The space utilization factor of the bay in pick mode j is:

$$\frac{\text{Total volume of all cases in one bay of pick mode } j}{\text{Bay width} \times \text{Bay depth} \times \text{Bay height}} \tag{50}$$

- **Lock the dimension.** Although the PSO algorithm can deliver the optimal orientation of the cases in the slots, which results in the highest spaces utilization, some SKUs need to be locked by certain dimension(s). The lock dimension for each SKU is a binary input. For example, setting the "height case locked" to one SKU indicates that the slotted case orientation height must match the Case Height. Even an SKU is locked in one dimension, the orientation of other two dimensions is fixed.

2 Numerical example with a large data set

To test the pick mode model, we used real warehouse data with 6,000 SKUs. The historical order data contains 3,501,347 order lines over 391 days. The projected SKU and volume growth are 10% each. The replenishment trigger percentage is set to 10% based on their replenishment policies. The SKU and case dimensions are available and the dimensions are not locked. The rack types considered for the fast picking area are standard size pallet flow rack, carton flow rack and bin shelving (see Table 10).

Changing two parameters of aisle length and width, we designed six experiments with different sizes of the forward area's available space to study the pick mode model performance. The aisle width and length are increased in steps of $1000''$ and $500''$, respectively. Table 11 presents the relevant numerical results. $[N_1^B, N_2^B, N_3^B]$ and $[N_1^B, N_2^B, N_3^B]^*$ are the relevant solutions without and with the time window analysis. The time window analysis reduces the zones' size, where there is the opportunity of sharing the locations among the fast movers with no overlap in their active periods. Figure 16 shows that the picking and replenishment costs will no longer be reduced after experiment 3 and the optimal size of the forward area is $4000'' \times 2000''$.

The reason is that the travel cost of larger forward areas are greater. The solution time rises in extending the area.

TABLE 11

Experiments for pick mode analysis

No.	Aisle Width	Aisle Length	$[N_1^B, N_2^B, N_3^B]$	$[N_1^B, N_2^B, N_3^B]$*	Cost	Sol. Time
1	2000	1000	[10, 50, 160]	[10, 50, 159]	3064664986	587.16
2	3000	1500	[15, 165, 155]	[15, 165, 129]	1863577592	888.26
3	4000	2000	[20, 280, 82]	[20, 278,44]	1676469592	1890.83
4	5000	2500	[26, 286, 52]	[25, 284, 40]	1719429196	3485.69
5	6000	3000	[31, 310, 62]	[30, 306, 21]	1811408626	7045.21
6	7000	3500	[36, 324, 72]	[29, 319, 12]	1923196148	15686.47

Figure 17 compares the optimal number of bays of the carton flow rack and bin shelving in our six experiments. One insightful outcome of this figure is that larger forward areas to have more carton flow racks and smaller forward areas tend have larger bin shelving area. The allocated space to the SKUs in the bin shelving area is less than the carton flow rack area. In our example, one bin is given to each SKU in the bin shelving area, and each bin can contain one case, if the case and the bin dimensions fit. However, carton flow rack slots can provide more than one case per SKU because the carton flow rack's slots are larger than the bins in the bin shelving area. The bin shelving mode provides the opportunity of having the fast movers in small forward areas. As a result, the best storage mode of the SKUs selected for the forward area depends on the size of the available space for designing the forward area.

A Conclusion

We proposed the PSO algorithm for both profiling and slotting optimization. It determines the best size of a different pick mode in the forward area, along with

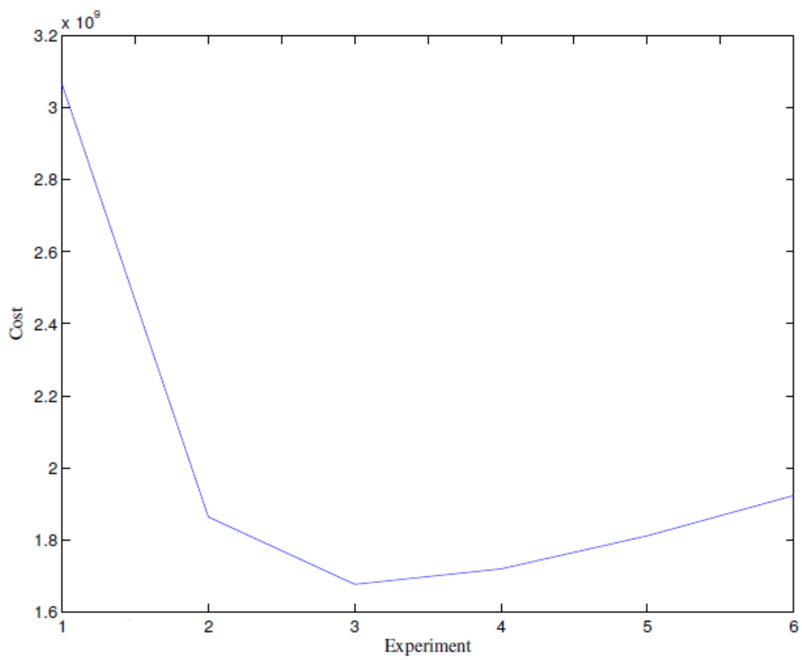

Figure 16. Pick mode cost for experiments 1 through 6

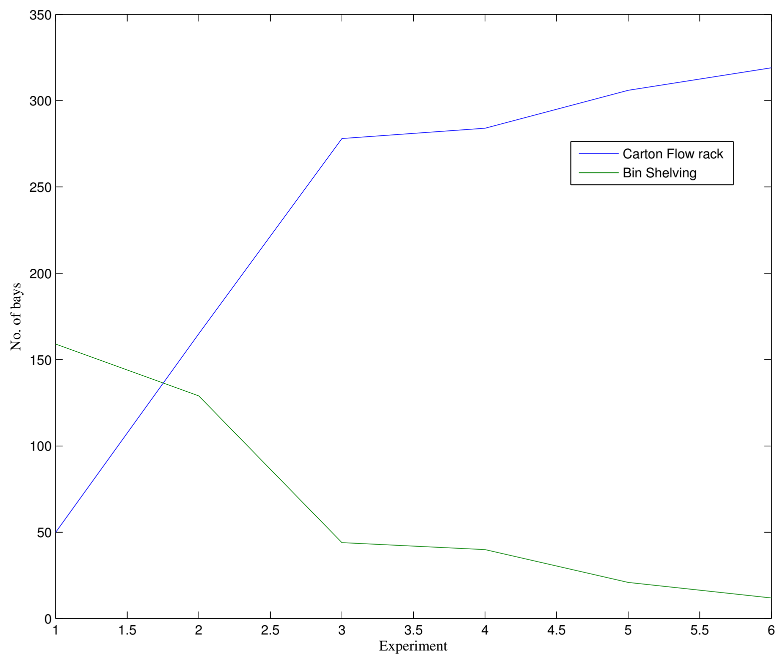

Figure 17. Carton flow rack and bin shelving comparison

the SKU assignment and slot allocation. PSO includes a replenishment unit fit test and implements the case orientation test for maximizing space utilization within a slot type. We showed how the time window analysis, which is based on sharing the storage in the forward area among the seasonal fast movers with different active periods, makes the size of each pick mode smaller, causing the overall size of the forward area to decrease. Using this idea, we can make the picking and replenishment travel distances shorter without giving the seasonal SKUs a smaller allocated space in the forward area. Finally, the results of our six experiments on a real warehouse data set showed that the larger available space for the forward area leads to having more carton flow racks than bin shelving settings.

CHAPTER IV

DEMAND FORECASTING

The traditional forward-reserve problem fails to consider the effect of significant changes in demand. In some cases, only the products with seasonal demand patterns are considered for the forward area. Consequently, important opportunities and costly threats may be missed. The static slotting optimization addresses a problem, when the decisions about the forward area are made periodically and ignores the historical demand trends of SKUs. We define dynamic slotting optimization as a methodology that uses information available during the planning period to affect storage and retrieval decisions (e.g. how to fill empty slots).

We address the problem of forecasting different types of demand trends in this section. In the first step, we recognize the type of demand trend using the Neural Network (NN). Next, the demand quantity is predicted using the appropriate method depending on the demand trend of the SKU. For example, some SKUs have an irregular demand pattern, for which traditional smoothing-based forecasting methods do not work. The SKUs with an intermittent demand have many zero values of demand during the planning horizon. So the method of forecasting them is different from the SKUs that are ordered frequently.

The selection of a method depends on, but is not limited to, the relevance and availability of the historical data, the desirable degree of accuracy, the time period to be forecasted, and the time available for making the analysis. Our purpose here is to present an approach for acquiring demand forecast data as an input for the dynamic forward-reserve problem. We will employ the Artificial Neural Network (ANN) technique as a tool for qualitative step and a time series data analysis for the quantitative step.

The rest of the chapter IV is organized as follows. In section A, we propose a new method based on ANN for forecasting different types of demand trends. After recognition of a demand trend type, the demand quantity is forecasted in section B.

A Qualitative model

The Control Chart Pattern Recognition (CCPR) technique, which is an effective tool in Statistical Process Control (SPC) for detecting process mean shifts, has been applied for the demand trend recognition. In Statistical Process Control, selected statistics are used to monitor processes for instability. The process is said to be "out of control" if the statistic falls outside of the defined control limits or follows a trend. Assume the demand of each SKU as the statistic plotted on the control chart. Figure 18 shows the Control Chart (CC), which monitors the demand statistics during the planning horizon.

The demand patterns considered in this research include normal, down trend, up trend, systematic, down shift, up shift, cyclic, and intermittent patterns. Previous studies on CCPR consider one pattern as normal and all other patterns are defined as different kinds of abnormality.

There are several methods for CCPR pattern recognition in the literature. Artificial Neural Network (ANN) is a common tool for classification problems and pattern recognition. Figure 19 illustrates a basic structure of a NN with three layers: input, hidden, and output. It contains artificial neurons and interconnections similar

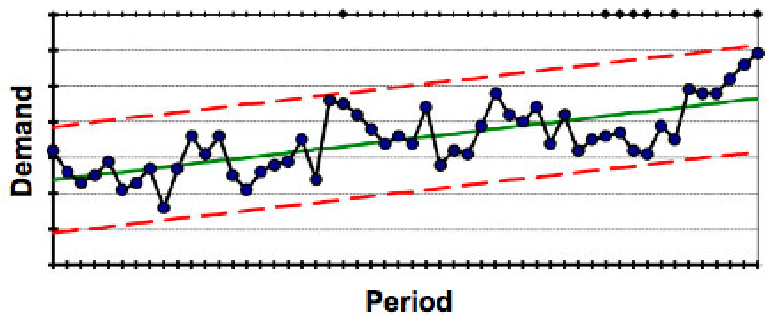

Figure 18. Demand trend shown by Control Chart

to the human brain. Each processing unit (neuron) receives and combines the input and then transforms them into a single output. The network connects an input layer to an output layer through hidden, or internal, nodes.

Two stopping criteria in modeling a neural network are defined. The hidden nodes are added until the further addition no longer reduces the forecast error or until the forecast error is within a defined tolerance level. Some advantages of NN are non-linearity, the capability of learning from instances, adaptivity, evidential response, fault tolerance, and the uniformity of analysis and design (Kantardzic, 2011).

Pattern recognition is defined as *the process whereby a received pattern is assigned to one of a prescribed number of classes* (Kantardzic, 2011). We perform pattern recognition through a learning process. The ANN first operates the training phase, during which the network receives a set of historical demand patterns along with the class of which each specific pattern belongs. Next, a new demand pattern is given to the network during the testing phase to identify the category of that

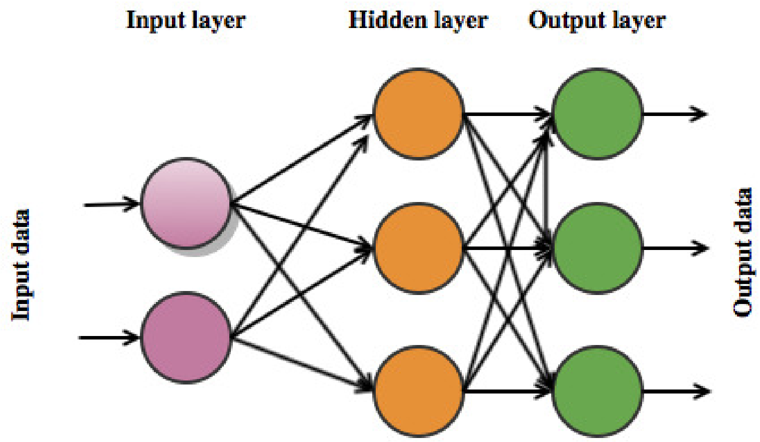

Figure 19. Basic structure of a NN

particular demand pattern.

Pattern recognition accuracy is designed as a performance measure in ANN-based approaches. Guh and Hsieh (1999) proposed an ANN that not only recognizes the abnormal pattern, but also estimates the abnormality parameters, such as trend slope and shift magnitude. Perry et al. (2001) report an ANN that automatically detects and corrects out-of-control states. Purintrapiban and Corley (2012) develop a NN-based model for autocorrelated processes. They state that all previous applications assume that the monitoring statistic is independent and identically distributed. Masood and Hassan (2012) present issues corresponding to input data representation, training, diagnosis, and recognizer design.

Recently, there has been a trend towards applying feature-based input representation techniques and hybrid recognition systems. According to Motoda and Liu (2002), feature selection is the process of choosing a subset of features, while feature extraction is the process of creating a new set of features. Pham and Wani (1997) use a feature extraction module on unprocessed data to raise the recognition accuracy of pattern shapes. Gauri and Chakraborty (2006) introduce eight new features to enhance ANN and recognizer performance. Hassan et al. (2003) demonstrates that a feature-based ANN pattern recognizer for SPC, gives significantly better results compared to a raw data-based recognizer. Ranaee and Ebrahimzadeh (2011) improve the classification performance of a proposed feature-based Support Vector Machine (SVM) by integrating this classifier with a Genetic Algorithm (GA) for SVM parameter optimization.

Based on mathematical models described in studies by Al-Assaf (2004), Gauri and Chakraborty (2006), Gauri and Chakraborty (2009), and Shao (2012), we simulated normal and abnormal patterns, as illustrated in Figure 20.

The mean of abnormal pattern, $a(t)$, consists of two important components of a constant term: μ and a particular abnormal function $d(t)$ that models a particular abnormal pattern. This term $d(t)$ is zero in the normal demand pattern. The mathematical model for the mean of simulated patterns can be expressed by the following:

$$a(t) = \mu + d(t) \tag{51}$$

In equation 51, $d(t)$ is defined as the following for different abnormal patterns:

1. Up/Down trends: $d(t) = \lambda t$, where λ is the trend slope in terms of σ_ε. The parameter $\lambda > 0$ is selected for up trends and $\lambda < 0$ for down trends.

2. Up/Down shifts: $d(t) = \gamma$, where parameter γ shows the shift magnitude. The parameter $\gamma > 0$ is selected for up shifts and $\gamma < 0$ for down shifts.

3. Cyclic pattern: $d(t) = \kappa(\frac{2\pi t}{\Omega})$, where κ is the amplitude of the cyclic patterns, and Ω is the cyclic pattern period.

4. Systematic trends: $d(t) = \nu(-1)^t$, where ν is the magnitude of systematic pattern.

In obtaining the demand patterns, we first generate a random number ρ_t from the normal distribution with mean $a(t)$ and standard deviation parameter σ at time t. Then, we apply the Exponentially Weighted Moving Average, EWMA, technique, where the demand at time t depends on the EWMA statistic, which is an exponentially weighted average of all prior demand data, including the most recent demand. We compute successive demand points Z_t using all preceding demand points and the weighting factor of Θ. The EWMA static is calculated as:

$$Z_t = \Theta a(t) + (1 - \Theta) Z_{t-1} \tag{52}$$

With respect to the broad spectrum of parameters levels in relevant studies, (Gauri and Chakraborty, 2009) and (Shao, 2012), a trial-and-error approach is taken in this research to improve the model's parameters.

The CCPR problem has been formulated into a classification problem with ANN. In the example shown in Figure 20, we generated random independently and identically normal and abnormal distributed samples with size $m = 30$ for different patterns during the observation window length $w = 20$.

We assume the cyclic pattern is influenced by the seasonal factors with fixed and known periods (e.g., the quarter of the year, the month, or day of the week). Therefore, the term cyclic pattern can be replaced by seasonal pattern. Visualization tools such as Figure 20 provide us with the trend cycle in cyclic and systematic trends.

After generating eight demand patterns, we scale the minimum and maximum values of each sample to [-1,1] for better training. Next, we divide the targets into

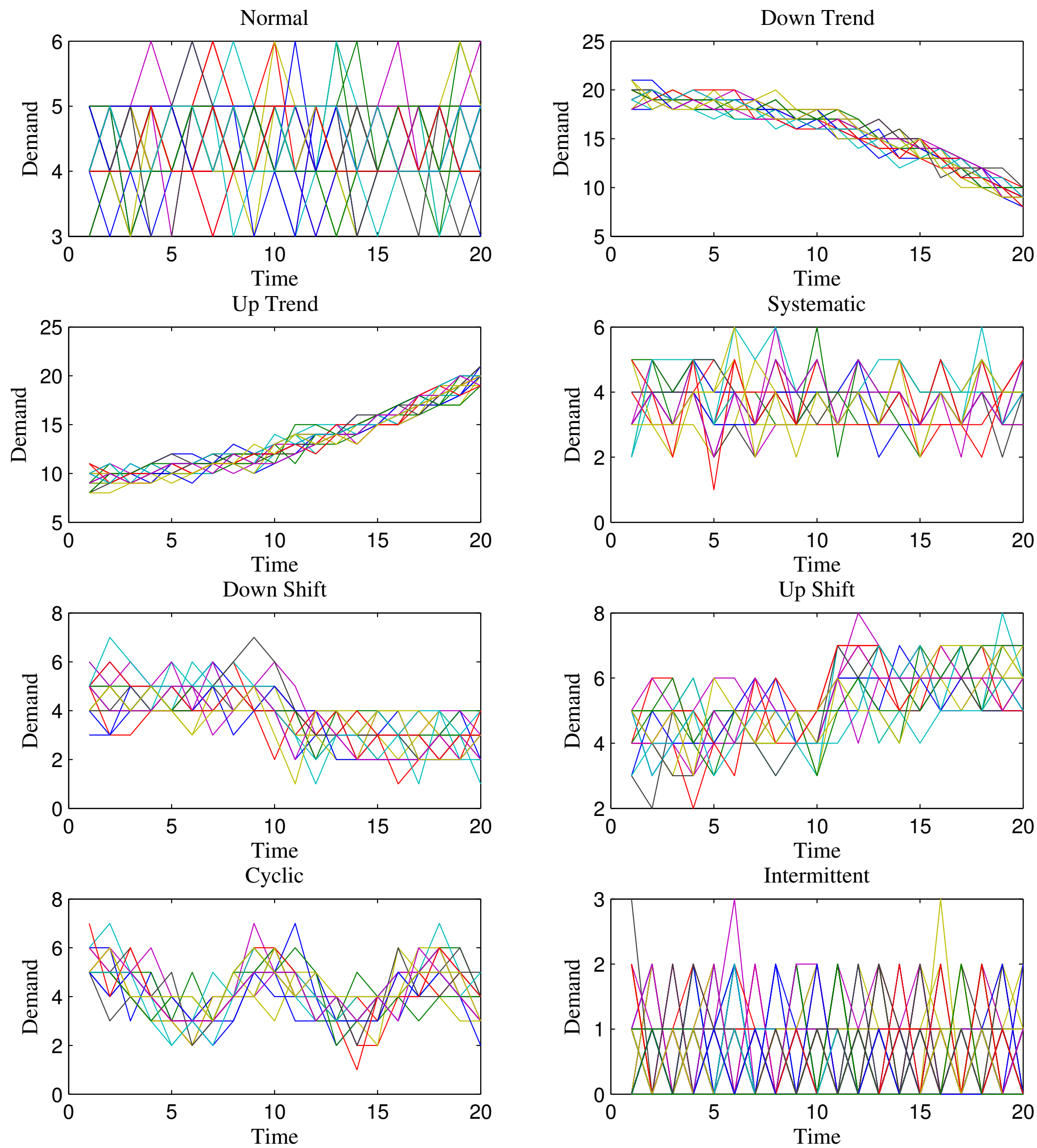

Figure 20. Example of different generated demand patterns.

three sets: training (70%), validation (15%), and testing (15%) sets for our network. The Levenberg-Marquardt back propagation algorithm suits our network for training process. The Logsig transfer function that calculates the NN layer's output from its net input is used. These choices of transfer function and training algorithm give the best classification accuracy by trial-and-error and are often used in the literature. The Mean Square Error (MSE) performance function with error weighting is used as the stopping criteria.

As the confusion matrix in Figure 21 shows, the performance of the ANN classifier for detecting demand pattern is 99.6% for all cases, which demonstrates the capability of NN for predicting the demand trends. In this figure, each row represents the instances in an actual class, while each column presents the instances in a predicted class (type of demand trend). The high accuracy shown in this matrix verifies the ANN's capability in recognizing demand trends. The Mean Square Error (MSE) of this classification problem is .00391 after 6 iterations with solution time of less than five seconds. The error histogram plot, shown in Figure 22, validates the small values of errors for the three phases. This small error confirms the quality and fitness of the ANN classifier for this pattern recognition problem.

B Model

It is not possible to perfectly forecast the future, but ignoring the forecast is very expensive. The predictive models will not tell us what will happen in the future. Instead, they determine what will probably happen with an acceptable level of error. Assessing demand trends using real-time order transaction data is an essential aspect of a warehouse management system. Selecting the method of demand forecasting differs for different demand trends. Models for time series demand data can have

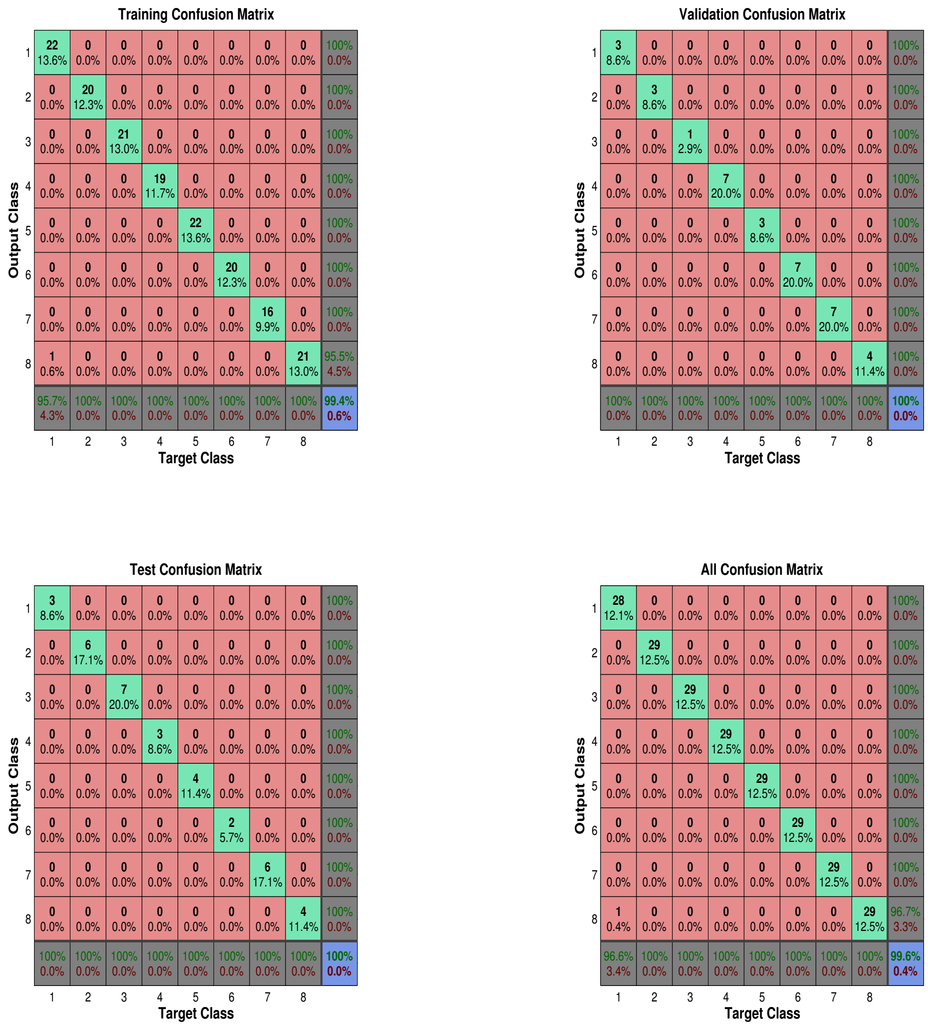

Figure 21. Confusion matrix for demand trend classifier.

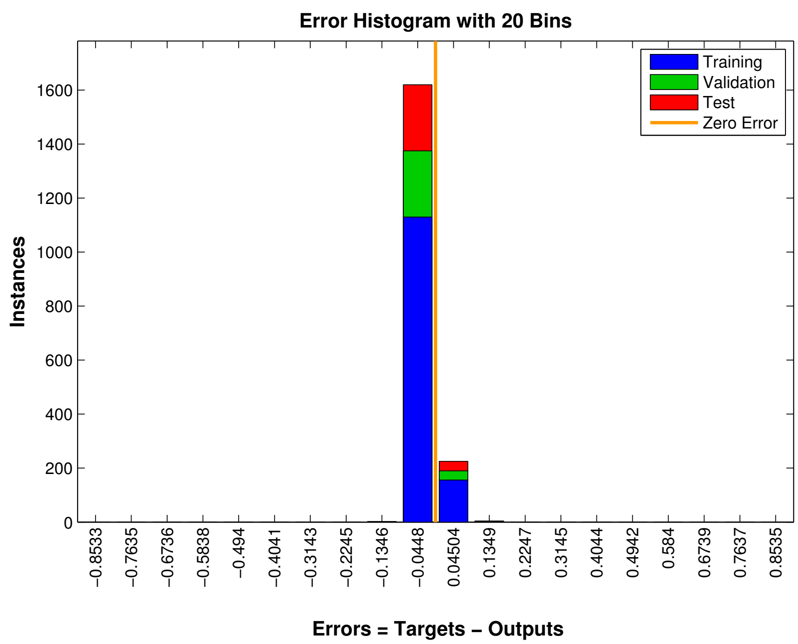

Figure 22. Error Histogram for demand trend classifier.

many forms.

Three classes of the autoregressive (AR), the integrated, and the moving average (MA) models are the most common ones for forecasting time series points. These three classes depend linearly on previous points on times series (Gershenfeld, 1999). Bootstrapping methods are more accurate for forecasting an intermittent demand time series. In this section, we will first describe the forecasting method of demand trends 1 to 7 in Figure 20, and then the forecasting approach for intermitent demand data will be discussed.

1 Autoregressive Integrated Moving Average (ARIMA)

One key assumption of ordinary regression analysis is that the errors are independent of each other. However, the ordinary regression residuals usually are correlated over time with time series data. This statistical assumption makes the ordinary regression analysis undesirable for time series data. There are regression models for time series analysis with the capability of adjusting estimated regression coefficients and standard errors when the errors have an AR structure.

As a consequence of violating the assumption of independent errors on ordinary regression, the statistical tests of the significance of the parameters and the confidence limits for the predicted values would be false. Further, the estimates of the regression coefficients are more effective when considering autocorrelation. The dependency of the regression residuals can improve the prediction of future values. In this study, AR error correction or a serial correlation correction is used to forecast the demand time series data except for the intermittent trend.

Before applying the regression model with AR errors, one may start by doing an ordinary regression and storing the residuals. If the residuals from the ordinary regression seem to have an AR structure, applying the regression model with the AR model improves the accuracy of forecasting.

A simple regression model with AR errors can be written as:

$$Y_t = \Upsilon_0 + \Upsilon_1 X_t + \xi_t \tag{53}$$

$$\xi_t = \chi_1 \xi_{t-1} + \chi_2 \xi_{t-2} + ... + \chi_m \xi_{t-m} + \varepsilon_t \tag{54}$$

$$\varepsilon_t \sim iid\, N(0, \sigma^2), \tag{55}$$

where Y_t and X_t are time series variables, Υ_t is the regression coefficient, χ_i is the autoregressive error model parameters, and ξ_t is the autoregressive error model variable. The notation $\varepsilon_t \sim iid\, N(0, \sigma^2)$ shows that each ε_t follows a normal distribution with mean 0 and variance σ^2 and is identically and independently distributed. The parameter ε_t is called white noise. For a higher order AR, the adjustment variables are calculated in the same way with more lags.

Since the current value of an AR series is correlated with all previous values, the AR model has a relatively "long" memory. Therefore, the AR model cannot be a good representative of the series, where the current value is only correlated with a few previous values. The "very short memory" property of the MA model makes it a favorable approach for modeling univariate time series. If it is algebraically equivalent to a converging infinite order AR model, the MA model will be invertible (AR coefficients decrease to 0 as we move back in time). The MA model is defined as the following:

$$Y_t = \mu + \xi'_t \tag{56}$$

$$\xi'_t = \varrho_1 \varepsilon_{t-1} + \varrho_2 \varepsilon_{t-2} + ... + \varrho_m \varepsilon_{t-m} + \varepsilon_t \tag{57}$$

$$\varepsilon_t \sim iid\, N(0, \sigma^2), \tag{58}$$

where the ϱ_is are the parameters of the MA model, μ is the expectation of Y_t, and the ε_t is a white noise error term and $\varepsilon_t \sim iid\, N(0, \sigma^2)$.

As a result, the ARMA model, which contains both AR and MA models, is written:

$$Y_t = c + \xi_t + \xi_t' \tag{59}$$

In the time series analysis, the ARIMA model is the integration of the AR and MA models. In other words, the ARIMA model is a generalized version of an ARMA model. The notation of ARIMA$(p, d, q)(P, D, Q)$ represents the model with p order of autoregressive model, d degree of differencing, and q order of the MA model. The parameters P, D and Q are respectively the autoregressive, differencing, and moving-average terms for the seasonal part of the ARIMA model. Note that the ARIMA(0,1,1) model without a constant is equivalent to the Simple Exponential Smoothing model.

Figure 23 represents our forecasting algorithm applied for the demand quantity prediction of trends 1 through 7. First, the demand trend class is found by the NN method described in section A. If the demand trend belongs to any class in 1 through 7, which were defined before, the ARIMA model will be executed. If the demand trend is intermittent, the bootstrapping method will be implemented.

Figure 23 shows that we can execute the *auto.arima*() function in the R software to find the best order for the ARIMA model. On the other hand, we produced a procedure to reach the best model, which examines some performance measures that test the alternative ARIMA models to select the best one.

The non-automated procedure scans four decision criteria, which include the autocovariance or autocorrelation function (ACF), partial autocorrelation function (PACF), root mean square error (RMSE), and σ^2, to find the appropriate model.

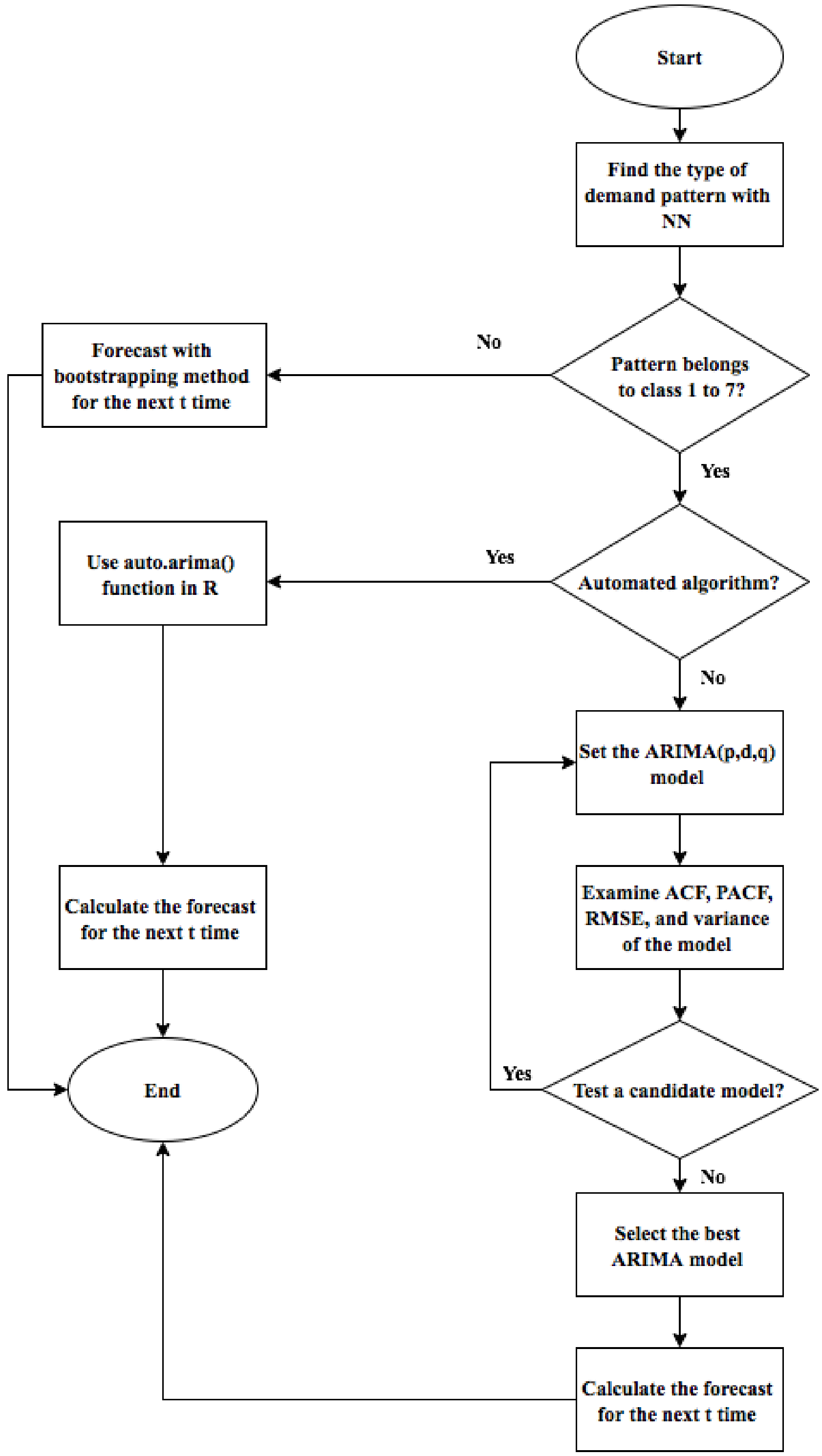

Figure 23. Algorithm of demand forecasting

The automated function takes care of these steps by using the embedded function in the R software *auto.arima*. The σ^2 that resulted from the automated algorithm is equal or greater than the non-automated algorithm in all cases.

TABLE 12

The ARIMA model results for instances of demand trends classes of 1 through 7

Trend	ARIMA Order	Coefficients							σ^2	AIC
1	(0,0,0)	Intcp.							5.76	242.66
		4.75								
2	(5,0,0)	ar1	ar2	ar3	ar4	ar5			0.02	-43.75
		0.65	0.56	0.38	-0.42	-0.18				
3	(5,0,0)	ar1	ar2	ar3	ar4	ar5	Intcp.		0.00	-111.79
		0.51	0.65	0.30	0.07	-0.54	3.24			
4	(1,0,3)	ar1	ma1	ma2	ma3	Intcp.			0.06	16.84
		-0.99	1.20	0.27	-0.04	1.18				
5	(4,0,2)	ar1	ar2	ar3	ar4	ma1	ma2	Intcp.	0.08	34.39
		-0.02	-0.33	0.70	0.34	0.40	0.76	1.03		
6	(1,0,1)	ar1	ma1	Intcp.					0.02	-38.30
		0.91	-0.33	1.69						
7	(1,0,1)(0,1,2)	ar1	ar2	ar3	ma1	sma1	sma2		0.52	133.94
		1.03	-0.46	-0.34	-0.59	-1.48	0.52			

One way to find the best order of the ARIMA model is through the visual inspection of the ACF and PACF plots and making a decision about the AR and MA orders by a sharp cutoff that appears in the ACF and PACF plots. For example, the PACF with a sharp cutoff while the slow decay of ACF represents an “AR signature” rather than “MA signature”. Notwithstanding, this method is not practical in our case, since we have to forecast the demand of very large number of the SKUs in the warehouse. Instead, we involve statistics obtained from these functions in our analysis.

Table 12 presents the results of the algorithm from Figure 23, for the seven demand trend classes of 1 through 7. The algorithm suggests the best model for forecasting the demand trend of the instances. Class 7, which represents a cyclic trend, has two AR terms: one MA term and two seasonal terms. After having the coefficients of the best ARIMA model, we are able to forecast the demand quantity of t time units ahead.

2 The bootstrapping method for intermittent demand data

Forecast errors can be costly in terms of keeping obsolete SKUs inside the forward area when using dynamic slotting. Although the traditional forecasting methods predict smooth demand data with proper accuracy, they are not capable of producing accurate forecasting for intermittent demand time data because these time series have a large number of zero values. Many of them assume that the probability distribution of the total demand over a planning horizon follows a normal distribution, which is not true. Croston (1972) was the first to recognize this phenomenon. This section is motivated to explore a way that increases the accuracy of the intermittent demand data forecasting using the bootstrapping methods.

Kourentzes (2013) propose a NN for forecasting intermittent demand data. They consider an inter-arrival rate of demand events to improve Croston's method. Gutierrez et al. (2008) also compare the NN forecasts against Croston's method, single exponential smoothing, and the Syntetos-Boylan approximation.

Wallström and Segerstedt (2010) evaluate performance/error measurements for the intermittent demand, since the comparison of different techniques is highly dependent on choosing appropriate decision criteria. Besides the traditional performance measures (e.g. MSE, RMSE), the new measurements' "number of shortage" and "periods in stock" are assessed to suggest a complementary measure.

Teunter and Duncan (2009) compare Exponential Smoothing, the Simple Moving Average, the Croston's method, and the bootstrapping methods in terms of forecasting accuracy. They conclude that Croston's method and the bootstrapping technique for forecasting intermittent demand outperform the MA and single exponential smoothing.

Croston's method applies exponential smoothing separately to the intervals between nonzero demands and the demand quantities in order to predict the mean demand per unit time. Willemain et al. (2004) show that the bootstrapping method generates more accurate forecasts of the demand distribution over a fixed planning horizon, compared to the exponential smoothing and the Croston's forecasts.

Following Croston's method, Willemain et al. (2004) apply a normal distribution with a specific mean and a standard deviation. The authors also show that there is no statistically significant difference between the Croston's method and exponential smoothing at forecasting the entire lead time and that the bootstrapping method outperforms both. Furthermore, the accuracy of the Croston's method encounters a very serious bias compared to the other techniques (Teunter and Sani, 2009). Thus, we select the Willemain et al's method for intermittent demand data forecasting.

Forecasting the demand size and the nonzero demand intervals are two important issues, which are considered in intermittent demand forecasting. We implemented Willemain et al's algorithm, as shown in Figure 24. This algorithm performs a two-state first order Markov process to model the autocorrelation. The forecast of the sequence of zero and nonzero values are conditional on having or not having a demand in the last period ($X(T) = 1$ or $X(T) = 0$).

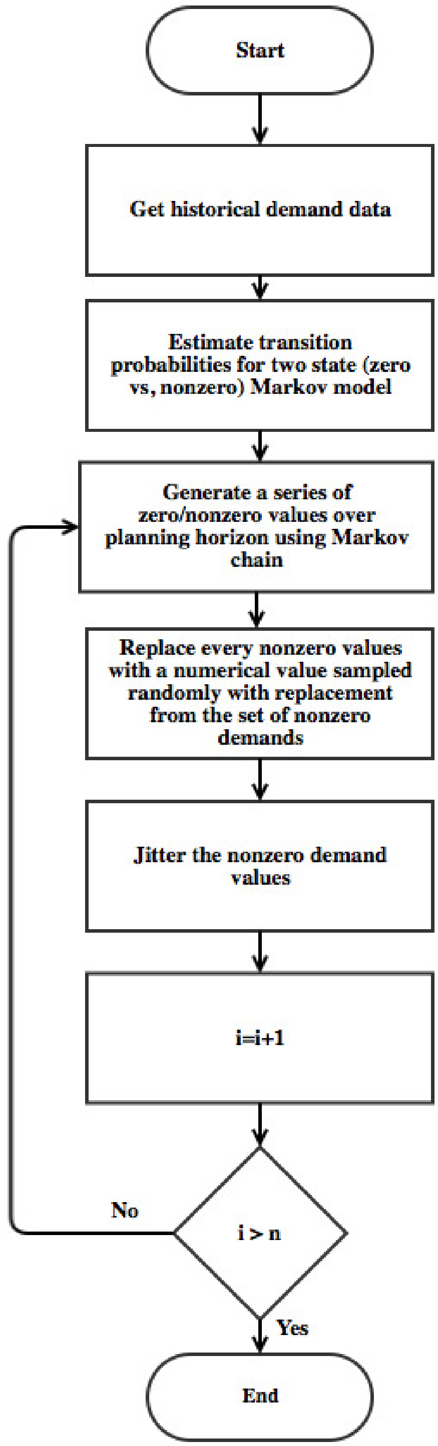

Figure 24. Algorithm for forecasting the intermittent demand data

They define the jittering process as follows:

$$JITTERED = \begin{cases} X^* & \textit{If} \quad JITTERED \leq 0 \\ 1 + \lfloor X^* + Z\sqrt{X^*} \rfloor & \text{otherwise,} \end{cases}$$

where X^* is one of the historical demand values randomly selected and Z is the standard normal random deviate.

TABLE 13

Summary statistics for intermittent demand forecasting

	Forecasting lead time (% of data size)				
	10%	20%	30%	40%	50%
MMSE	0.1575	0.1588	0.1588	0.1587	0.1580
MRMSE	0.2691	0.3269	0.3580	0.3694	0.3752
MMAE	0.1393	0.1403	0.1404	0.1399	0.1393

Table 13 summarizes the statistics of the intermittent demand forecasting for 6000 SKUs that follow this pattern during 52 days of historical demand data. On average, each SKU demand profile has a 87.05% zero value. We evaluate different chunks of the forecasting horizon in terms of the fraction of available historical demand. For example, if the forecasting horizon is supposed to be 10% of the 52 days, the first 52-5=47 days would be the training input data for the algorithm in Figure 24, and the remaining 5 days are the testing data. A bootstrap size of 1000 was arbitrarily chosen.

The results from Table 13 show the capability of the Willemain et al's method for intermittent demand data forecasting, which is a good fit for forecasting class 8 of our defined demand patterns. The errors do not experience fast growth by enlarging the lead time and look satisfactory even for larger forecasting horizons.

In summary, we developed an algorithm for the demand forecasting of different demand trends. The Control Charts Pattern Recognition (CCPR) using ANN presented excellent accuracy in terms of detecting the demand trend. The ARIMA method is recommended for forecasting the demand quantity of patterns belonging to classes 1 through 7. However, the bootstrapping method fits well for the intermittent demand pattern, class 8.

In summary, CCPR using ANN presented excellent accuracy in terms of detecting demand trends. The ARIMA method is recommended for forecasting the demand quantity of smooth patterns that belong to classes 1 through 7. However, the bootstrapping method is a good match for the last class, intermittent demand pattern. The forecasted demand data are applied to the dynamic model in the next chapter, when the perfect future demand data is not available.

CHAPTER V

THE DYNAMIC FORWARD-RESERVE PROBLEM

In the static forward-reserve problem, the SKUs' positions are fixed and a particular SKU designated for a slot in the forward area will be replenished in the same slot during the planning horizon. Because the warehouse environment is dynamic, why not consider storing those SKUs with certain demand trends in the forward area?

Dynamic Forward-Reserve Problem (DFRP) changes the layout of the forward area by real-time replenishments of the correct SKUs in the naturally emptied slots by picks. This approach should not be confused with similar concepts. *DFRP* is different from "warehouse reshuffling", which refers to the process of converting the current slotting to a designated target map. Relocating SKUs to convert from the current slotting to the target map (optimal layout of the forward area) obtained from static slotting optimization are typically called "moves" or "slotting moves." The interval between the first and last move for getting from current state to target map is called the "reshuffling period."

We have heard from warehouse managers that they want to avoid the large number of moves suggested by the static slotting software. The *DFRP* approach improves the layout of the forward area on a frequent basis by using the replenishment of empty slots with proper SKUs.

The question of "Should we reslot our warehouse?" is popular in practice. One strategy is running the static model periodically, e.g. monthly, but warehouse managers should know how often they need to re-layout the warehouse to update the current slotting to the target map.

The best interval for the forward area re-layout is uncertain. If the selected period is shorter than the best time for re-layout, it will be disruptive. Some SKUs are moved to the reserve area without attaining their expected savings from being stored in the forward area. On the other hand, if the selected period is long, we cannot be sure if we have the optimal layout of the forward area over time. As a result, improper SKUs, which are no longer eligible to be in the forward area due to their demand trend, will stay there. A good reslotting methodology not only reslots seasonal items, but also corrects the mis-slotted items of other demand pattern classes.

Therefore, having a strategy that guarantees the best layout of the forward area is critical. The goal is to make sure that the assignment and allocation of SKUs to the forward area are always updated during the planning horizon. Three concepts of updating the layout of the forward area are:

1. **Dynamic warehouse reshuffling**: This method is based on repositioning and (or) the adding/dropping of SKUs in the forward area by moving them. The number of empty slots and their positions vary in each state depending on the demand profile. Given the varying current slotting and the target map at each unit of time, this model suggests the best moves to convert the current slotting to the target map during the planning horizon. The goal is to have both the minimum number of moves and the shortest moves from the origin slot to the destination slot. This methodology takes advantage of the empty slots in a real-time process to replenish the correct items in the correct locations. In this strategy, the slots in the forward area are not identical, and moving the SKUs located close to each other costs less than moving the farther ones.

2. *DFRP* **with no moves**: Given an empty slot in a forward area and full information about the historical demand, this concept decides whether the restocking is with the same SKU or a different one. This approach explicitly schedules real-time replenishment activities and considers the sequence of replenishments.

3. *DFRP* **with moves**: This method, which is the complement of the second method, gives the opportunity of extracting slow movers in the forward area and moving them to the reserve area. Therefore, we not only make decisions about picking locations and replenishments, but also about those SKUs that require to be moved to the reserve area. The slots of the forward area are identical in the second and third approaches. These approaches do not require the slotting map as an input.

Since the warehouse environment is dynamic, it is necessary to continuously update the forward area layout. In this chapter, we will address the last two concepts — *DFRP* with no target map or designated moves. Factors that account for this dynamic environment include the seasonal item demand fluctuations, promotional policies, and the general economic conditions that affect the demand trends. We call these factors "destabilizing events". When these events occur, it is likely that temporarily excluding and including some SKUs in the forward area can reduce costs. Instead of using accumulated annual demand data, we use raw order data, which preserves the knowledge that can be obtained from real-time demand trends. We will use the demand forecasting method explained in chapter IV to provide the forecast demand data for the dynamic model that will be discussed.

A Literature Review

In this section, we first review the studies with a dynamic approach in a warehousing context, such as the dynamic order picking system, the dynamic order replenishment system, the dynamic inventory strategy, the dynamic slotting of the correlated SKUs, and the dynamic lot-sizing. Next, we address the necessity for developing the dynamic assignment and allocation of SKUs in the forward area.

For a unit load warehouse, Goetschalckx and Ratliff (1990) develop a shared storage model, where different SKUs are stored in the same slot over time. They assume that each unit load requires the same space. Therefore, the cost of replenishing or picking a unit load from a storage unit is independent of the SKU type. They present two heuristics for static and adaptive policies, in addition to an optimal storage policy for a balanced system, where the number of arriving units is equal to the number of departing units. They conclude that a shared storage policy based on duration of stay will reduce travel time.

For a less than unit load warehouse, Landers et al. (1994) develop a framework for a dynamic order picking system. Their study considers the correlated and commonality of demand within families. These considerations lead to resizing the slots and SKU reslotting. A clustering algorithm determines the group of SKUs that are stored together based on the long-run average correlation. As a result, a long run average flow may cause ineffective slotting. Sadiq et al. (1996) also study the dynamic environment, in which the items go through life cycles and product mix changes. They show that their algorithm for the Dynamic Stock Location Assignment Algorithm (SLAA) outperforms the cube per order index at the order processing time minimization when popularity and correlation of demand are changing over time.

Yingdea and Smith (2012) address the dynamic slotting problem based on SKU correlations. While we improve the picking and replenishment costs by dynamic SKU assignment and allocation in the forward area, Yingdea and Smith (2012) improve the picking efficiency by assigning correlated SKUs to their adjacent slots in a warehouse. The authors propose an ant colony optimization with a slot-exchange policy to assign the correlated SKUs. The limitation of this paper is the assumption that the cartonization information is known. They use the same Mixed Integer Programming (MIP) formulation proposed by Kim and Smith (2012), whose objective is to minimize the pick wave span and maximize the total completion time among all pickers. Kim and Smith (2012) study SKU assignment to zone-based carton picking DC, where the WMS makes the routing decisions dynamically. They propose four two-phase heuristics for the slotting problem. One of these heuristics applies simulated annealing based on correlated interchanges. Their picking area becomes completely emptied each day and is replenished after every pick wave.

A limited number of studies evaluate the dynamic perspective by analyzing the warehouse activities like the dynamic order picking system and the dynamic order replenishment. Bukchin et al. (2012) develop a Markov-based model that determines whether to go on a tour and pick the accumulated orders or to wait for more orders to arrive at every period. Therefore, the solution decides the batching orders in a dynamic, finite-horizon environment. Order tardiness and overtime costs of the pickers are minimized in their model.

Gong and De Koster (2008) develop a dynamic order picking system, in which orders arrive in real-time and the picking information is dynamically changed during the picking operation. Therefore, the pick locations are not fixed in a picking cycle and the response time is reduced. They show that the polling-based picking systems outperform the traditional batch picking systems using optimal batch sizes in terms of order waiting and throughput time.

Further, Berman and Kim (2001) studied a dynamic order replenishment system, which follows an Erlang distribution. They show that Erlang lead times are more stable than exponential lead times in terms of the cost. Finally, they recommend dynamic policies with an adjusted reorder point based on customer orders and the inventory status. This method is more efficient than traditional inventory policies.

Strack and Pochet (2010) propose an integrated model for warehousing and inventory planning with different levels of integration. They demonstrate that this integration will considerably reduce the cost of warehousing and the inventory system, since the space allocation and replenishment decisions are closely dependent and can be translated to each other. Their model determines: 1) the products that are assigned to and only picked from the reserve area, 2) the products that are directly supplied in the fast picking area and are only picked from the forward area, and 3) the products that are supplied in the reserve area and are picked from the forward area and a number of locations allocated to them in the forward area. They introduce two integration levels: the lower level and the higher level. The lower level considers the limited space in the warehouse but the inventory model ignores the routes taken by the products, which includes external supplies sent to the reserve area or directly to the forward area, and the number and capacity of the locations in the forward area. The higher level considers the capacity constraint of the forward and reserve areas as well as the reception cost of each product.

Our dynamic model makes decisions about the forward area assignment, the allocation, and the replenishment regarding the SKUs' inventory in this area. Berman and Kim (2004) study the dynamic inventory strategy and the replenishment policy.

They propose a Markov based model to reach an optimal dynamic inventory strategy and maximize the facility's profit. As an application of dynamic programming in inventory management, Shapiro (2011) delivers an adjustable multistage robust optimization model. This research also analyzes a risk averse stochastic programming.

Many articles have studied the Dynamic Lot-Sizing Model (DLSM), which consists of the inventory problem for single or multiple item(s) and transportation functions. Kim and Lee (2012) propose a metaheuristic for the dynamic lot sizing and shipment scheduling problem. The objective is to minimize the total cost including the associated costs of ordering, inventory, holding, and freight. A genetic algorithm is recommended by Kim et al. (2012) for solving the problem of inbound ordering and outbound dispatching. The authors consider the dynamic demands over a discrete finite time horizon. Kim and Lee (2013) also suggest a heuristic for solving the problem of scheduling multiple products with a dynamic demand. These three articles determine the order and shipment quantities of product i and the number of containers used in period t.

The rest of the chapter V is organized as follows. Section B describes the mathematical model of a dynamic discrete forward-reserve problem (method 1). The comparison of the dynamic model with the static forward-reserve model will be discussed in section D.

B Mathematical Model

The static FRP has several assumptions:

1. **A fixed set of SKUs selected for the forward area**. The set of SKUs assigned to the forward area is fixed during the planning horizon. So, if a slot becomes empty, the same SKU as before will be replenished in that slot. The

dynamic model decides based on the SKUs' demand patterns whether or not the same SKU should be replenished in the empty slot(s).

2. **A fixed number of slots allocated to the SKUs selected for the forward area**. Due to the demand fluctuation, the number of slots given to the SKUs in the forward area should change over time. The dynamic allocation allows varying number of slots to the assigned SKUs to the forward area, when required.

3. **Restricting the model to always pick from the forward area if the SKU exists there**. If an SKU is assigned to the forward area, no matter the order quantity, it is picked from the forward area and not from the reserve area. Nonetheless, the dynamic model intends to command the picking of SKUs with a high order quantity from the reserve area, not the forward area, to reduce the replenishments.

4. **Stocking the fixed steps of multiple units in the forward area**. The quantity stocked in the forward area is defined as the steps of multiple units. In other words, the replenishment units is the factor based on the number of allocated slots (e.g. if the maximum storage unit for SKU x is 100 cases, and the allocated slots is two slots, 200 cases is stocked in each restocking event). However, there is the chance of replenishing of the non-empty slots with less than the maximum storage units.

5. **Pick quantity is always less than the full allocation**. The pick quantity of the SKUs in the forward area is always less than the full allocation of the SKU in the forward pick area. However, the large order quantities may be referred from the reserve area to save the replenishment costs.

6. **The unlimited restock quantity by the restocker**. Likewise, the number of cases that a restocker can restock is unlimited in a static model. However, the material handling device has a limited capacity in the dynamic model.

7. **A non-integral number of replenishments**. The number of restocks is not integral in a static forward-reserve problem. This assumption and also the free first restock affect the optimal solution. A dynamic model delivers an integer number as the number of replenishments for each SKU.

8. **The frequency of running the static model is assumed as known**. The static model does not address the question of "how often to run the optimization model to update the assignment and allocation of the forward area". Rather, the dynamic model optimizes the layout of the forward area continuously.

In this section, we propose a generic Mixed Integer Programming (MIP) formulation for dynamic and discrete assignment and allocation of SKUs into the forward area. The aforementioned assumptions are relaxed in the proposed MIP model. The real world warehouse requirements inspired us to avoid the static model assumptions. We embedded some constraints in the model to address these critical requirements. The inputs of this model include: the item file containing the SKUs' dimension, the slot file containing the information of the slots in the forward area, and the order file.

Parameters:

N: Number of SKUs (i=1,2,...,N)

T: Number of periods (t=1,2,...,T)

c: Restock cost

c_1: Cost of picking from the forward area

c_2: Cost of picking from the reserve area

e_i: Number of units of SKU i that can be stored in one slot

d_{it}: Demand of SKU i at time t

p_{it}: Number of picks of SKU i at time t

η: Total number of slots in the forward area

Decision Variables:

x_{it}: 1 if SKU i is picked from the forward area at time t; 0 otherwise

y_{it}: 1 if SKU i is restocked at time t; 0 otherwise

I_{it}: Inventory of SKU i in the forward area at the end of time t

R_{it}: Number of units of SKU i that are restocked at time t

n_{it}: Number of slots occupied by SKU i at time t, $n_{it} \in \{0, 1, 2, 3, ...\}$

1 The generic MIP model of *DFRP*

We propose the MIP mathematical model for the dynamic-discrete forward-reserve problem for the first time as follows:

$$\text{Min} \quad C_1 = c_1 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it} x_{it} + c_2 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it}(1 - x_{it}) + c \sum_{i=1}^{N} \sum_{t=1}^{T} y_{it} \tag{60}$$

subject to:

$$-I_{it} + I_{i,t-1} + R_{it} - d_{it} x_{it} = 0 \qquad \forall i, t \geq 2 \tag{61}$$

$$\sum_{i=1}^{N} n_{it} \leq \eta \qquad \forall t \tag{62}$$

$$R_{it} \leq \eta e_i y_{it} \qquad \forall i, t \tag{63}$$

$$n_{it} \geq \frac{I_{i,t-1} + R_{it}}{e_i} \qquad \forall i, t \tag{64}$$

$$R_{it}, I_{it} \geq 0 \qquad \forall i, t \tag{65}$$

$$x_{it}, y_{it} \in \{0, 1\} \qquad \forall i, t \tag{66}$$

$$n_{it} \in \{0, 1, 2, 3, ...\} \qquad \forall i, t \tag{67}$$

This model decides the replenishment of empty slots, which appears during the order picking, with the same SKU or a different one. Unlike the static forward-reserve problem, which is non-linear, the dynamic model is linear. The objective function in equation 60 is the total cost of picking and replenishments of SKUs assigned to the forward or reserve area.

In this dynamic model, SKUs may occupy more than one slot in the forward area. This model takes advantage of a "shared storage" policy in which the residual empty space generated by order picking might be aggregated, such that more SKUs could be put in the forward area. In each time period, some slots that become empty provide the opportunity of storing the appropriate SKUs in the forward area. However, each SKU reserves its own restocking slots as it nears the time for replenishment, as in "dedicated storage."

The concept of time period t in the dynamic model may vary in different warehouses with different picking/replenishment methods and SKU activities. For example, the warehouses, which apply the "wave" picking method, may choose the length of the wave as t. A wave is constructed with groups of orders. t may also be corresponded to the daily items' flow. It is expected that t expresses a shorter intervals in more active warehouses with higher product flows. Comparing to the inactive warehouses, the inventory level of the slots in active warehouses are depleted faster due to more frequent picks. t should be small enough to capture the changes in inventory status of slots and trigger the replenishments. Choosing a short t interval (e.g. hour) in inactive warehouses will not add value to the dynamic model because of no change in inventory levels of slots over a sequence of t.

The unit of inventory in our problem is selling units. Constraint 61 guarantees that the demand is satisfied by picking either from an on-hand inventory in the forward area or from the reserve area. We can track the existence of an SKU in the forward area by its inventory level, I_{it}.

If the demand of SKU i is satisfied from the forward area at time t, then $x_{it} = 1$. However, this model does not mandate picking from the forward area, if $I_{it} > 0$, since product i stocked in the forward area can more efficiently be picked from the reserve area in the case of high order quantity per pick. The model makes such decisions implicitly.

Constraint 62 does not allow the total allocated slots to the SKUs in the forward area to exceed the total number of slots. Further, constraint 63 makes the binary variable of replenishment, y_{it}, equal to 1 if SKU i is restocked at time t. Constraint 64 establishes the number of slots given to each SKU i at time t.

Our solution shows that relaxing the integrality constraint for variables R_{it} and I_{it} leads to the integer solution. However, n_{it} should be selected from the integer numbers. n_{it} is the ceiling function of R_{it} and $I_{i,t-1}$ as below:

$$n_{it} = \left\lceil \frac{I_{i,t-1} + R_{it}}{e_i} \right\rceil$$

C Numerical example

Before discussing our numerical example, we first describe the procedure for running the dynamic model in Figure 25. When the perfect information about the future demand data is not available, the dynamic model is not a one-time run model because the forecasted demand data is updated at each run r. The dynamic model is run w (time window) times with the updated data based on the last H periods. The picks and replenishments' solutions of each run are saved for computing the final total cost during the planning horizon.

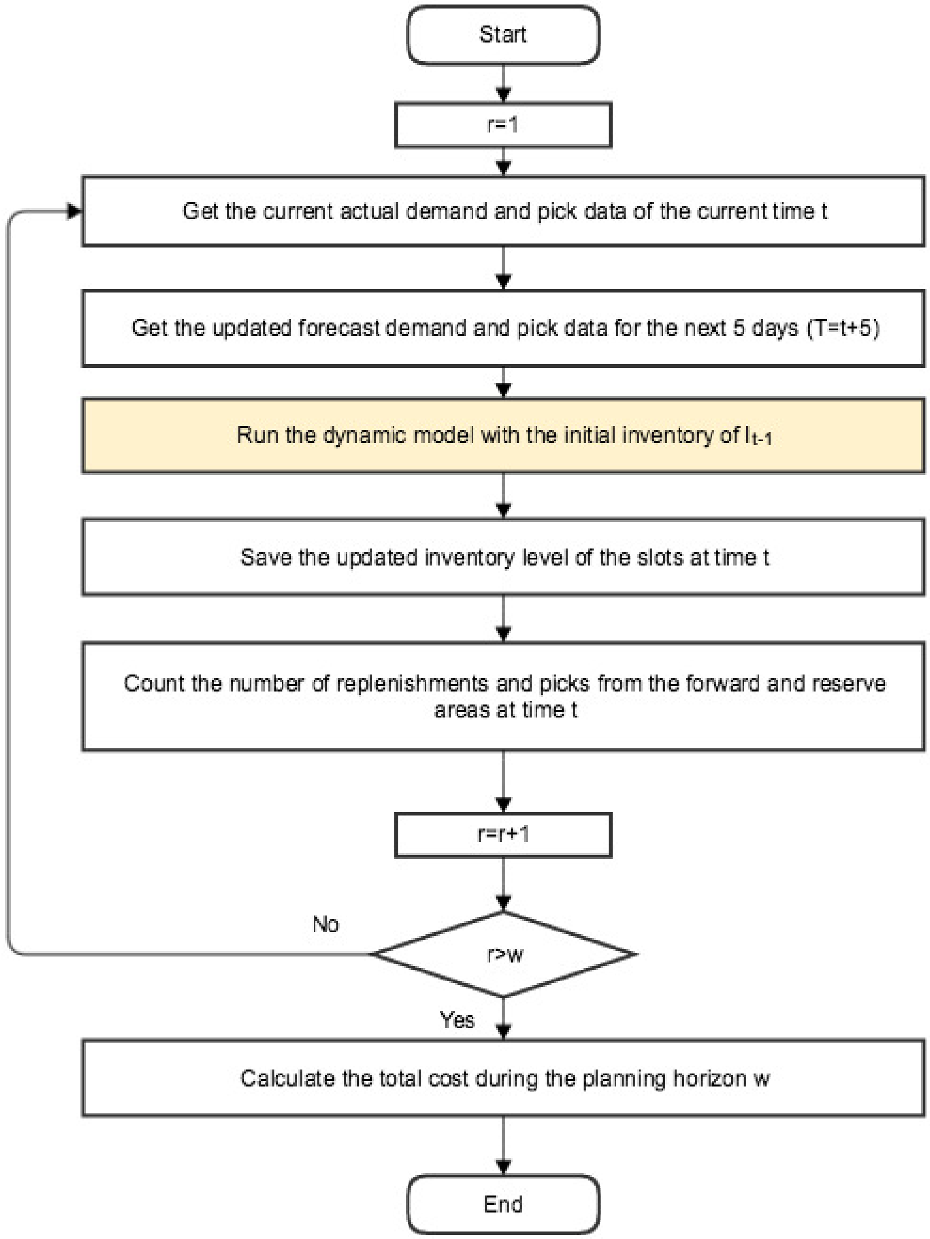

Figure 25. The run procedure for the dynamic slotting model

The dynamic model plans for the next T periods at each run r. Since the picks and the demand data of the initial period are assumed known, the inventory level reductions that corresponded to the first period are actual at each run r. The initial period replenishments are commanded and the inventory level of the slots are updated. The runs continue to cover the whole planning horizon.

Selecting a sufficient time window w makes the comparison between the dynamic and static models s more reliable. It is worth noting that the dynamic model can plan for the next 6-8 periods in a reasonable time, depending on the number of SKUs. On the other hand, when T is not sufficiently large, the model generates short-sighted decisions. Therefore, selecting a proper T is challenging in different industries. In addition, the duration between two consecutive runs of the dynamic model, period t, is important. The large time segments can result in a delay of the replenishment or moves decisions. However, choosing small periods is not computationally efficient. In our numerical examples, we found that the daily decisions made about the forward area are sensible.

We compare the static model $(G_2 - A_4)$ with the dynamic model. In our example, we consider a warehouse with 5000 SKUs. The relevant values of other parameters of the model can be found in Table 14. We used the information of the SKUs' dimensions, which belong to a real world warehouse. Our order generator simulates the eight different types of demand trends explained in section A. The order data for 50 days of history is simulated, and a dynamic slotting strategy that uses the daily demand quantity forecasting for 30 days ahead is delivered.

We run 7 experiments designed for the different sizes of the forward area listed in Table 15 in order to perform our comparison. In this example, the unit of time is one day in the dynamic strategy. The perfect information about the future demand data is assumed as known in this example. The parameters associated with c_1, c_2 and c costs have been selected from the default values of a commercial slotting software. The units used for measuring the travel distance of picking and replenishment activities, such as inch, describes the unit of unit of cost in this dissertation.

TABLE 14

The values of model's parameters

Item Info	Slot Info.	Order Info.	Cost Info.
SKU dimensions file	$W = 18$ $H = 16.5$ $L = 96$	Order generator	$c = 170$ $c_1 = 27$ $c_2 = 100$

TABLE 15

Results of cost and solution time comparisons of the static versus dynamic model

η	C_{stat-H}	ST_{Stat-H}	$C_{Stat-PI}$	$ST_{Stat-PI}$	C_{Dyn-PI}	ST_{Dyn-PI}	$Imp_1\%$	$Imp_2\%$	Gap
100	1761513	0.06	1677445	0.06	1538725	422	12.65	8.27	0
150	1516117	0.07	1438928	0.05	1425630	388	5.97	0.92	0.0014
200	1446542	0.09	1356799	0.05	1329306	486	8.10	2.03	0.0036
250	1368497	0.08	1272363	0.06	1244400	563	9.07	2.20	0
300	1296999	0.08	1197283	0.06	1161791	630	10.42	2.96	0.0131
350	1236235	0.07	1120830	0.06	1084762	1113	12.25	3.22	0.0197
400	1165466	0.08	1050649	0.06	1009547	794	13.38	3.91	0.0192

C_{Stat-H} = Cost of static model using historical demand data,
$C_{Stat-PI}$ = Cost of static model using perfect information about the future data,
C_{Dyn-PI} = Cost of dynamic model using perfect information about the future data,
ST = Solution time (seconds),
Gap = Absolute MIP gap tolerance for the dynamic model,
$Imp_1\%$ = Improvement percentage of the $Dyn - PI$ over $Stat - H$ model,
$Imp_2\%$ = Improvement percentage of the $Dyn - PI$ over $Stat - PI$ model.

Assuming that the picking cost from the forward area is fixed in all experiments, we expect that the larger forward area results in a lower total cost. Results of Table

15 for both static and dynamic strategies confirm this expectation. Columns $Imp_2\%$ and $Imp_2\%$ of Table 15 presents the improvement percentage of the dynamic over the static model for different sizes of the forward area.

Table 15 shows that the dynamic model outperforms the static model for any size of the forward area. The costs of the dynamic model are considerably lower than their static counterparts in all cases, and the greatest benefit of 12.65% is achieved where the forward area is very small (100 slots).

The results of Table 15 shows that the dynamic strategy is more effective for small and large sizes of the forward area compared to the medium size in this example. In small forward areas, since a few number of slots are available, selecting the best set of SKUs for the forward area is underlined, and poor decisions about the assignment and allocation are more expensive. The dynamic model can introduce new fast movers to the forward area and keep the layout updated in the small size case. In large forward areas, it is expected that the proportion of improper SKUs suggested by the static model for the forward area are higher than the medium size forward area, resulting in a total cost increase. Larger forward areas provide the dynamic model more opportunity and flexibility in dynamic slot allocation (changing the number of allocated slots over time based on the changes in demand), which results in cost improvements.

D Comparison of the static and the dynamic models with multiple runs of the static model

The best time to re-layout the forward area regarding its current state is still a critical unanswered question in practice. One may assume that running the static model periodically competes with the dynamic strategy. In this section, we compare these two strategies and quantify the benefits of the dynamic model over updating the forward area in certain intervals by assessing three scenarios:

1. PI: The dynamic forward-reserve problem with Perfect Information (PI) about the next k units of time (day).

2. FI: The dynamic forward-reserve problem with Forecasting Information (FI) about the next k units of time (day).

3. S: The static forward-reserve problem with updating the layout of the forward area in certain points of the planning horizon.

Since there are no established tests for running the discrete forward-reserve problem in certain periods, we design an example that includes time periods. Running the static model in a certain time interval to get the most updated layout of the forward area is a common way to take the demand changes into account. Then the "moves" from the forward area to the reserve area are designated to exclude the obsolete slow movers from the forward area. We charge each SKU move for "transition" costs equal to the replenishment cost, when an SKU should move from the forward area to the reserve area in the update points.

The demand of every SKU in the warehouse, which is an input of the static model, is not fixed and changes over time in unanticipated ways. The source of this change is due to the changes in customers' behavior over time. Following the language of predictive analytics, we call the demand per year, which is supposed to be predicted, the *concept* and the process of shifting the concept over time is called the *concept drift*.

As Figure 26 shows, the horizontal axis represents both a history and future planning horizon. The history is up to point zero in the diagram. The static model is run only once at time zero in the first scenario. The layout of the forward area would be fixed during the planning horizon w.

While the first scenario has a single chunk of data, others have sequential data chunks of sizes a_1, a_2, ..., a_m. We run the static model every a_1 units of time (e.g. days) in the second configuration. We shrink the intervals in the next runs until a_m days in the last configuration, which is the shortest period of running the static model ($a_1 > a_2 > ... > a_m$). Note that in all configurations, the static model will receive the demand profile of the last H days as an input, and the *concept*, which is the total demand in the last H days, is drifting.

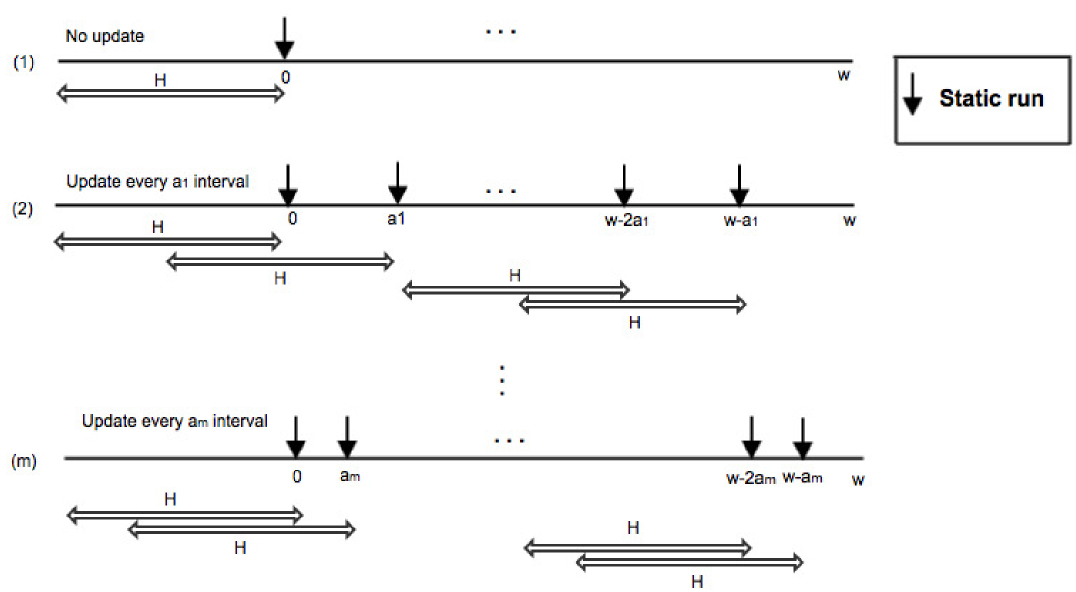

Figure 26. The planning horizon diagram with different run intervals for the test example

As Figure 27 illustrates, we run the static model once (no update), every 15, 10, and 6 days ($a_1 = \frac{30}{2}, a_2 = \frac{30}{3}, a_3 = \frac{30}{5}$). Here, the historical demand is defined as the most recent 50 days ($H = 50$) and the planning horizon incorporates 30 days

($w = 30$). Other parameters' values can be found in Table 14. We selected heuristic $G_2 - R_4$ for solving the static model, since it showed the best performance among all heuristics discussed in Chapter II.

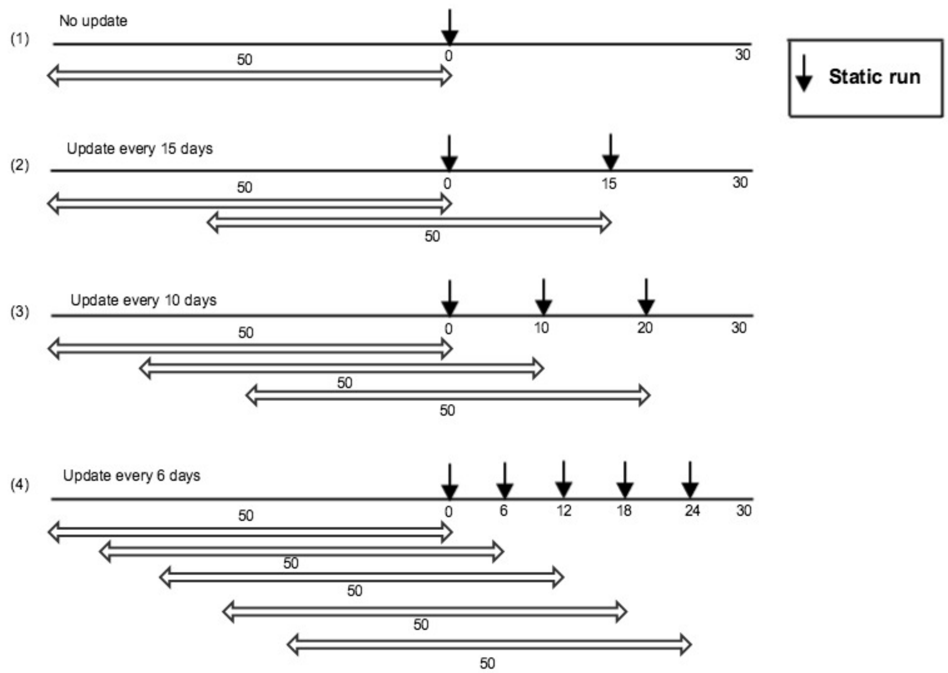

Figure 27. The planning horizon diagram with different run intervals

Table 16 represents the total picking and replenishment cost of the static model solved by G_2 for different scenarios. For the first scenario that has a single data chunk, we have only one total cost over days 31-80. Nonetheless, other scenarios have more than one cost considering a different data chunk i ahead, which we show with c^i. For example, the scenario with $a_2 = 10$ calculates the costs of chunks 51-60, 61-70, and 71-80, which are equal to c^1, c^2, and c^3, respectively. The sum of these three costs provides the total picking and replenishment cost, C, during 30 days for this scenario.

The results of C in the last column of Table 16 depicts how costly it would be to select an inappropriate interval for running the static model, a_i. The total cost of running the static model every 30 days (no update) is less than early update intervals 15, 10 and 6 days that impel "early shock" to the static model. Depending on the activity distribution of the items in the forward area, it takes longer for many SKUs to emerge as cost effective. The SKUs that encounter the replenishment cost should stay for awhile in the forward area to generate expected savings by picks. Deleting them early from the forward area and moving them to the reserve area not only incurs the moving cost, but also prevents expected savings per pick.

The results of the last two scenarios in Table 16 with two and four updates show that the costs go down and up. Since the slotting of the first period is initiated with an empty forward area, all slots in that period are replenished. However, the following periods begin with non-empty slots. Therefore, the number of replenishments is lower in the middle of the planning horizon, which reduces the total cost. The slots are depleted over time and need to be replenished again in the final periods. As a result, the costs will eventually grow again.

TABLE 16

Picking and replenishment cost for the static model.

	c^1	c^2	c^3	c^4	c^5	C
No update	1368497	0	0	0	0	1368497
One update	697707	684351	0	0	0	1382058
Two updates	499925	416805	485430	0	0	1402160
Four updates	315898	288248	240906	279170	317616	1441838

In the cases of running the static model more than once if the set of SKUs in the new layout is different from the previous layout, we move the SKUs not found in the new layout to the reserve area with a cost equal to the replenishment cost. Table 17 shows the different SKUs (DSK_i) and different slots (DSL_i) in chunk i of the previous and the next state. The first columns of Table 17 are zero, since we start from an empty forward area. Table 18 presents the cost of moves, cm_i, in data chunk i. Finally, the total cost, $C_S = C + CM$, in the last column of Table 18 will display the total cost of each scenario in the static model.

There is a trade off between continuously going with the previous layout of the forward area and having the most updated layout of the forward area but undergoing the moving cost. Table 18 suggests not to reslot before 30 days.

TABLE 17

The number of different SKUs (DSK_i) and the different slots (DSL_i) in the previous and next states of the forward area in a static model

	DSK_1	DSL_1	DSK_2	DSL_2	DSK_3	DSL_3	DSK_4	DSL_4	DSK_5	DSL_5
No update	0	0	0	0	0	0	0	0	0	0
One update	0	0	15	15	0	0	0	0	0	0
Two updates	0	0	14	14	18	18	0	0	0	0
Four updates	0	0	9	9	5	5	13	13	16	16

TABLE 18

Cost of moving to reserve area in the static model

Config.	cm_1	cm_2	cm_3	cm_4	cm_5	CM	C_S
No update	0	0	0	0	0	0	1368497
One update	0	2550	0	0	0	2550	1384608
Two updates	0	2380	3060	0	0	5440	1407600
Four updates	0	1530	850	2210	2720	7310	1449148

We used the same data set for the dynamic MIP model discussed in section B with having the Perfect Information (PI) of the future demand. The cost of the dynamic model with perfect information about the future demand is C_{PI}.

Table 19 presents the comparison results of the static model and the dynamic model with perfect information about the future order transactions. The last column of this table shows the promising cost improvements of the dynamic model over the static model. The total costs of dynamic scenarios, PI, is always less than the static model. Note that the availability of the movers to convert the previous state of the forward area to the new layout in this short interval is questionable. In reality, it takes time to get from the current state to the target map, whereas we assume no delay for reslotting. The percentages of saving attained by the dynamic model in Table 19 provide the cost justification of using the dynamic model for the forward-reserve problem rather than the static model.

TABLE 19

The total cost and savings (%) obtained from the static and dynamic models (PI).

Config.	C_S	C_{PI}	saving (%) $C_{PI \to S}$
No update	1368497	1244400	9.07
One update	1384608		10.13
Two updates	1407600		11.59
Four updates	1449148		14.13

The traditional wisdom assumes that running the static model more frequently generates more savings than less frequent runs. Nevertheless, Table 19 shows that the savings are greater when the layout of the forward area is updated in longer intervals. The reason is that each SKU in the forward area has its own "minimum payback" period. If an SKU exists in the previous layout but is not found in the new target map, it is moved from the forward area to the reserve area. This approach prevents the receiving of the whole expected benefits after the last replenishment. In other words, the SKUs may leave the forward area before reaching their minimum payback point. In each update, it is assumed that the next decision about the layout of the forward area is not influenced by "what state the SKU is in." In essence, the next update "forgets" how much time has elapsed from the last replenishment of the SKUs in the forward area. This *memorylessness* property is assumed in each update, causing more frequent updates that result in higher costs. Our results suggests to run the static model no earlier than 30 days in this example.

These comparisons provide a basis for warehouse managers to select their desired methodology for updating the forward area. While the static model requires the movers to convert the previous state to the target map, the dynamic strategy takes the advantage of pick clean (having empty slots by picks) to replenish new items in the slot and updates the layout of the forward area.

At the end of this chapter we will show the warehouses that keep short life cycle products, such as fashion products, as well as highly volatile products in order to receive more benefits from the dynamic model, compared to the warehouses that store conventional and fixed demand products.

E Model enhancement

In this section, we test the dynamic model by applying our forecasting system and assess the potential change requirements for making the model more realistic. Then, we will compare the static model, S, to the dynamic model with forecasted demand data, FI, and new features, including the option of moving to the reserve area and selecting the replenishment policy. We improve the *DFRP* in three ways:

1. Estimating the model parameters and fixed costs adjustments. We enhance the dynamic model to adjust the costs of picking from the forward area, the reserve area, and the replenishments with respect to the changes in each area's size. The forward area can be recognized from the reserve area in Figure 28. Obviously, having a forward area with more aisles will increase the picking cost from this area. c, c_1 and c_2 are then

$$c_1 = 2\frac{L}{2} + \frac{Zv}{2} = L + \frac{Zv}{2} \tag{68}$$

$$c_2 = 2\frac{L}{2} + \frac{Zu}{2} = L + \frac{Zu}{2} \tag{69}$$

$$c = \frac{L}{2} + \frac{Zu}{2} + \frac{Z(u+v)}{2} + \frac{L}{2} + \frac{Zv}{2} = L + Z(u+v) \tag{70}$$

Where

L: Length of each picking aisle

v: Number of picking aisles in the forward area

u: Number of picking aisles in the reserve area

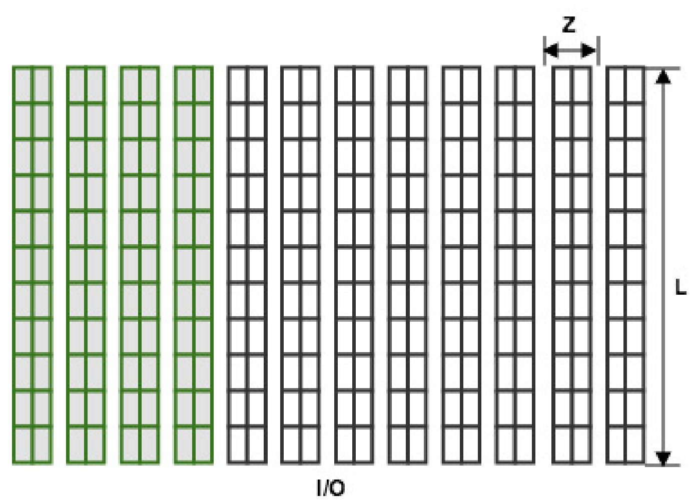

Figure 28. Warehouse layout (Forward area: green aisles; Reserve area: black aisles)

The picking cost from the forward/reserve areas in equations 68 and 69 is the sum of cross aisle travel and picking aisle travel. The replenishment cost in equation 70 includes three costs: the retrieval cost from the reserve area, the travel cost from the reserve to the forward area, and the storage cost in the forward area. The replenishment cost will not change, when enlarging or shortening the forward area, since it is relative to the total number of aisles $(u + v)$.

2. Moving the slow movers to the reserve area. The opportunity of moving the slow movers nested in the forward area for a long time, due to the lack of sufficient orders in the dynamic problem, is addressed. These moving costs are equal to the replenishment cost. If we have the perfect information about future orders, this problem is automatically solved because we always replenish the exact amount of the future demand and will have the slot empty at some point in the future without any move. Nevertheless, it is not true for the dynamic model to use the forecasted data. The decisions about moving the slow movers to the reserve area determine which SKUs should leave the forward area.

3. Applying different types of replenishment. One limitation of the generic *DFRP* discussed in section B is when the inventory level of a certain slot gets very low and the actual order is greater than the inventory in the forward area. Three options are available for order fulfillment in this case:

a) Move the item to the reserve area and pick the whole order from the reserve area $(c + c_2)$.

b) Replenish the rest of the order (or more) in the forward area and pick the order from the forward area $(c + c_1)$.

c) Leave the low inventory item in the forward area and pick the whole order from the reserve area $(c_2 + c_w)$, where c_w is a waiting time cost to get an order quantity equal to the inventory and have the slot empty.

Option a is always more expensive than option b, so the model will not suggest that. Since moving the item to the reserve area results in a cost equal to the replenishment cost, the model rarely suggests the move to the reserve area. It inclines to wait to get an order quantity equal to the inventory level, which makes the slot empty for free. However, this waiting time in option c – leaving some slots with a low inventory level in the forward area to get the order quantity exactly the same as the low inventory– postpones generating the pick savings from the forward area, which is not efficient. We will show in the following that the size of the forward area impacts these decisions. We will address this issue by discussing different replenishment policies.

F Replenishment policies

In section B, we addressed a general form of the dynamic model by having the restock quantity as the decision variable. In this section, we will elaborate on the different replenishment strategies, the quantity replenishment (model 1), and the full replenishment of a slot (model 2.) Each of these models contains sub-models. Note that model 2 is not a special case of model 1, since the dynamic model is not just a one-time run model for the whole planning horizon and works with forecasted data. The demand input is updated in every t.

The dynamic model, which uses the forecasted data, is run at each t to make both the pick decisions from the forward or reserve area as well as the replenishment decisions. Therefore, the actual demand quantity and picks is prone to the forecast errors. We may not receive exactly the same orders as the forecasted ones. As a result, some SKUs may stay in the forward area with low level inventory.

One way to pick again from those low level inventory slots is by replenishing them up to full or less than full with the same SKU, even if they are not empty.

Another way to remove the remaining inventory is by moving them to the reserve area anytime, with a cost equal to the replenishment cost. Therefore, moving the low inventory from the forward area to the reserve area and then restocking the emptied slot with another SKU costs twice as much as the replenishment cost.

1 Quantity replenishment (M^1)

The replenishment quantity is an integer decision variable in model 1 (M^1). Model M^1_{LH} considers a limited horizon aiming to reduce the problem size. If an SKU is selected for the forward area in this strategy, there is the risk of restocking an amount equal to the demand of the limited forecast horizon and losing the chance of a full replenishment of the slot. Consequently, the initial inventory of the slot in the next run of the dynamic model would be less than the full replenishment strategy.

The model considering the whole horizon ($T = 21$) is complex and will not deliver the solutions in a reasonable time. In the unlimited horizon model, M^1_{ULH}, we enlarge the period t by aggregating the forecast demand data of 3 consecutive days ($t \in \{1, ..., \frac{21}{3}\}$.) As a result, the final number of periods ahead (T) will be reduced from 21 to 7. Although the shorter periods result in more prompt responses and decisions about the picks, replenishments, and moves, it is not computationally efficient when addressing the whole horizon. One limitation of expanding the period is that the decisions about the forward area are released every 3 days, not daily.

Model 1 is similar to the general model introduced in section B but has the option of moving the slow movers to the reserve area when required.

$$M^1: \quad \text{Min} \quad C_1 = c_1 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it} x_{it} + c_2 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it}(1 - x_{it}) + c \sum_{i=1}^{N} \sum_{t=1}^{T} (y_{it} + w_{it})$$

subject to:

$$-I_{it} + I_{i,t-1} + R_{it} - d_{it}x_{it} - s^f_{it} = 0 \qquad \forall i, t \geq 2 \tag{71}$$

$$\sum_{i=1}^{N} n_{it} \leq \eta \qquad \forall t \tag{72}$$

$$R_{it} \leq \eta e_i y_{it} \qquad \forall i, t \tag{73}$$

$$n_{it} \geq \frac{I_{i,t-1} + R_{it} - s^f_{it}}{e_i} \qquad \forall i, t \tag{74}$$

$$s^f_{it} \leq \eta e_i w_{it} \qquad \forall i, t \tag{75}$$

$$R_{it}, I_{it}, s^f_{it} \geq 0 \qquad \forall i, t \tag{76}$$

$$x_{it}, y_{it}, w_{it} \in \{0, 1\} \qquad \forall i, t \tag{77}$$

$$n_{it} \in \{0, 1, 2, 3, ...\} \qquad \forall i, t \tag{78}$$

Where

Parameters:

N: Number of SKUs (i=1,2,...,N)

T: Number of periods (t=1,2,...,T)

c: Restock cost

c_1: Cost of picking from the forward area

c_2: Cost of picking from the reserve area

e_i: Number of units of SKU i that can be stored in one slot

d_{it}: Demand of SKU i at time t

p_{it}: Number of picks of SKU i at time t

η: Total number of slots in the forward area

Decision Variables:

x_{it}: 1 if SKU i is picked from the forward area at time t; 0 otherwise

y_{it}: 1 if SKU i is restocked at time t; 0 otherwise

I_{it}: Inventory of SKU i in the forward area at the end of time t

R_{it}: Number of units of SKU i that are restocked at time t

n_{it}: Number of slots occupied by SKU i at time t, $n_{it} \in \{0, 1, 2, 3, ...\}$

s^f_{it}: Units of SKU i that are moved from the forward to the reserve area at time t.

w_{it}: 1, if SKU i is moved from the forward area to the reserve area at time t; 0 otherwise.

Constraint 75 makes the binary variable of the move from the forward to the reserve area, w_{it}, equal to 1 if any units of SKU i are moved to the reserve area at time t ($s^f_{it} > 0$).

2 Full replenishment (M^2)

Model 2 restocks the full allocated slot(s). If U_{it} slot(s) are given to the SKU i at time t, the replenishment quantity will be $e_i U_{it}$. We investigate three different strategies for model 2, named M^2_a, M^2_b, and M^2_c.

M^2_a: Can replenish up to full if the slot(s) are empty.

M^2_b: Can replenish up to full if the slot(s) are empty and can replenish partially if the slot(s) are non-empty. The partial replenishment can make the slot less than full or full.

M^2_c: Can replenish full any time.

Figure 29 summarizes the replenishment policies. The three MIP models defined for model 2 are demonstrated below:

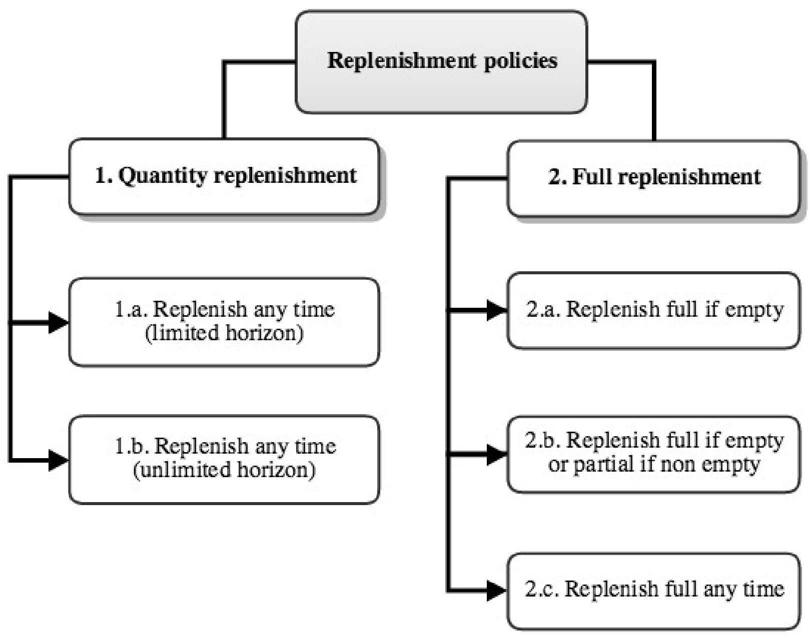

Figure 29. Replenishment policies

Replenish up to full if empty (M_a^2)

$$\text{Min} \quad C_1 = c_1 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it} x_{it} + c_2 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it}(1 - x_{it}) + c \sum_{i=1}^{N} \sum_{t=1}^{T} (y_{it} + w_{it})$$

subject to:

$$-I_{it} + I_{i,t-1} + e_i U_{it} - d_{it} x_{it} - s_{it}^f = 0 \qquad \forall i, t \geq 2 \qquad (79)$$

$$\sum_{i=1}^{N} n_{it} \leq \eta \qquad \forall t \qquad (80)$$

$$U_{it} \leq \eta y_{it} \qquad \forall i, t \qquad (81)$$

$$n_{it} \geq \frac{I_{i,t-1} + e_i U_{it} - s_{it}^f}{e_i} \qquad \forall i, t \qquad (82)$$

$$s_{it}^f \leq \eta e_i w_{it} \qquad \forall i, t \qquad (83)$$

$$I_{it}, s_{it}^f \geq 0 \qquad \forall i, t \qquad (84)$$

$$x_{it}, y_{it}, w_{it} \in \{0, 1\} \qquad \forall i, t \qquad (85)$$

$$n_{it}, U_{it} \in \{0, 1, 2, 3, ...\} \qquad \forall i, t \qquad (86)$$

Constraint 81 makes the replenishment binary variable y_{it} equal to 1, if U_{it} slots are given to SKU i at time t. The number of replenishment units for SKU i at time t is $e_i U_{it}$ in constraint 82.

Replenish up to full if empty or partially if not empty (M_b^2)

$$\text{Min} \quad C_1 = c_1 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it} x_{it} + c_2 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it}(1 - x_{it}) + c \sum_{i=1}^{N} \sum_{t=1}^{T} (y_{it} + w_{it} + h_{it})$$

subject to:

$$-I_{it} + I_{i,t-1} + e_i U_{it} - d_{it} x_{it} - s_{it}^f + s_{it}^r = 0 \qquad \forall i, t \geq 2 \qquad (87)$$

$$\sum_{i=1}^{N} n_{it} \leq \eta \qquad \forall t \qquad (88)$$

$$U_{it} \leq \eta y_{it} \qquad \forall i, t \qquad (89)$$

$$n_{it} \geq \frac{I_{i,t-1} + e_i U_{it} - s_{it}^f + s_{it}^r}{e_i} \qquad \forall i, t \qquad (90)$$

$$s_{it}^f \leq \eta e_i w_{it} \qquad \forall i, t \qquad (91)$$

$$s_{it}^r \leq \eta e_i h_{it} \qquad \forall i, t \qquad (92)$$

$$s_{it}^r \leq e_i - I_{i,t-1} \qquad \forall i, t \geq 2 \qquad (93)$$

$$I_{i,t-1} \geq h_{it} \qquad \forall i, t \geq 2 \qquad (94)$$

$$I_{it}, s_{it}^f, s_{it}^r \geq 0 \qquad \forall i, t \qquad (95)$$

$$x_{it}, y_{it}, w_{it}, h_{it} \in \{0, 1\} \qquad \forall i, t \qquad (96)$$

$$n_{it}, U_{it} \in \{0, 1, 2, 3, ...\} \qquad \forall i, t \qquad (97)$$

Where

s_{it}^r: Units of SKU i that is restocked in the non-empty slot containing SKU i at time t (partial replenishment units of SKU i at time t.)

h_{it}: 1 if s_{it}^r units of SKU i are restocked in the non-empty slot containing SKU i at time t; 0 otherwise.

Since model M_b^2 provides both partial and full replenishment opportunities for the products of the forward area, n_{it} is corresponded with both variables U_{it} and s_{it}^r in constraint 90. U_{it} covers the full replensihments and s_{it}^r accounts for the partial ones. Constraint 92 makes the binary variable h_{it} 1, if any partial replenishment occurs. Constraint 93 assures that the units of partial replenishment are less than or equal to the available capacity of the non-empty slot containing SKU i at time t. Constraint

94 controls the partial replenishments and makes sure that they are executed for only non-empty slots.

Replenish up to full any time (M_c^2)

$$\text{Min} \quad C_1 = c_1 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it} x_{it} + c_2 \sum_{i=1}^{N} \sum_{t=1}^{T} p_{it}(1 - x_{it}) + c \sum_{i=1}^{N} \sum_{t=1}^{T} (y_{it} + w_{it} + h_{it})$$

subject to:

$$-I_{it} + I_{i,t-1} + e_i U_{it} - d_{it} x_{it} - s_{it}^f + s_{it}^r = 0 \qquad \forall i, t \geq 2 \qquad (98)$$

$$\sum_{i=1}^{N} n_{it} \leq \eta \qquad \forall t \qquad (99)$$

$$U_{it} \leq \eta y_{it} \qquad \forall i, t \qquad (100)$$

$$n_{it} \geq \frac{I_{i,t-1} + e_i U_{it} - s_{it}^f + s_{it}^r}{e_i} \qquad \forall i, t \qquad (101)$$

$$s_{it}^f \leq \eta e_i w_{it} \qquad \forall i, t \qquad (102)$$

$$s_{it}^r \leq \eta e_i h_{it} \qquad \forall i, t \qquad (103)$$

$$s_{it}^r \leq e_i - I_{i,t-1} \qquad \forall i, t \geq 2 \qquad (104)$$

$$s_{it}^r \geq e_i - I_{i,t-1} - e_i(1 - h_{it}) \qquad \forall i, t \geq 2 \qquad (105)$$

$$I_{it}, s_{it}^f, s_{it}^r \geq 0 \qquad \forall i, t \qquad (106)$$

$$x_{it}, y_{it}, w_{it}, h_{it} \in \{0, 1\} \qquad \forall i, t \qquad (107)$$

$$n_{it}, U_{it} \in \{0, 1, 2, 3, ...\} \qquad \forall i, t \qquad (108)$$

Model M_c^2 can replenish fully at any time. This model is similar to model M_b^2, except in one constraint: model M_c^2 does not need constraint 94. Instead, we added constraint 105 to link to constraint 104 and replenish an amount exactly equal to the available capacity of the slot ($e_i - I_{i,t-1}$) when a partial replenishment is required (h_{it}=1).

G Heuristics for the dynamic forward-reserve problem ($T.P$)

In this section, we will investigate a simple threshold policy that performs almost as well as the dynamic MIP model M_c^2 in section F. The problem gets significantly more computationally expensive for large amounts of data. The suggested intuitive heuristic $T.P.$ delivers a near optimal solution within a reasonable computing time as well as an acceptable performance consuming the sensible number of SKUs and size of the forward area in practice. It is assumed that the replenishments can be made over time with a negligible operational time and when the pick list for the current period ($t = 0$) is known. The demand data for the next ω period is forecasted.

$T.P.$ uses heuristic G_2 explained in chapter II for an SKU assignment and slot allocation of the forward area. In the case of the initial empty forward area, we first run the G_2 to get the initial layout and slot allocation. In each period, the inventory level of the slots drop based on the SKU demand in that period. If the inventory level of the SKU is zero or below, we run G_2 to decide the SKU re-assignment in the forward area. If re-assigned, it will be replenished and the inventory level gets updated. If not, the slot gets empty by pick. Note that the SKUs that are available in the forward area with $I_{it} > 0$ are excluded from the candidate set of SKUs imported to G_2. Finally, all replenishments and picks from the forward and reserve areas counted for the total cost calculations.

$T.P.$ is an algorithm used simultaneously for a dynamic SKU and an assignment of the forward area. In other words, this heuristic not only keeps the currency of the forward area by updating the set of assigned SKUs to the fast picking area, but also adjusts the allocated slots to them. The pseudo code for the heuristic $T.P.$ that updates both the SKU assignment and the discrete space allocation in the forward area can be found in the Appendix.

Examining the effect of a slot allocation on the total cost, algorithm $T.P.'$ is developed for the dynamic slotting problem that only considers the assignment of SKUs to the forward area, not the space allocation. Heuristic $T.P.'$ assumes one slot per SKU in the forward area and is based on the following four steps:

Dynamic SKU assignment in the forward area (heuristic $T.P.'$)

Input: The generic MIP DFRP model's parameters.

Output: The dynamic SKU assignment in the forward area over time.

For ($t = 1$ to T)

1. Find empty slots. Find e, the total number of the empty slots and the slots that become empty by the order picking at time t.

2. Sort. Rank the SKUs by the labor efficiency of SKU it at time t (le_{it}) using the forecast demand data for the forecasting window, where $f_{4it} = \frac{d_{it}}{b_i}, le_{it} = \frac{p_{it}}{\sqrt{f_{4it}}}$.

3. Update. Update the list by excluding those SKUs that still have inventory in the forward area, even after order picking at time t.

4. Assign SKUs to the empty slots. Assign the first e SKUs of the list to the emptied slots. Each assigned SKU gets one slot.

EndFor

H Model Validation and Numerical Discussions

In this section, we first compare the static model of an SKU assignment and a discrete space allocation of the forward area (G_2 in chapter II) to the most similar dynamic model, which is M_2^a. The solution of the problem with perfect information about the future demand demonstrates the resulted gap due to the demand forecasting process. The effect of an activity distribution of items on the total saving by the dynamic model is addressed. Then, we compare the performance of $DFRP$ to the different replenishment strategies in Figure 29; a static FRP and two threshold policies developed in section G. Finally, we examine how the volatility of the demand patterns impact the computational results using a variety of data sets.

1 Comparison of the static and dynamic models using the forecast demand data

The static and dynamic model picking and replenishment costs are presented in Table 21. While the demand data of the first period ($t = 1$) is assumed as known, the demand data for the rest of the planning horizon is forecasted and updated at each run t. Therefore, the pick and replenishment decisions of the first period using the known demand and picks data are actual, causing the inventory level of slots to drop by the actual demand values, but those for $t > 1$ provide the planning insights. Note that all $DFRP$ results are associated with forecasted data, unless we mention the PI for the perfect information.

It is observed in Table 21 that the dynamic model always outperforms the static model. The cost improvement by the dynamic model is greater when the static model is interrupted more frequently (four updates or every four days, $T = 20$). The reason is that some SKUs leave the forward area in each update before finishing their minimum payback period (T_i'). T_i' is the minimum time that SKU i should stay in the forward area to make a profit. The smallest number that satisfies non-equality 110 below is T_i'.

$$c + c_1 \sum_{i=1}^{T_i'} p_{it} < c_2 \sum_{i=1}^{T_i'} p_{it} \quad \forall t \tag{109}$$

$$\sum_{i=1}^{T_i'} p_{it} > \frac{c}{c_2 - c_1} \quad \forall t \tag{110}$$

T_i's values are not the same for every SKU. Thus, re-layouting the forward area in a certain interval will be disruptive for the SKUs, which have been stored for less than T_i' periods after their last replenishment in the forward area and are forced to leave the forward area during update times.

TABLE 21

Total cost and savings (%) obtained from static(S) and dynamic model M_a^2.

	C_S	$C_{M_a^2}(PI)$	$\%Imp_{M_a^2(PI)\to S}$	$C_{M_a^2}$	$\%Imp_{M_a^2\to S}$
No update	1643452	1392004	15.30	1403328	14.61
One update	1625704		14.38		13.68
Three updates	1756932		20.77		20.13
Four updates	1822404		23.62		23.00

In order to study the performance of the dynamic model for a different activity level of the items in the facility, we generated the experiments listed in Table 22. Active items are those items that are picked frequently. The percentage of the fast movers ranges from 10% to 85%. The results of this Table shows that active warehouses with large percentage of fast movers can benefit more from the dynamic model rather than the slow warehouses, which contains a large fraction of slow movers. As expected, a saving of 9.72%, which was obtained from the dynamic model over the static model, is still considerable for the inactive warehouse in our designed experiments.

2 Comparison of different replenishment strategies of the dynamic model, static model and threshold policies

In this section, we first provide our mechanism for constructing the data set used in our comparisons. We drastically reduce the size of our data set by applying the ABC analysis. The slow movers are excluded from the candidate set of SKUs for

TABLE 22

Cost comparisons of the activity distribution of items (M_a^2)

No. of SKUs	No. of slow movers	% fast movers	% $Imp_{(M_a^2 \rightarrow static)}$
5000	4500	10	9.72
2500	2000	20	11.94
1666	1166	30	12.92
1250	750	40	13.48
1000	500	50	13.84
600	100	85	14.45

the forward area by this method.

While a large portion of the SKUs in the warehouse are slow movers, a small portion accounts for most of the picking activities and makes up a large percentage of orders. We need to know the fast movers, which can be candidates in *DFRP* analysis. Traditionally, the ABC analysis classifies the SKUs based on their activities in three groups: a small fraction of fast movers, medium movers, and a large fraction of slow movers.

Wild (2007) suggests that the breakdown of ABC classes as 10% of items represents class A, 20% of items represents class B, and 70% of items represents class C. Hausman et al. (1976) propose a continuous demand model for representing ABC analysis.

The small fraction of SKUs (the fast movers) matters in making decisions for the forward area. The intermittent demand trend (trend 8) discussed in chapter IV, which contains many zeros in the demand profile, forms a large portion of slow movers.

Figure 30 shows the demand curve of the total 5000 SKUs in the warehouse. We ranked the items in decreasing order based on their contribution to the total

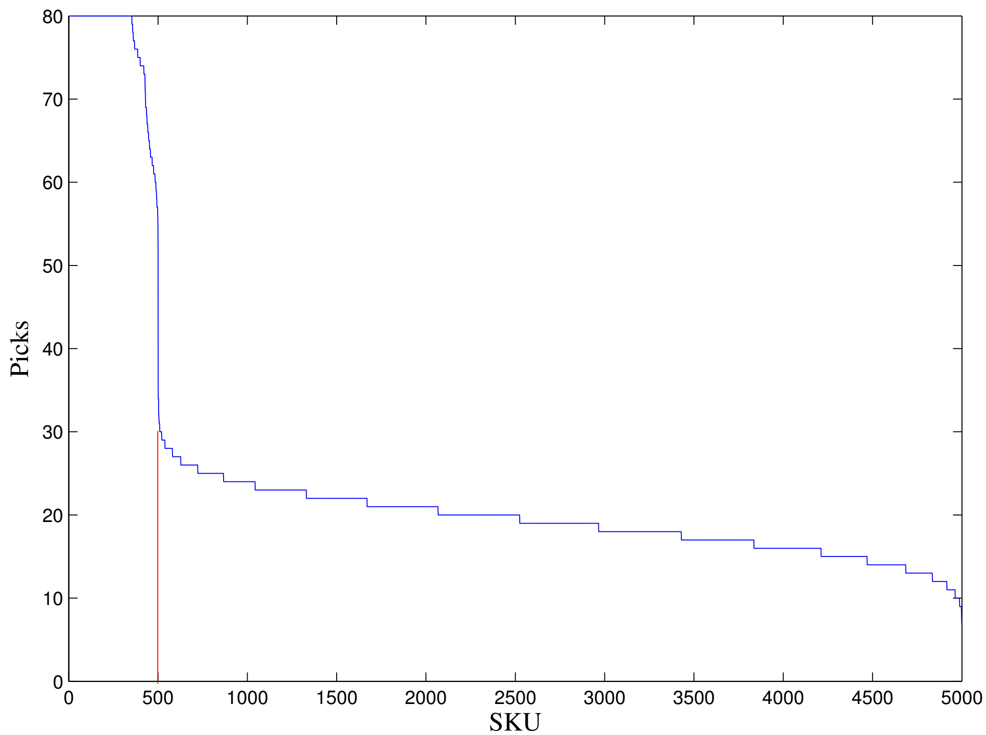

Figure 30. ABC analysis

demand. The SKUs up to the red line are considered in *DFRP*. Since there is no fixed threshold for each class, depending on the size of the forward area, the red line can move left or right based on objective criteria in the dynamic assignment and allocation problem to consider less or more SKUs as *DFRP* input. The forward area in the following numerical examples has 320 slots. Respecting the recommended break down for ABC classes, we truncated 500 of the fastest movers (10% of all SKUs) as the candidates to be stored in the forward area. This 10% accounts for 68.73% of all picks in our data set.

We compare the total costs of the discussed replenishment strategies for the dynamic model, along with the dynamic, static, and dynamic heuristics $T.P.$ and $T.P.'$ comparisons. Table 23 contains the results of our computational study on the *DFRP* with different replenishment policies. The full replenishment models (M^2) always outperform the quantity replenishment models (M^1). The quantity replenishment models have a higher number of partial replenishments compared to the full replenishment models. Part of the partial replenishments of M^1 models is due to vacating the slots based on the forecasted demand data. However, the actual demand may not be exactly the same as the forecasted demand and so the slots cannot get empty. This risk does not concern the M_a^2 and M_c^2 models, where only full replenishments are allowed.

M_{LH}^1 model considers a limited horizon, where the replenishment quantity may be less than the full slot. In this case, the chance of restocking the whole slot is missed. Consequently, the number of replenishments rises, but still less than the static model. The unlimited horizon in M_{ULH}^1 does not not fix this problem. Similarly, it may fill a portion of a slot, even having the whole horizon forecast, in the hopes of vacating a slot. However, slot vacating may not come true by the actual demand. Another

TABLE 23

Results for different replenishment policies of *DFRP*

	Static	M^1_{LH}	M^1_{ULH}	M^2_a (PI)	M^2_a	M^2_b	M^2_c	$T.P.$	$T.P.'$
Full Replens	1659	-	-	674	577	350	350	1020	1226
Partial replens	-	1134	691	-	-	667	667	-	-
Move to reserve	37	0	66	0	1	0	0	222	213
Replens&moves	1696	1134	757	674	578	1017	1017	1242	1439
Forward picks	5396	5061	4103	4351	4028	4900	4991	5269	5842
Reserve picks	4277	4612	5570	5322	5645	4773	4682	4404	3831
Total cost	1625704	1461724	1460248	1392004	1403328	1441092	1378604	1471980	1462036
% Imp. Over static	-	10.09	10.18	14.38	13.68	11.36	15.20	9.46	10.07

limitation of model M^1_{ULH} is that the length of period t is 3 days (t in the other models is one day) due to the computational complexity reduction. So, the delivered solutions are corresponded to the 3 days demand data, not the daily demand. In other words, the decisions about the forward area can only be updated every 3 days. It is observed that M^1_{ULH} generates the largest number of moves to the reserve area.

Among all three replenishment policies defined for the full slot replenishment in M^2, model M^2_b, which allows the partial replenishment of a non-empty slot along the operations, suffers from the aforementioned limitations of the partial restocking, including the high number of replenishment and so a greater total cost. Compared to M^2_a, which does not have the option of a partial replenishment of the non-empty slots, M^2_b allows more picks from the forward area rather than the reserve area, but the number of moves and replenishments in M^2_b is 43% higher than M^2_a. Therefore, it is suggested to not partially replenish the forward area slots any time.

Model M^2_c, which fully replenish the empty slot(s) and also have the option of replenishing the non-empty slots up to full capacity, e_i, is the best strategy with the lowest cost among all of the *DFRP*'s replenishment policies. Although the number of replenishments and moves in M^2_c is not minimum among all other models, this model has the minimum number of picking and replenishment costs. The cost of M_c^2 is even less than the $M_a^2(PI)$ cost because it can replenish *any* time up to full while $M_a^2(PI)$ can replenish full only when the slot is empty. It is worth noting that although M_c^2 is more constrained than the quantity replenishment models (M^1), its lower total cost during the planning horizon after multiple runs of M_c^2 with the updated forecasted demand data at each t justifies the fitness of this model in the dynamic slotting (See Figure 25). We are not comparing the one-time run models with fixed input data. The models are fed with the varying demand and pick data at each t and the models' decisions are changed with the updated inputs at each t.

It can also be referred from Table 23 that the dynamic threshold policies $T.P.$ and $T.P.'$ are almost as good as the dynamic model with partial replenishment. Space allocation by $T.P.$ heuristic makes 0.61% more savings than allocating the same amount of space (one slot per assigned SKU) in the $T.P.'$ heuristic.

The number of moves to the reserve area depends on the size of the forward area. Figure 31 shows that the smaller forward areas experience a higher number of moves from the forward area to the reserve area due to the open space for the candidate SKUs in the forward area.

We investigate the dynamic slot allocation behavior for different replenishment policies. SL_i is the set of allocated slots to the SKU i during the planning horizon, $SL_i = \{n_{i1}, n_{i2}, n_{i3}, ..., n_{iT}\}$, where T is the length of the planning horizon and $n_{it} \geq 0$. We define parameter K_i as the number of unique values in SL_i. Higher values of parameter K_i show that the SKU experiences a more diverse number of allocated slots in the forward area. For example, if an SKU is given, sometimes 2 slots, other times 3 slots, in the forward area, SL_i will have two unique values ($K_i = 2$). Note that n_{it} can be zero, which means that SKU i has not been in the forward area at time

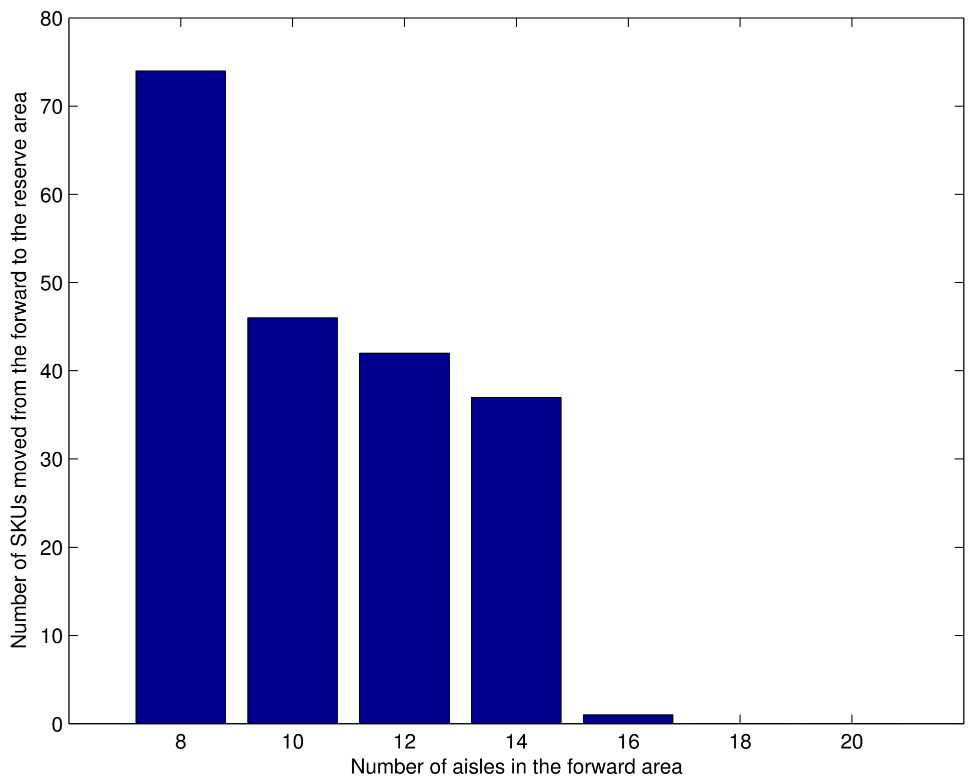

Figure 31. Number of moves from the forward area to reserve area

t ($n_{it} = 0$). Our example's solutions reveal that the assigned SKUs to the forward area never receive more than 3 slots in all of the strategies ($K_i \leq 4$). We split $K_i = 1$ to two cases:

$$K_i = \begin{cases} 1' & \text{if all } n_{it} = 0 \\ 1'' & \text{otherwise} \end{cases}$$

$K_i = 1'$ refers to the SKUs that are always picked from the reserve area and so their allocated slots are always zero. $K_i = 1''$ refers to the SKUs that always have a fixed number of allocated slot(s) in the forward area.

Table 24 returns the number of SKUs with different K_i for the static model with one update and the dynamic model with different replenishment strategies. It is observed that the static model excludes the greatest number of SKUs for being stored in the forward area. Compared to the full slot allocation models, M^2, the partial replenishment models, M^1, face a higher number of SKUs with $K_i \geq 3$, meaning that more allocated slots to the SKUs in the forward area and a higher variability in slot allocation as well.

TABLE 24

No. of SKUs with different values of K_i

K_i	Static	M^1_{LH}	M^1_{ULH}	M^2_a (PI)	M^2_a	M^2_b	M^2_c	$T.P.$	$T.P.'$
$1'$	200	137	177	138	142	160	138	51	49
$1''$	231	140	84	117	122	285	201	138	169
2	69	139	197	211	212	54	134	285	282
3	0	80	40	34	24	1	27	26	0
4	0	4	2	0	0	0	0	0	0

Figure 32 displays the stacked bar graph of distribution of items, based on K_i, for different models. Each bar is multicolored, with colors corresponding to K_i and showing the relative contribution that different values of K_i make to the total number of SKUs. This figure shows that M_b^2, which can partially replenish the non-empty slots, is the most similar case to the static model regarding the slot allocation. While M_a^2 and M_b^2 has a large portion of SKUs in storage mode 2 and 1, respectively, M_c^2 –the best replenishment strategy using the forecasted demand data– has the SKUs more evenly distributed among these two storage modes. In dynamic slotting strategies, on average 39% of the SKUs experience more than one storage mode ($K_i \geq 2$) in the forward area. However, updating the forward area periodically in the static model changes the storage mode of only 6% of the SKUs.

Although the static model has the option of periodically updating the forward area, Figure 32 also shows that the K_i values of this model do not exceed 2, which shows the less variability and flexibility in the number of allocated slots to the SKUs.

3 Volatility

What industries can benefit most from implementing the dynamic slotting? Is the dynamic model more effective in high volatile periods? Demand volatility is a reality in the logistics industry. The dynamic slotting model can alleviate the adverse effects of the demand volatility on the decisions about the forward area over time. Given the historical demand data, we aim to find out in what periods of the year a warehouse will benefit more from the dynamic approach compared to the static strategy. It is valuable to assess the effectiveness of the dynamic model in two situations:

1. The demand trends of the majority of SKUs in a warehouse is normal (the first demand trend defined in chapter IV) or,

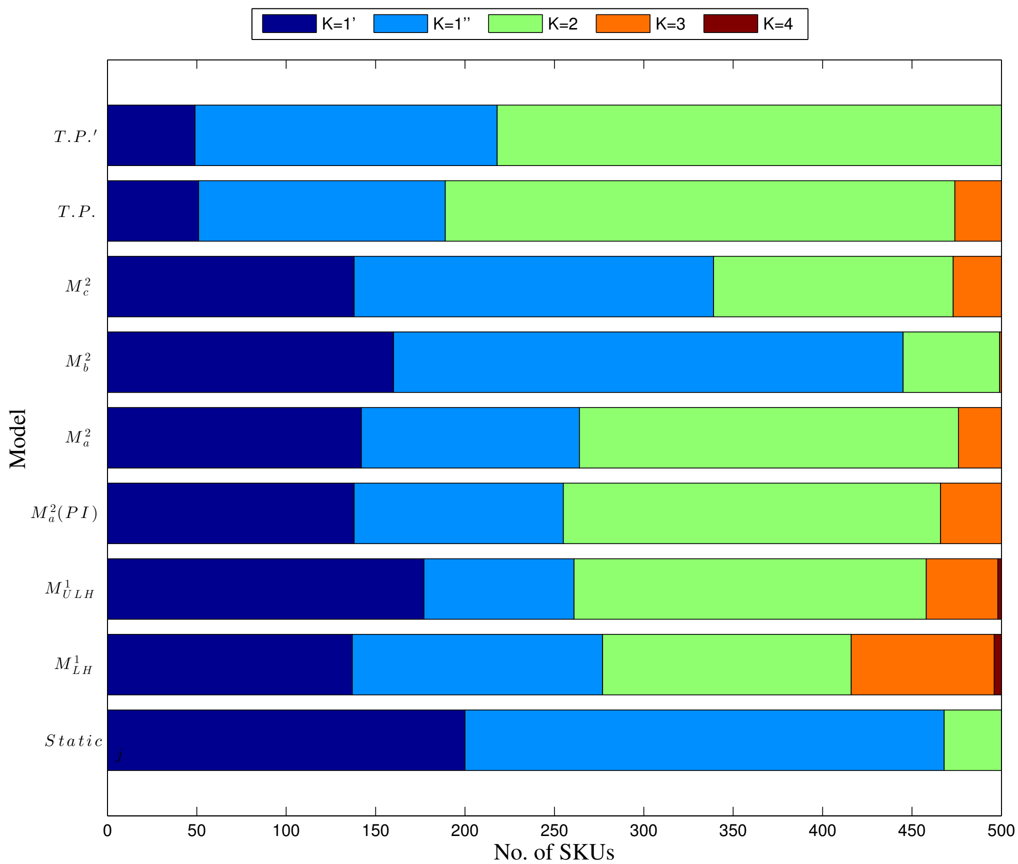

Figure 32. Distribution of SKUs with different values of K_i

2. The majority of SKUs shows a high level of volatility in their demand trends (the second to seventh demand patterns defined in chapter IV)

The SKUs' demand trends are statistically equivalent to the time series. The volatility index discussed in this section does not represent the variation within the demand trends. It is also different from the beta factor in finance, which measures the stock's volatility over time in relation to the overall market. Nevertheless, it aims to denote the non-similarity between the SKUs' demand trends over time. The volatility index contains the influence of the abnormal demand patterns. One simple way to compute this index is by detecting the change in the linear trend (slope) through the use of the end points from the time segments. The slope between the consecutive break points is a simple measure that can provide the information about the demand pattern variation. The high volatility index refers to the high variance between the slopes of SKUs' demand trends in each time segment.

Time-varying demand volatility implies that the volatility is itself subject to swings at various points in time. In other words, the order data reflects the high and low volatility periods over time. We first investigate an algorithm to represent the volatility of the demand patterns over time. Second, we discuss the saving levels resulted from the dynamic model, which corresponds to the different amount of volatility.

Assessing the time-varying demand volatility, we develop the following procedure:

Step 1. Shift the SKU demand curves towards the mean of the mean curve.

Step 2. $\forall t$, compute the slope, s_{it}, of the next T^s period for each SKU, using the following formula:

$$s_{it} = \frac{D_{i,t+T^s} - D_{it}}{T^s} \tag{111}$$

Step 3. Compute the variance of the slopes over time.

We provide the example below to clearly explain the algorithm for finding the volatility index.

Example: Our numerical example for the volatility index algorithm contains two SKUs. The second and third columns of Table 25 show the demand data for the SKU 1 and 2. The fourth column of this table forms the mean curve points, shown in Figure 33, based on the mean of D_1 and D_2 columns at each t. The demand curves in Figure 33 are shifted toward the mean curve, an amount equal to the difference between the mean of the mean curve (13.71) and mean of the demand curves. Thus, D_1 is shifted $15.58 - 13.71 = 1.87$ and D_2 is shifted $13.71 - 11.83 = 1.88$ toward the mean curve for all t. The decimal values of the numbers in this table are rounded. Next, the slopes of the shifted demands are obtained in columns S_{1t} and S_{2t}, using the equation 111 and the arbitrary value of $T^s = 2$. The last column of Table 25, which is corresponded to the curve in Figure 34, calculates the variance of the slopes at each t. These variance values are the volatility indexes of our example over time. Figure 34 shows that the demand data volatility of this data set is rising, which starts at period 4 and will go down until period 12 where the demand curves start to follow the smooth and stationary pattern again with no up/down trend.

The index that is obtained from step 3 of the algorithm is named the *Volatility index* in our analysis. Figures 35 and 36 display the variation of this index over time. When the majority of SKUs follows a normal demand pattern, the volatility index is close to zero. On the other hand, the volatility index rises in periods when the order

TABLE 25

Example for the volatility index calculation

Time	D_1	D_2	MD	SD_1	SD_2	S_{1t}	S_{2t}	Var
1	3	12	7.5	4.88	10.125	0.00	0.00	0.00
2	3	12	7.5	4.88	10.125	0.00	0.00	0.00
3	3	12	7.5	4.88	10.125	0.00	0.00	0.00
4	3	12	7.5	4.88	10.125	1.50	-0.50	2.00
5	3	12	7.5	4.88	10.125	2.50	-1.00	6.13
6	6	11	8.5	7.88	9.125	2.50	-1.00	6.13
7	8	10	9	9.88	8.125	3.00	-0.50	6.13
8	11	9	10	12.88	7.125	2.00	-1.00	4.50
9	14	9	11.5	15.88	7.125	1.00	-1.50	3.13
10	15	7	11	16.88	5.125	1.00	-1.00	2.00
11	16	6	11	17.88	4.125	0.50	-0.50	0.50
12	17	5	11	18.88	3.125	0.00	0.00	0.00
13	17	5	11	18.88	3.125	0.00	0.00	0.00
14	17	5	11	18.88	3.125	0.00	0.00	0.00
15	17	5	11	18.88	3.125	0.00	0.00	0.00
16	17	5	11	18.88	3.125	0.00	0.00	0.00
17	17	5	11	18.88	3.125	0.00	0.00	0.00
18	17	5	11	18.88	3.125	0.00	0.00	0.00
Mean	15.58	11.83	13.71					

D_1 = Demand of SKU 1,
D_2 = Demand of SKU 2,
MC = Mean of SKUs demand,
SD_1 = Shifted demand of SKU 1,
SD_2 = Shifted demand of SKU 1,
S_{1t} = Slope of SKU 1 with $T^s = 2$,
S_{2t} = Slope of SKU 2 with $T^s = 2$.
S_{2t} = Variance of slopes of SKU 1 and 2.

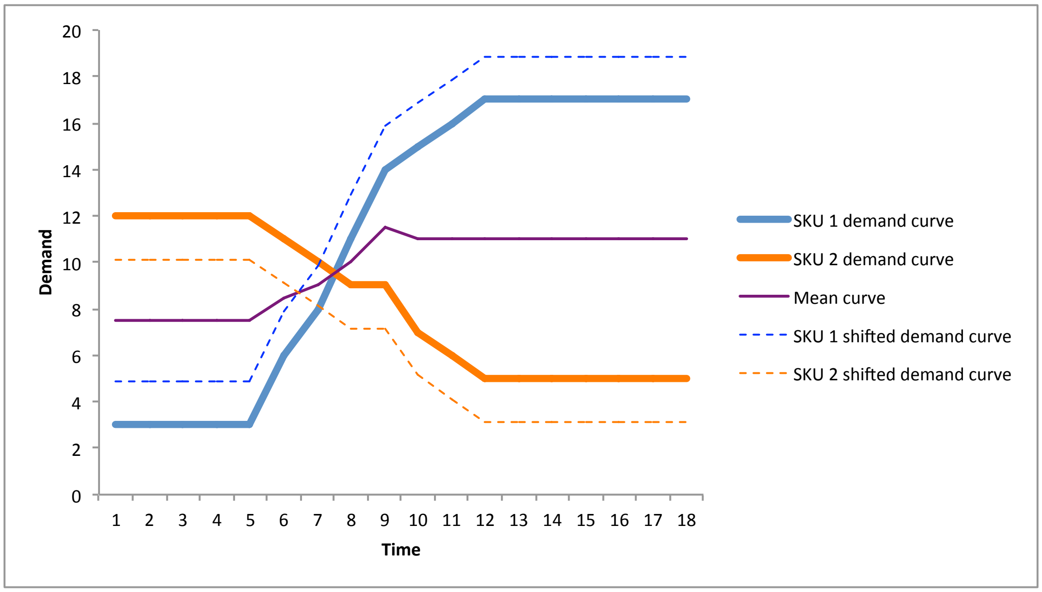

Figure 33. The curves corresponded to the Table 25

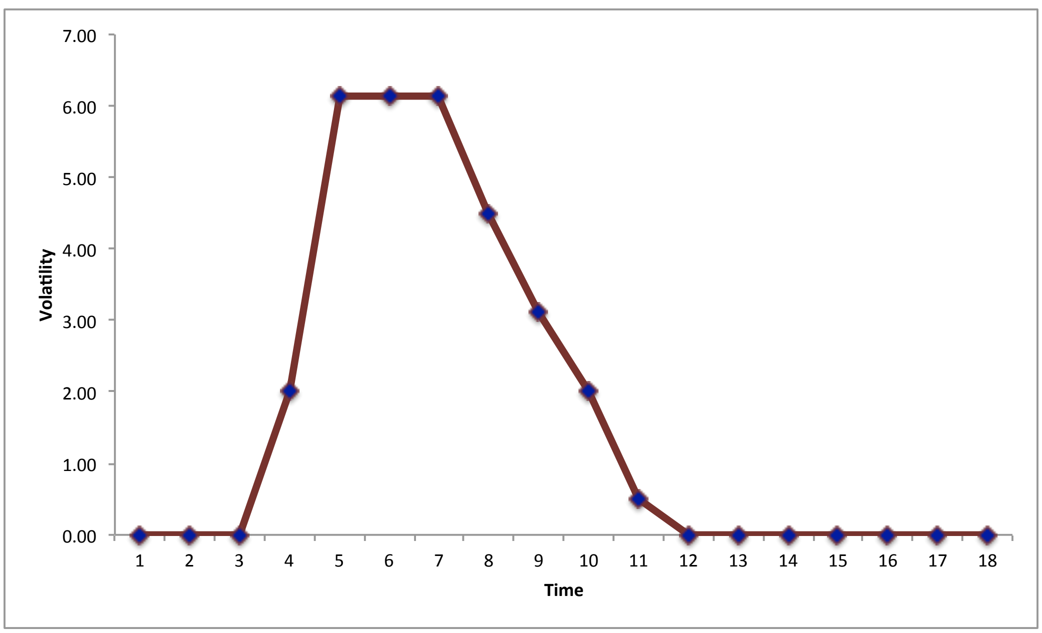

Figure 34. The volatility curve of the numerical example

data contains the variety of demand trends (up/down trends, up/down shifts, cyclic, systematic).

Investigating the improvement percentage of the dynamic over the static model, we simulated 7 order transaction data, as presented in Table 26. The portion of the SKUs with a normal demand pattern in the data set varies, as shown in the first column of this table. We observe that the dynamic model makes more profit when the demand volatility is higher. We will present the results of Table 26, using Figures 37 to 40.

Figure 37 represents the profits of the dynamic model for a different portion of the normal demand patterns in the data set. As this figure shows, the dynamic model generates more savings for high volatility cases (the lower portions of the normal demand trends in the data set). The maximum saving of the dynamic model over the static model in this example is 14% and it occurs when the portion of the SKUs with normal demand patterns is the lowest (1/7).

The main insight of this section is that once one has decided to use the dynamic re-slotting strategy, the profits are higher during the periods of the year when the SKUs' demand patterns encounter instability (e.g. seasonality, SKU growth, demand growth, promotions, competitor's offering, etc.). However, it does not mean that the stable time conveys no benefit, since the first data set with 100% normal data still results in 2.26% saving.

Figure 38 shows the number of picks from the forward and reserve areas versus the demand volatility for different models. Interestingly, the number of picks from the forward area in the dynamic model decreases when the demand volatility is higher. Therefore, the picks from the reserve area increases during high volatile periods. The static model is less sensitive in this regard; once an SKU is assigned to the forward

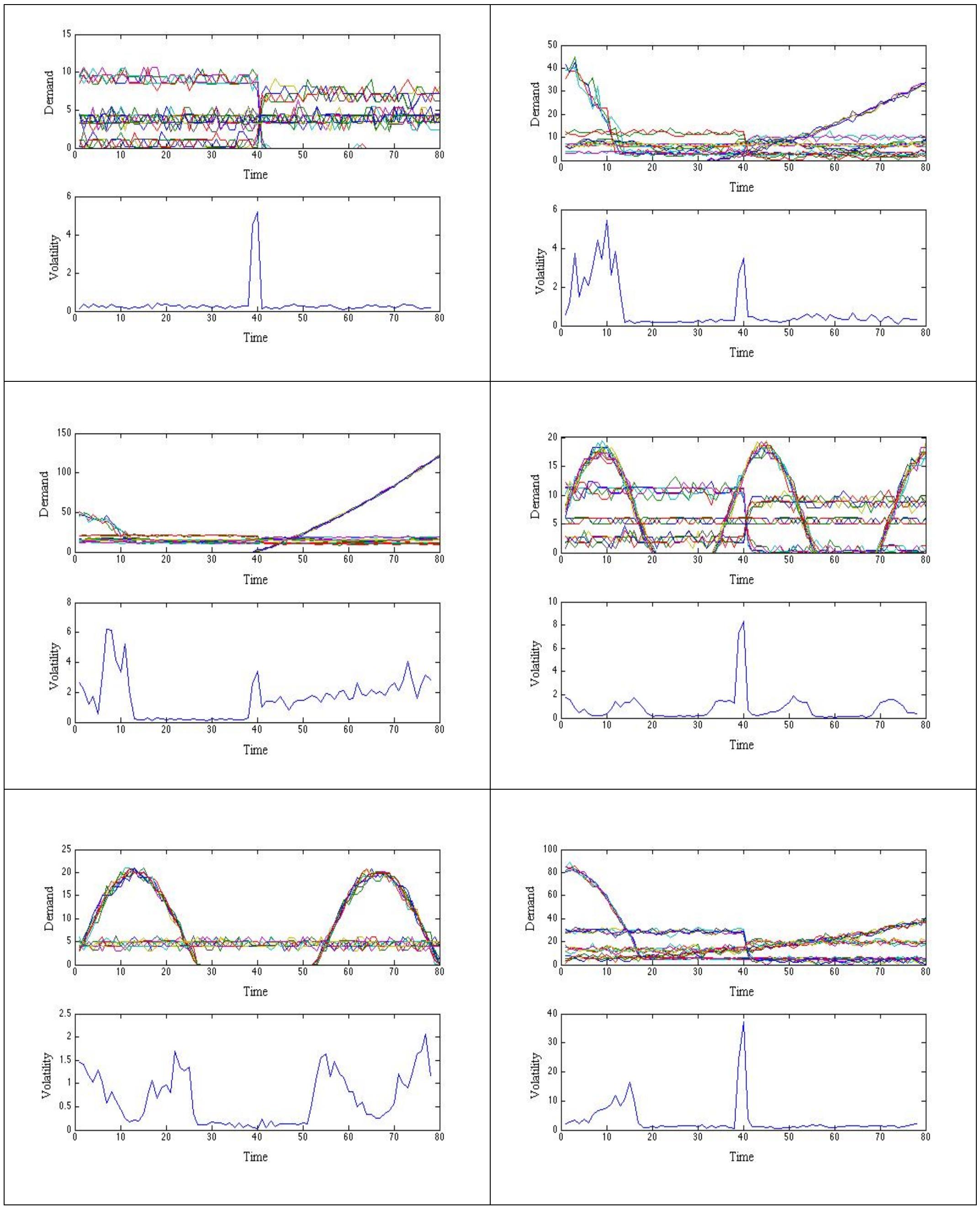

Figure 35. Volatility diagrams of simulated order data 1 through 6

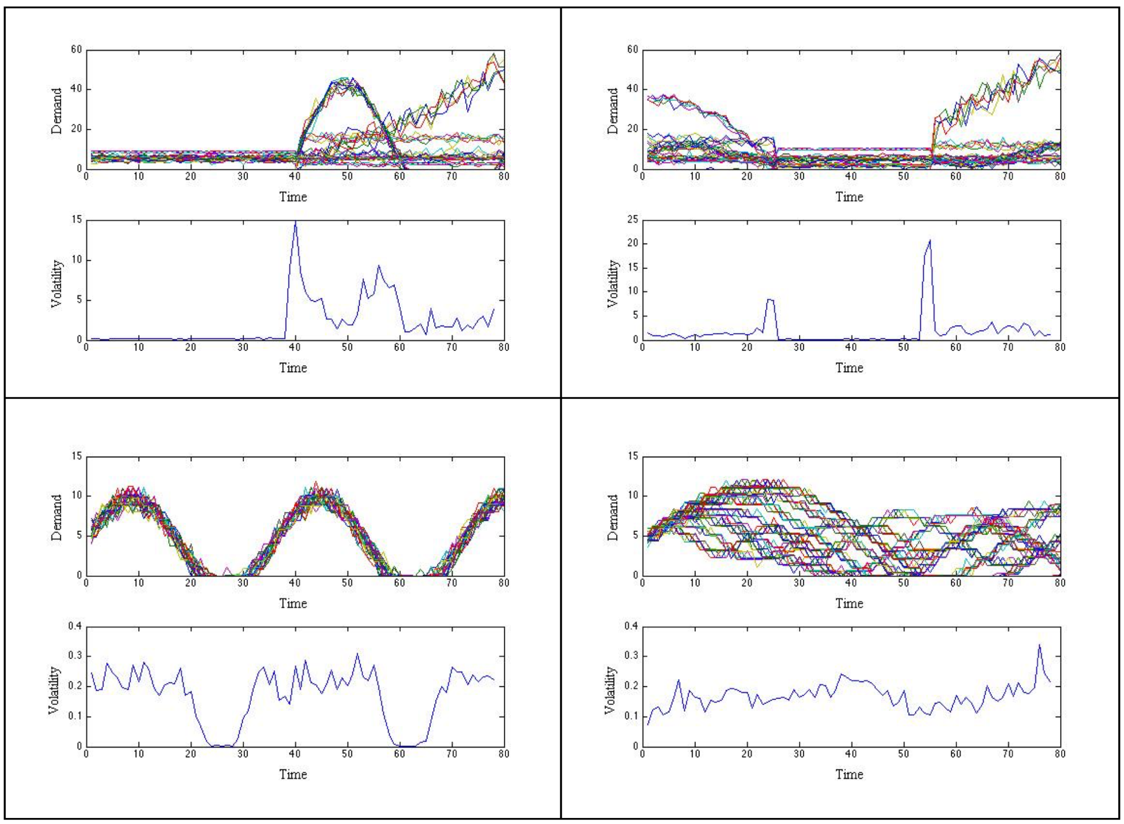

Figure 36. Volatility diagrams of simulated order data 7 through 10

TABLE 26

Comparison of the dynamic model, threshold policies $T.P.$ and $T.P.'$ and static model for warehouses with different portion of SKUs with normal demand pattern

Normal*	Model	FW. Picks	RES. Picks	Replens.	Moves to RES	Replens.+moves	Cost
100%	Dynamic	5330	3670	401	82	483	1050700
	$T.P.$	5582	3418	604	20	624	1066984
	$T.P.'$	5709	3291	670	33	703	1078208
	Static	5742	3258	674	-**	674	1062304
1/2	Dynamic	5156	3844	416	319	735	1172212
	$T.P.$	5172	3828	653	146	799	1194164
	$T.P.'$	5507	3493	890	181	1071	1247944
	Static	5688	3312	992	-	992	1191136
1/3	Dynamic	5110	3890	406	334	740	1180920
	$T.P.$	5232	3768	653	142	795	1183764
	$T.P.'$	5562	3438	898	144	1042	1228784
	Static	5706	3294	1102	-	1102	1230272
1/4	Dynamic	5033	3967	428	336	764	1213216
	$T.P.$	5028	3972	663	156	819	1223076
	$T.P.'$	5471	3529	949	145	1094	1262012
	Static	5616	3384	1230	-	1230	1292232
1/5	Dynamic	4903	4097	433	378	811	1238536
	$T.P.$	4940	4060	681	169	850	1247880
	$T.P.'$	5451	3549	1001	165	1166	1292332
	Static	5616	3384	1463	-	1463	1380772
1/6	Dynamic	4791	4209	451	378	829	1261952
	$T.P.$	4844	4156	727	186	913	1286028
	$T.P.'$	5398	3602	1055	164	1219	1320316
	Static	5562	3438	1671	-	1671	1467804
1/7	Dynamic	4721	4279	470	390	860	1284092
	$T.P.$	4849	4151	774	186	960	1303148
	$T.P.'$	5409	3591	1125	207	1332	1361628
	Static	5616	3384	1758	-	1758	1492872

* The first column represents the portion of all SKUs with normal demand patterns. The lower number in the Normal column refers to the higher volatility.
**The best solution of static model is associated with "no update" during the planning horizon. Thus, the moves to the reserve area is not applicable in a static case.

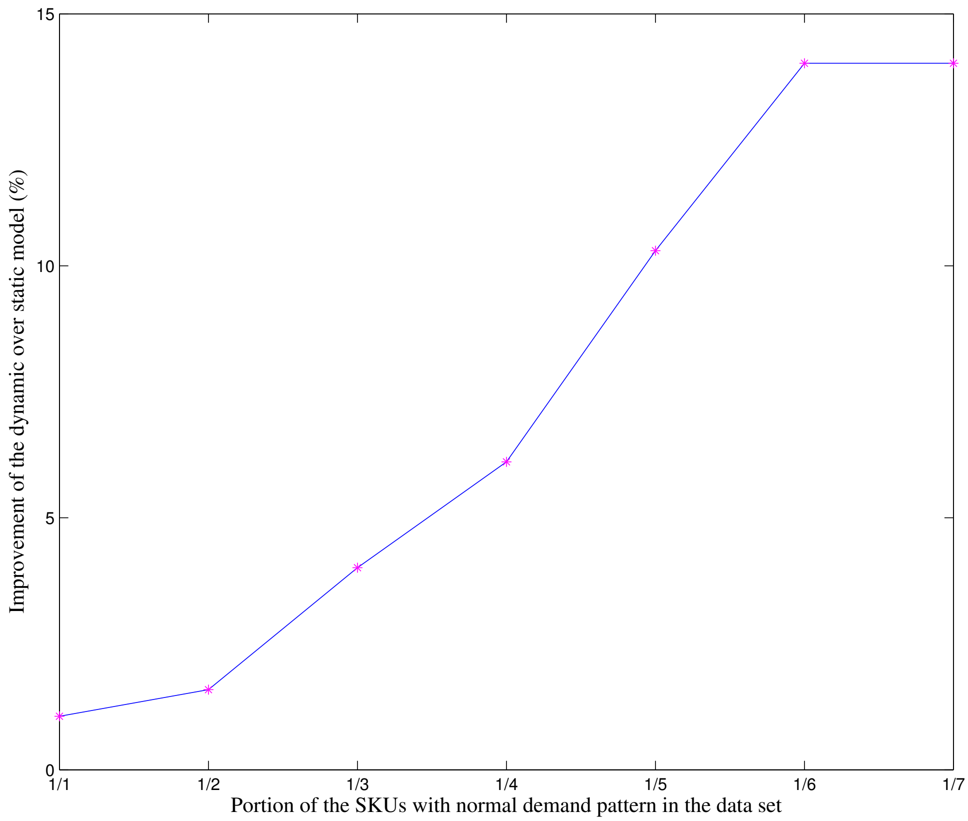

Figure 37. Dynamic model efficiency versus the demand volatility

area, it is always picked from the forward area. Nevertheless, the dynamic model allows picking from the reserve area when unusual orders are being received. The goal is avoiding extra replenishments and moves in high volatile periods.

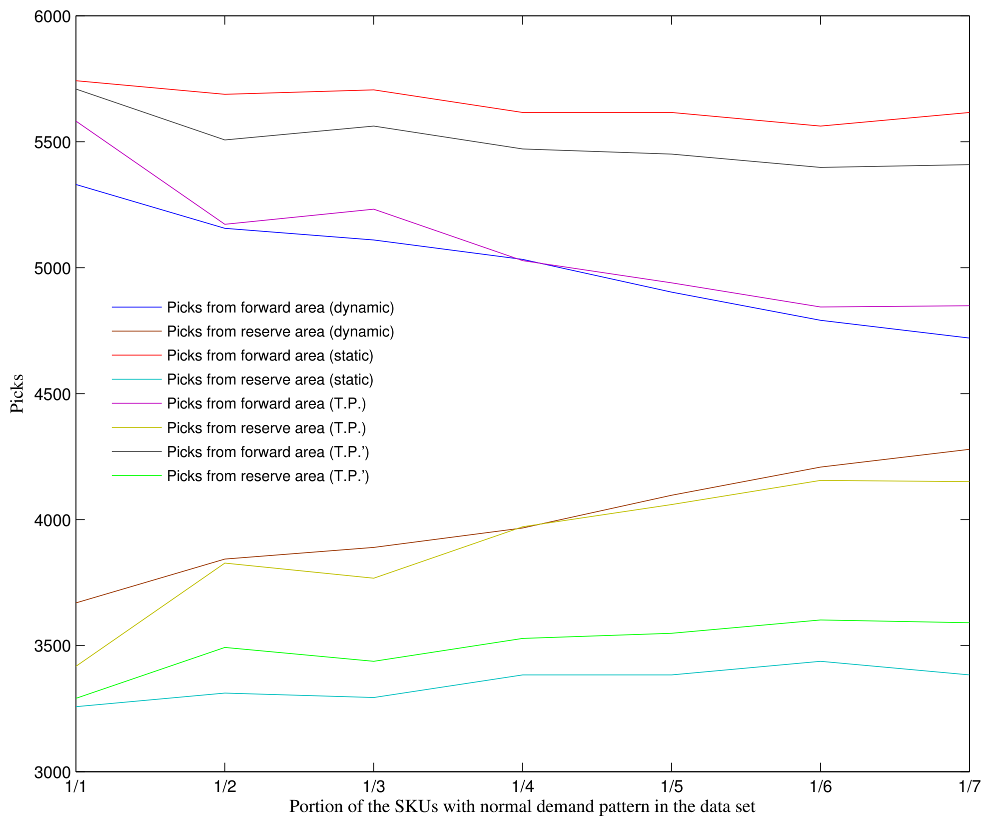

Figure 38. Demand volatility impacts on picks from the forward or reserve area

Figure 39 displays how the dynamic model moderates and controls the total number of moves and replenishments in medium and high volatile periods (1/2 normal and after). Nonetheless, the static model experiences the growth in the total number of the moves and replenishments when facing the demand abnormality.

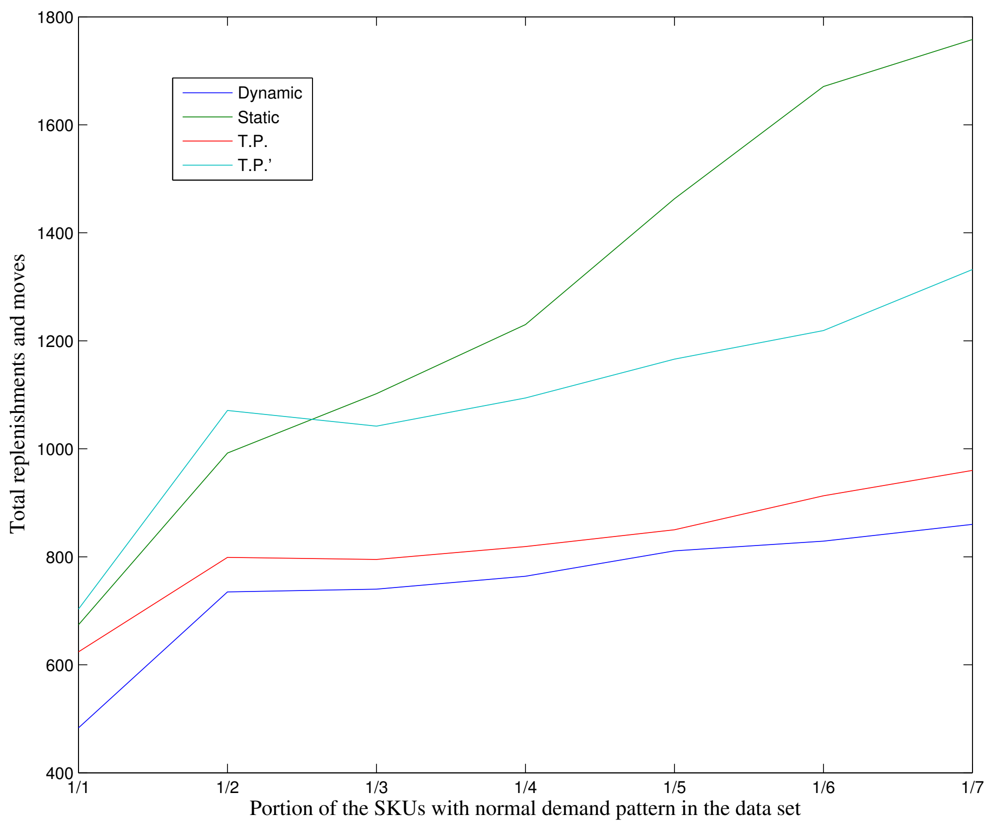

Figure 39. Demand volatility impacts on the total replenishments and moves

Finally, Figure 40 illustrates the total costs of the four models. The dynamic model has the lowest cost in all experiments 1 to 7. From this figure, it can be interpreted that the threshold policy $T.P.$ can fairly represent the dynamic model. The average gap between the dynamic model and the $T.P.$ is 1.21% in this figure. The total cost of the static model gets higher than the $T.P.'$ when the volatility increases (after 1/3 normal.)

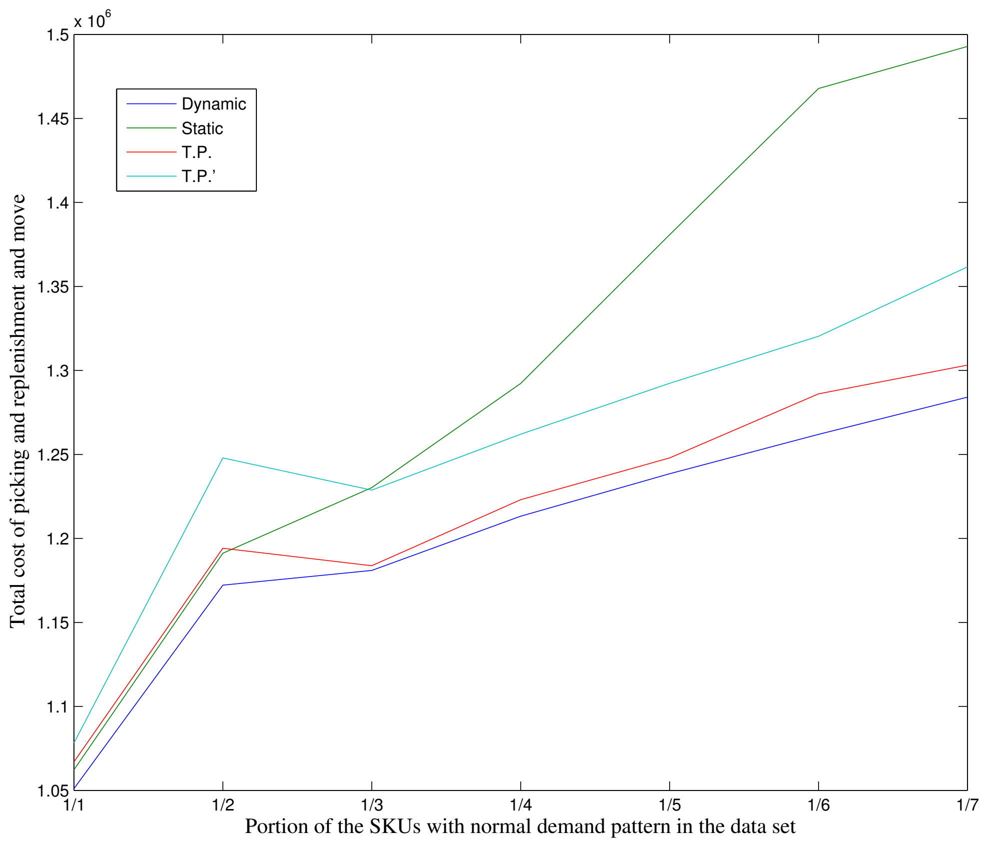

Figure 40. Demand volatility impacts on the total cost

I Checking the robustness of the models

This section checks the robustness of the dynamic model, the static model, and the threshold policies to establish the reliability, validity, and applicability of the results. We will present the experimental design and the statistical analysis to address the three concerns listed below:

1. A sensitivity analysis is carried out to study the effects of the parameters used for the demand trends' generation on the total picking and replenishment costs of the static, $T.P.$, $T.P.'$, and dynamic models. Hence, the total cost is the response variable.

2. We evaluate whether there is statistically a difference between the four aforementioned models.

3. For each model, we evaluate the significance level of difference between the four order data sets. We investigate if there is statistically a significant difference in the mean costs of the four order data sets with different portions of the normal demand pattern.

We study eight types of demand patterns, including normal, up/down trends, up/down shifts, cyclic, systematic and intermittent. The mean of an abnormal pattern, $a(t)$, consists of two important components of a constant term μ and a particular abnormal function $d(t)$ that models a particular abnormal pattern. This term $d(t)$ is zero for the normal demand pattern. The mathematical model for the mean of simulated patterns can be expressed by the following:

$$a(t) = \mu + d(t) \tag{112}$$

In equation 51, $d(t)$ is defined as the following for different abnormal patterns:

1. Up/Down trends: $d(t) = \lambda t$, where λ is the trend slope in terms of σ_ε. The parameter $\lambda > 0$ is selected for up trends and $\lambda < 0$ for down trends.

2. Up/Down shifts: $d(t) = \gamma$, where parameter γ shows the shift magnitude. The parameter $\gamma > 0$ is selected for up shifts and $\gamma < 0$ for down shifts.

3. Cyclic pattern: $d(t) = \kappa(\frac{2\pi t}{\Omega})$, where κ is the amplitude of the cyclic patterns, and Ω is the cyclic pattern period.

4. Systematic trends: $d(t) = \nu(-1)^t$, where ν is the magnitude of systematic pattern.

To obtain the demand patterns, we first generate a random number ρ_t from the normal distribution with the mean $a(t)$ and the standard deviation parameter σ at time t. Then, we apply the Exponentially Weighted Moving Average (EWMA) technique, where the demand at time t depends on the EWMA statistic. EWMA is an exponentially weighted average of all prior demand data, including the most recent demand. We compute successive demand points Z_t using all preceding demand points and the weighting factor of Θ. The EWMA static is calculated as:

$$Z_t = \Theta\rho_t + (1 - \Theta)Z_{t-1} \tag{113}$$

With respect to the broad spectrum of parameter levels in relevant studies, Gauri and Chakraborty (2009) and Shao (2012), a trial and error approach is taken in this research to adjust the model's parameters to our purpose.

The warehouse of our example contains 5000 SKUs. We generate the data sets containing 10% fast and medium movers. The slow movers, which have many zeros in their demand file and follow the intermittent demand pattern, are more efficiently picked from the reserve area. The fast and medium movers, following the demand patterns 1 to 7, are the candidates for being slotted in the forward area. Tables 33, 34, 35 and 36 are associated with the costs of our experiments and can be found in the Appendix.

1 Sensitivity analysis

We performed an experimental design to investigate the effects of six factors listed in table 27 on the cost of each model. We consider two levels, upper and lower bounds, for the six variables. Table 28 shows a two-level full factorial design ($2^6 = 64$ runs for each model) with six variables (factors). The response is the total picking and replenishment costs.

TABLE 27

Factors and levels in experimental design

Factor	Factor in ANOVA^1	Factor in ANOVA^2	Description	Level 1	Level 2
Normal %	X_1	A	% of normal patterns	10%	70%
λ	X_2	B	Up/Down trends	0.005	0.008
γ	X_3	C	Up/Down shifts	1.5	2
κ	X_4	D	Cyclic pattern	0.5	0.75
ν	X_5	E	Systematic trends	0.5	0.75
σ	X_6	F	Standard deviation	0.7	0.9

In the sensitivity analysis of the demand patterns' parameters, we perform the steps below for all models:

1. Run each model with 64 data sets, corresponding to our full factorial design. Each row of the table 28 represents one experiment out of 64 experiments.

2. Conduct a six-way analysis of variance (ANOVA) to extract the main factors with a P-value less than 0.05 (ANOVA^1). The results of the ANOVA^1 tests for

the static, $T.P.$, $T.P.'$, and the dynamic models have been presented in Figures 41, 44, 47, and 50, respectively.

3. Delete the negligible effect factors with a P-value greater than 0.05.

4. Conduct the second n-way ANOVA test (ANOVA^2), where $n = 6-$ (No. of deleted factors), by considering the two-factors' interactions and creating a generalized linear regression model. Figures 42, 45, 48, and 51 are the ANOVA^2 tests for the static, $T.P.$, $T.P.'$, and the dynamic models, respectively.

5. Plot the normal probability plots in Figures 43,46, 49, and 52, which verify the significant effects and interaction for the static, $T.P.$, $T.P.'$, and the dynamic models, respectively. The statistical and magnitude significance of the main effects and their interaction effects in a two-level factorial design can be compared using normal probability plots. If the effects were zero, we would expect the points to fall on the fitted line. Significant effects have a label and fall toward the left or right side of the graph. The negative effects are on the left side of the graph, and the positive effects are on the right side of the graph.

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	9121873819536	1	9121873819536	1950.22	0
X2	229132627684	1	229132627684	48.99	0
X3	579121	1	579121	0	0.9912
X4	786690304	1	786690304	0.17	0.6833
X5	956479329	1	956479329	0.2	0.6528
X6	2560967236	1	2560967236	0.55	0.4624
Error	266608842301	57	4677348110.5		
Total	9621920005511	63			

Figure 41. ANOVA^1 test for the static model

TABLE 28

Full factorial design 2^6

#	% Normal	λ	γ	κ	ν	σ	#	% Normal	λ	γ	κ	ν	σ
1	-	+	+	+	+	+	33	+	+	+	+	+	+
2	-	+	+	+	+	-	34	+	+	+	+	+	-
3	-	+	+	+	-	+	35	+	+	+	+	-	+
4	-	+	+	-	+	+	36	+	+	+	-	+	+
5	-	+	-	+	+	+	37	+	+	-	+	+	+
6	-	-	+	+	+	+	38	+	-	+	+	+	+
7	-	+	+	+	-	-	39	+	+	+	+	-	-
8	-	+	+	-	-	+	40	+	+	+	-	-	+
9	-	+	-	-	+	+	41	+	+	-	-	+	+
10	-	-	-	+	+	+	42	+	-	-	+	+	+
11	-	+	+	-	+	-	43	+	+	+	-	+	-
12	-	+	-	+	+	-	44	+	+	-	+	+	-
13	-	-	+	+	+	-	45	+	-	+	+	+	-
14	-	+	-	+	-	+	46	+	+	-	+	-	+
15	-	-	+	+	-	+	47	+	-	+	+	-	+
16	-	-	+	-	+	+	48	+	-	+	-	+	+
17	-	+	+	-	-	-	49	+	+	+	-	-	-
18	-	+	-	+	-	-	50	+	+	-	+	-	-
19	-	+	-	-	+	-	51	+	+	-	-	+	-
20	-	+	-	-	-	+	52	+	+	-	-	-	+
21	-	-	+	+	-	-	53	+	-	+	+	-	-
22	-	-	+	-	+	-	54	+	-	+	-	+	-
23	-	-	+	-	-	+	55	+	-	+	-	-	+
24	-	-	-	+	+	-	56	+	-	-	+	+	-
25	-	-	-	+	-	+	57	+	-	-	+	-	+
26	-	-	-	-	+	+	58	+	-	-	-	+	+
27	-	-	-	-	-	+	59	+	-	-	-	-	+
28	-	-	-	-	+	-	60	+	-	-	-	+	-
29	-	-	-	+	-	-	61	+	-	-	+	-	-
30	-	-	+	-	-	-	62	+	-	+	-	-	-
31	-	+	-	-	-	-	63	+	+	-	-	-	-
32	-	-	-	-	-	-	64	+	-	-	-	-	-

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
A	9.12187e+12	1	9.12187e+12	16959.09	2.68069e-75
B	2.29133e+11	1	2.29133e+11	426	6.08482e-29
A*B	2.38641e+11	1	2.38641e+11	443.67	2.07834e-29
Error	3.22725e+10	60	5.37875e+08		
Total	9.62192e+12	63			

Figure 42. ANOVA^2 test for the static model

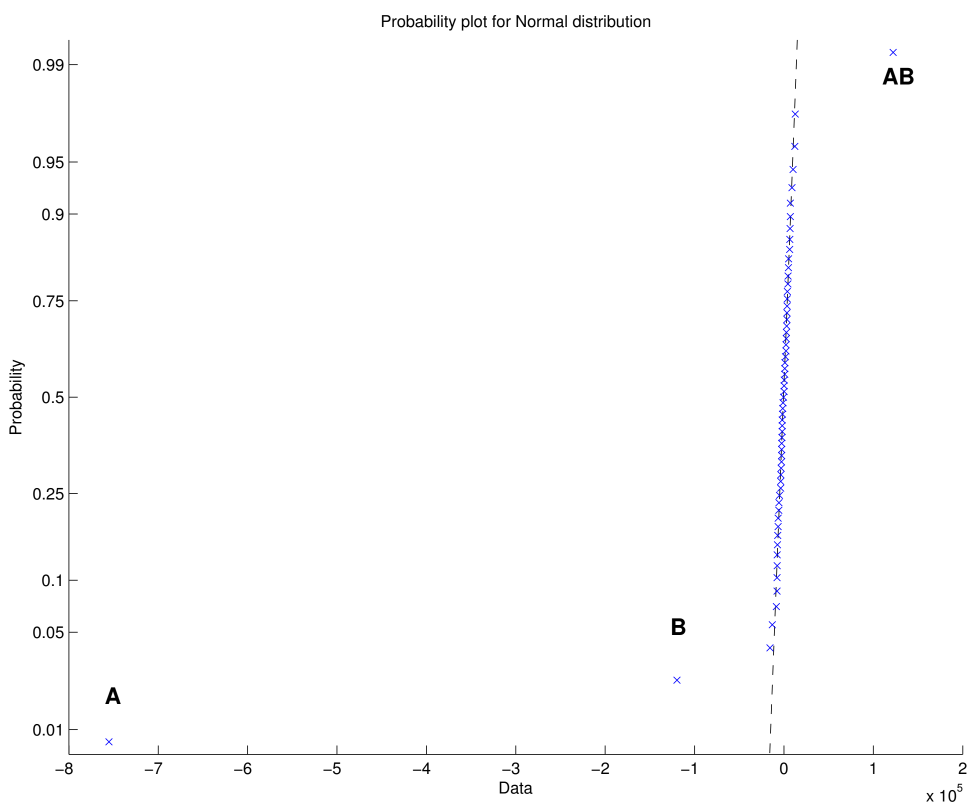

Figure 43. Normal Probability Plot for the static model

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	5371985922270.3	1	5371985922270.3	25756.9	0
X2	2993238810.2	1	2993238810.2	14.35	0.0004
X3	2699166162.3	1	2699166162.3	12.94	0.0007
X4	733592.2	1	733592.2	0	0.9529
X5	108420156.2	1	108420156.2	0.52	0.4739
X6	93190062.2	1	93190062.2	0.45	0.5065
Error	11888201636.2	57	208564941		
Total	5389768872689.8	63			

Figure 44. ANOVA[1] test for the T.P. model

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
A	5.37199e+12	1	5.37199e+12	38226.97	0
B	2.99324e+09	1	2.99324e+09	21.3	0
C	2.69917e+09	1	2.69917e+09	19.21	0.0001
A*B	2.14503e+09	1	2.14503e+09	15.26	0.0003
A*C	1.88317e+09	1	1.88317e+09	13.4	0.0006
B*C	5.22079e+07	1	5.22079e+07	0.37	0.5446
Error	8.01014e+09	57	1.40529e+08		
Total	5.38977e+12	63			

Figure 45. ANOVA[2] test for the T.P. model

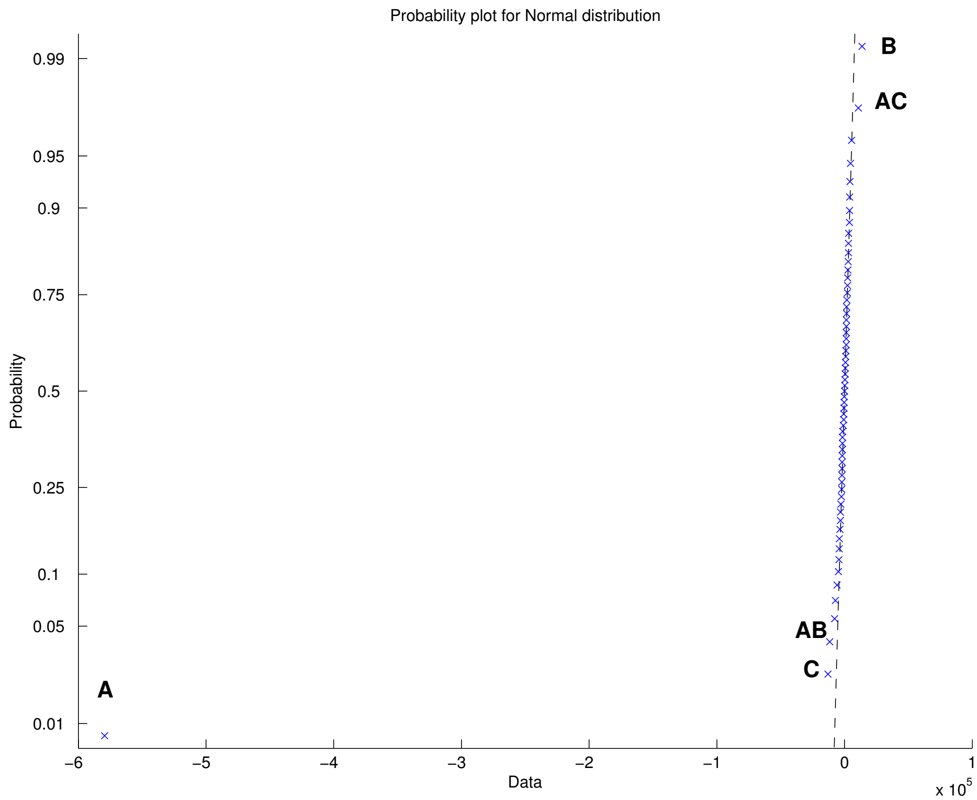

Figure 46. Normal Probability Plot for the T.P. model

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	5279050771542.2	1	5279050771542.2	28113.28	0
X2	1048173000.2	1	1048173000.2	5.58	0.0216
X3	1633574306.2	1	1633574306.2	8.7	0.0046
X4	537103800.2	1	537103800.2	2.86	0.0963
X5	42642.2	1	42642.2	0	0.988
X6	106450806.2	1	106450806.2	0.57	0.4546
Error	10703335964.2	57	187777823.9		
Total	5293079452061.8	63			

Figure 47. ANOVA[1] test for the T.P.' model

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
A	5.27905e+12	1	5.27905e+12	33230.04	0
B	1.04817e+09	1	1.04817e+09	6.6	0.0129
C	1.63357e+09	1	1.63357e+09	10.28	0.0022
A*B	1.73884e+08	1	1.73884e+08	1.09	0.2999
A*C	1.70028e+09	1	1.70028e+09	10.7	0.0018
B*C	4.17528e+08	1	4.17528e+08	2.63	0.1105
Error	9.05524e+09	57	1.58864e+08		
Total	5.29308e+12	63			

Figure 48. ANOVA² test for the T.P.' model

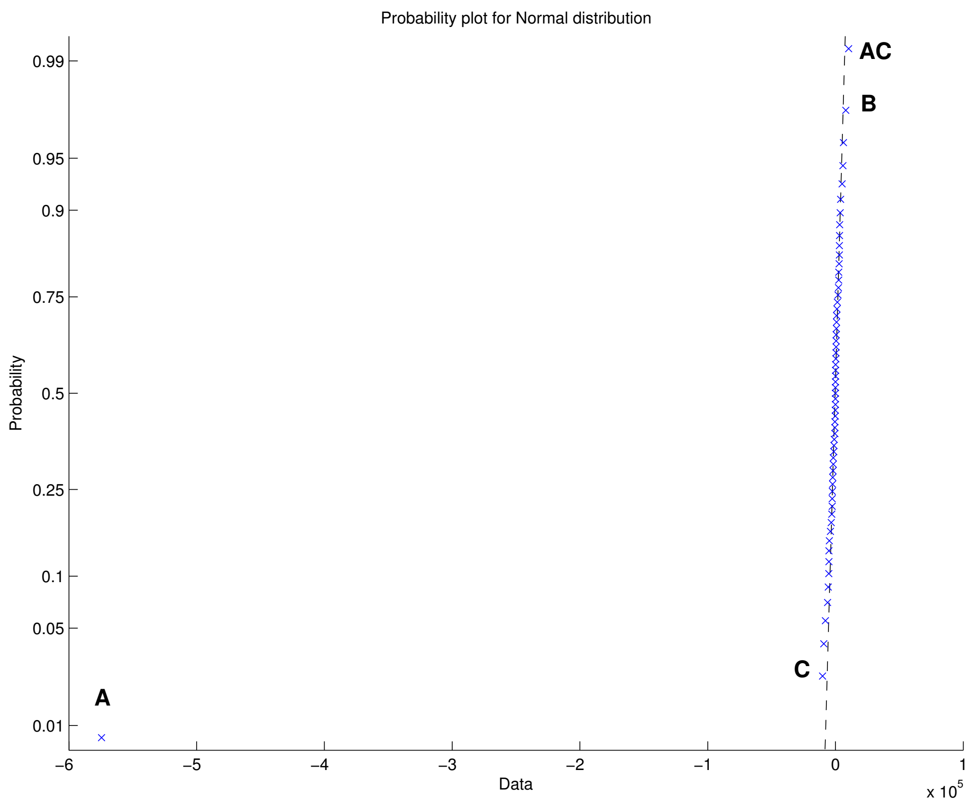

Figure 49. Normal Probability Plot for the T.P.' model

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	5.17941e+12	1	5.17941e+12	31099.31	0
X2	2.51989e+09	1	2.51989e+09	15.13	0.0003
X3	1.63014e+09	1	1.63014e+09	9.79	0.0028
X4	3.30331e+06	1	3.30331e+06	0.02	0.8885
X5	1.14041e+08	1	1.14041e+08	0.68	0.4114
X6	9.69043e+07	1	9.69043e+07	0.58	0.4487
Error	9.49302e+09	57	1.66544e+08		
Total	5.19327e+12	63			

Figure 50. ANOVA[1] test for the dynamic model

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
A	5.17941e+12	1	5.17941e+12	42955.08	0
B	2.51989e+09	1	2.51989e+09	20.9	0
C	1.63014e+09	1	1.63014e+09	13.52	0.0005
A*B	1.92024e+09	1	1.92024e+09	15.93	0.0002
A*C	8.6201e+08	1	8.6201e+08	7.15	0.0098
B*C	5.2114e+07	1	5.2114e+07	0.43	0.5136
Error	6.87291e+09	57	1.20577e+08		
Total	5.19327e+12	63			

Figure 51. ANOVA[2] test for the dynamic model

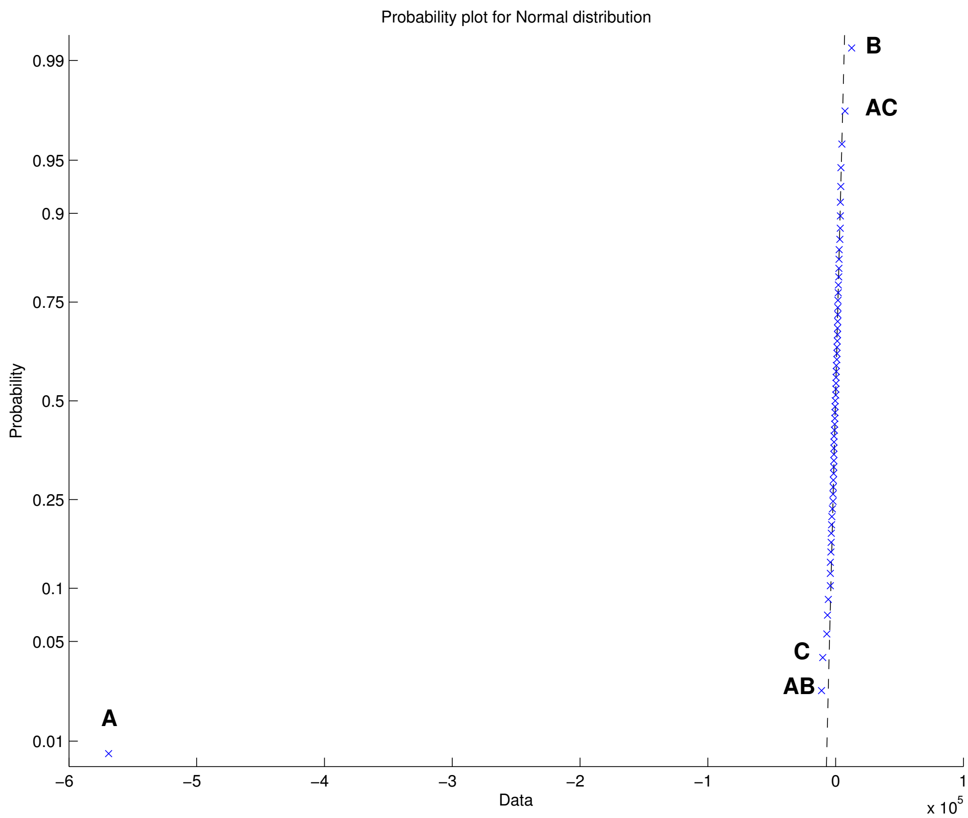

Figure 52. Normal Probability Plot for the dynamic model

Table 29 summarizes our sensitivity analysis results. It shows the main effects of all models. The percentage of the SKUs with a normal demand pattern (factor A) and the up/down trend parameter (factor B) are the main effects of all models. The $T.P.'$ model is the only model that is not significantly affected by the interaction of factor A and B. It is observed that while the dynamic model and threshold policies TP and $T.P.'$ are sensitive to the up/down shift parameter (factor C) and their interaction with factor A, the static model is not affected by factor C. Factors κ, ν, and σ (D,E,F) are not a main effect of all four models.

TABLE 29

Summary of the main effects of the models

	Static	T.P.	T.P.'	Dynamic
Main effects and interactions	A B AB	A B C AB AC	A B C AC	A B C AB AC

In this section, we extracted the main effects of the dynamic model, the static mode,l and the threshold policies. The results of Table 29 justifies the following conclusions:

- The portion of the SKUs with normal demand patterns and the up/down trend parameter are both the main effects of all aforementioned models. Therefore, the forward area's picks and replenishments decisions as well as the total costs are influenced by these two factors.

- While the static model is not sensitive to the up/down shift parameter, both the dynamic model and the threshold policies are sensitive to this parameter. If the SKUs' demand patterns are experiencing the up/down shift, it is expected that the dynamic model adjusts the layout of the forward area, causing its total cost to be significantly affected by this adjustment. Nevertheless, the up/down shifts in the demand data do not impact the static model decisions about the forward area.

- As expected, the main effects of the dynamic model and $T.P.$ are the same.

- The most interesting insight found from the results of Table 29 was that none of the discussed models are significantly affected by the cyclic and systematic demand patterns as well as the standard deviation used for generating the random normal number in the EWMA statistic. When the demand time series data exhibit rises and falls in the cyclic or systematic patterns, even the dynamic model and the threshold policies are not significantly affected by those fluctuations in the demand trends.

2 Statistical comparison of the models

A one-way analysis of variance with sample size 128 was performed to compare the static, dynamic, $T.P.$, and $T.P.'$ models in a cost manner. The hypotheses of interest in our ANOVA are as follows:

$H_0 : \mu^{static} = \mu^{T.P.} = \mu^{T.P.'} = \mu^{dynamic}$

H_1: The means are not equal,

where μ^{model} is the mean cost of model. The results of ANOVA test in Figure 53 with very small p-value (less tan 0.05) verifies there is significant difference between four models and the null hypothesis is rejected.

The box plot of the costs in Figure 54 shows the difference between the median of the dynamic model and the threshold policy $T.P.$ is negligible; however, the dynamic model outperforms the threshold policy as well as the static and $T.P.'$ models. The $T.P.'$ model, which is based on the dynamic approach, but disregards the slot allocation, competes with the static model, which considers both the SKU assignment and slot allocation.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	4.40567e+12	3	1.46856e+12	26.6	5.43037e-16
Error	2.80509e+13	508	5.52184e+10		
Total	3.24566e+13	511			

Figure 53. ANOVA table for comparison of the models

3 Effects of the size of normal patterns in order data

In this section, we will investigate our models separately to find out whether there is statistically significant difference in the mean cost among the four groups of order data with different portions of normal demand trends (10%, 30%, 50% and 70%). The ANOVA test is conducted for each model to compare the mean cost of the four groups. The groups are independent and the sample size is 32. The hypotheses of interest in our ANOVA are as follows:

$H_0 : \mu^M_{10\%} = \mu^M_{30\%} = \mu^M_{50\%} = \mu^M_{70\%}$

H_1: The means are not equal,

where $\mu^M_{x\%}$ represents the mean cost of the model $M \in \{\text{static}, T.P., T.P.', \text{dynamic}\}$, corresponding to the order data with x% normal demand trends.

Figures 55, 57, 59, and 61 present the results of ANOVA tests for the static, dynamic, $T.P.$ and $T.P.'$ models, respectively. The large F-statistics and small p-value

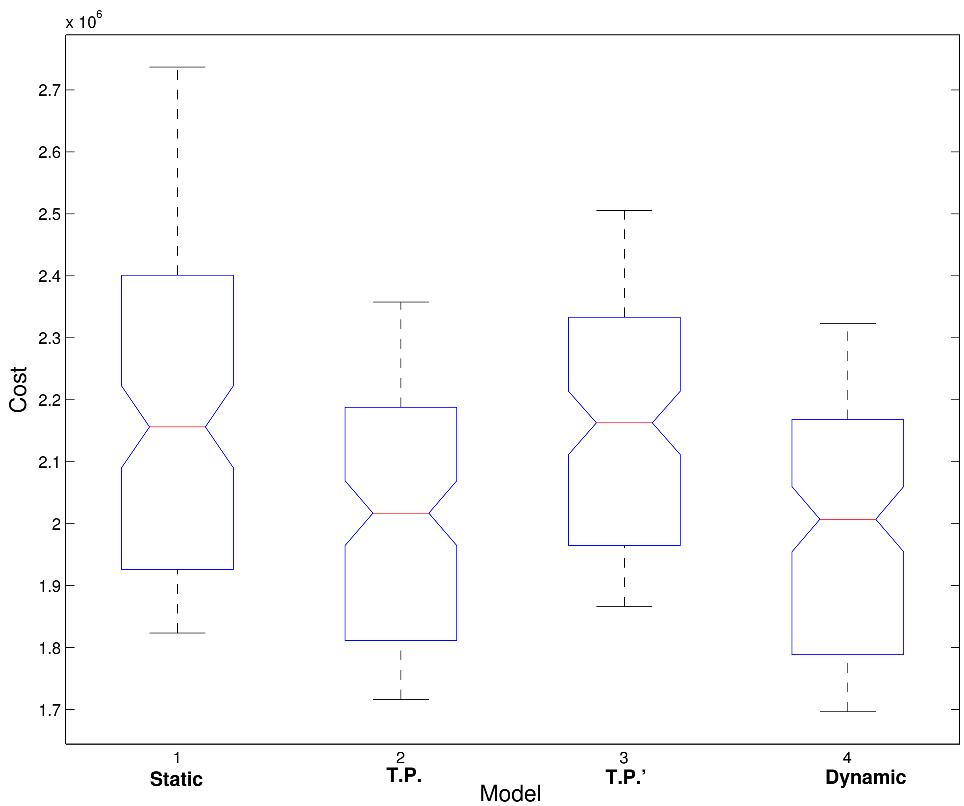

Figure 54. Box plot of the costs for four models

(less than 0.05) confirms that there is statistically significant difference between the four groups of order data in all models, causing us to reject the null hypothesis.

The box plots in Figures 56, 58, 60, and 62, corresponding to each of the aforementioned models, visually represents the cost data for the four groups with different portions of normal demand patterns. For all studied models, it is observed that the medians of groups are not equal and the data set with the lowest volatility (70% normal patterns) has the least median and variation. Comparing the four models, the static model shows the greater variation in cost data when the percentage of the normal demand patterns are 10%, 30% and 50%. Therefore, the models based on the dynamic slotting strategy ($T.P.$, $T.P.'$ and the dynamic model) are more robust than the static model.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	9.52882e+12	3	3.17627e+12	337.03	2.02602e-59
Error	1.1686e+12	124	9.42423e+09		
Total	1.06974e+13	127			

Figure 55. ANOVA table for significance of the normal patterns portion in the order data (static model)

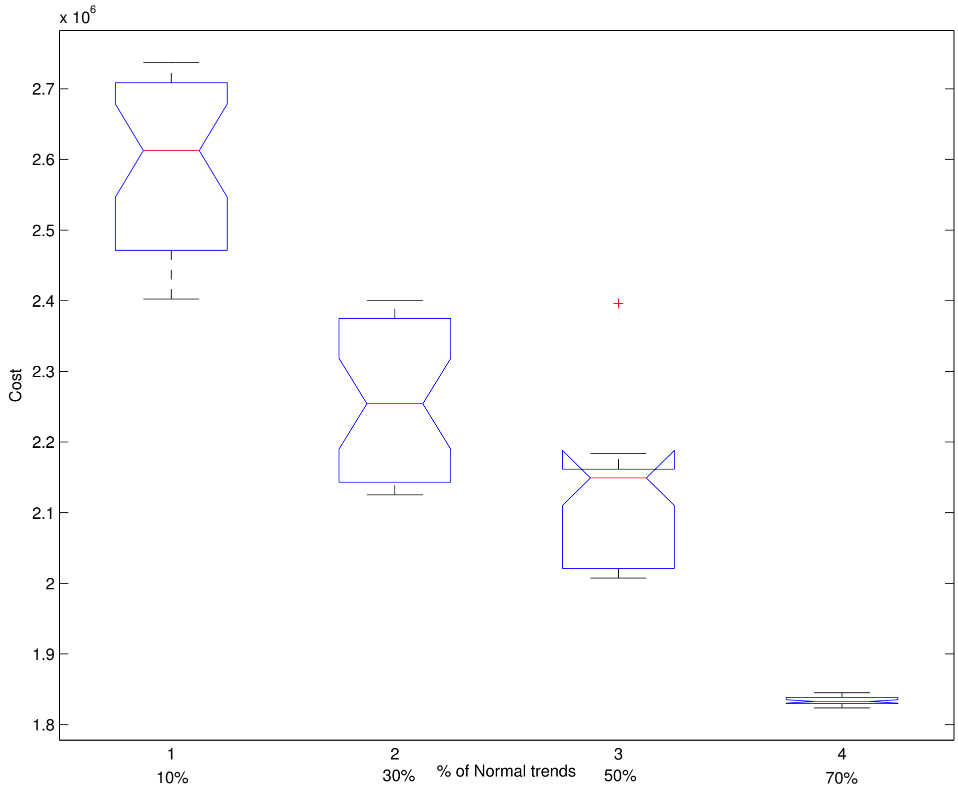

Figure 56. Box plot for significance of the normal patterns portion in the order data (static model)

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	5.80768e+12	3	1.93589e+12	3071.74	3.81288e-116
Error	7.8148e+10	124	6.30226e+08		
Total	5.88583e+12	127			

Figure 57. ANOVA table for significance of the normal patterns portion in the order data ($T.P.$ model)

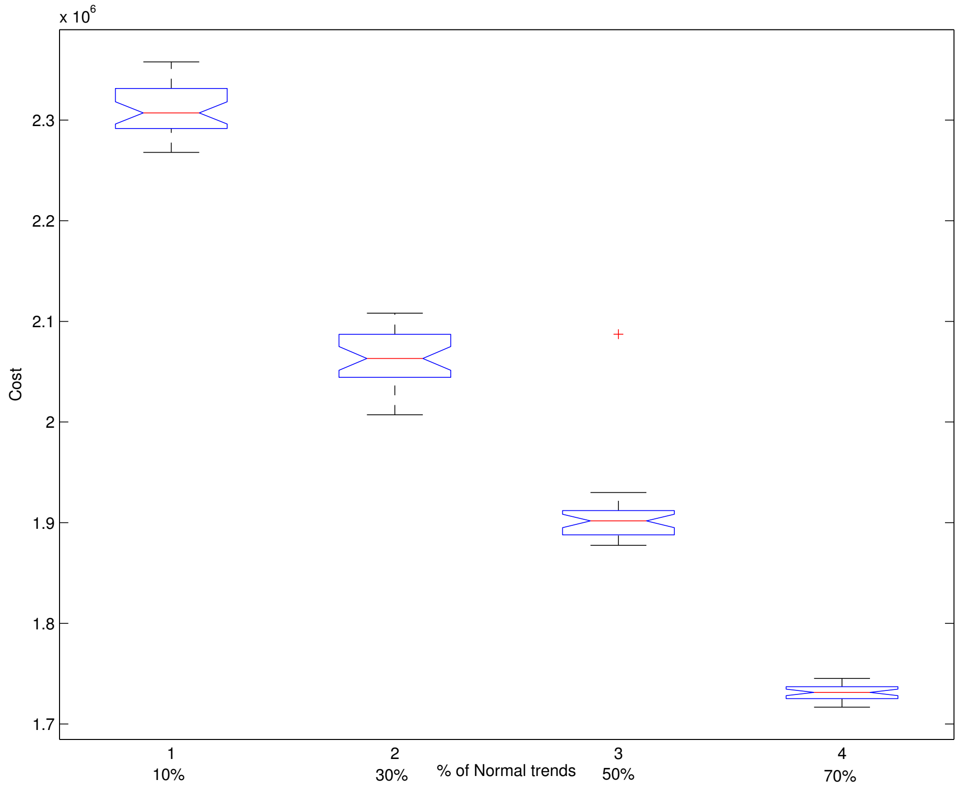

Figure 58. Box plot for significance of the normal patterns portion in the order data ($T.P.$ model)

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	5.65343e+12	3	1.88448e+12	3156.89	7.15736e-117
Error	7.40206e+10	124	5.9694e+08		
Total	5.72745e+12	127			

Figure 59. ANOVA table for significance of the normal patterns portion in the order data ($T.P.'$ model)

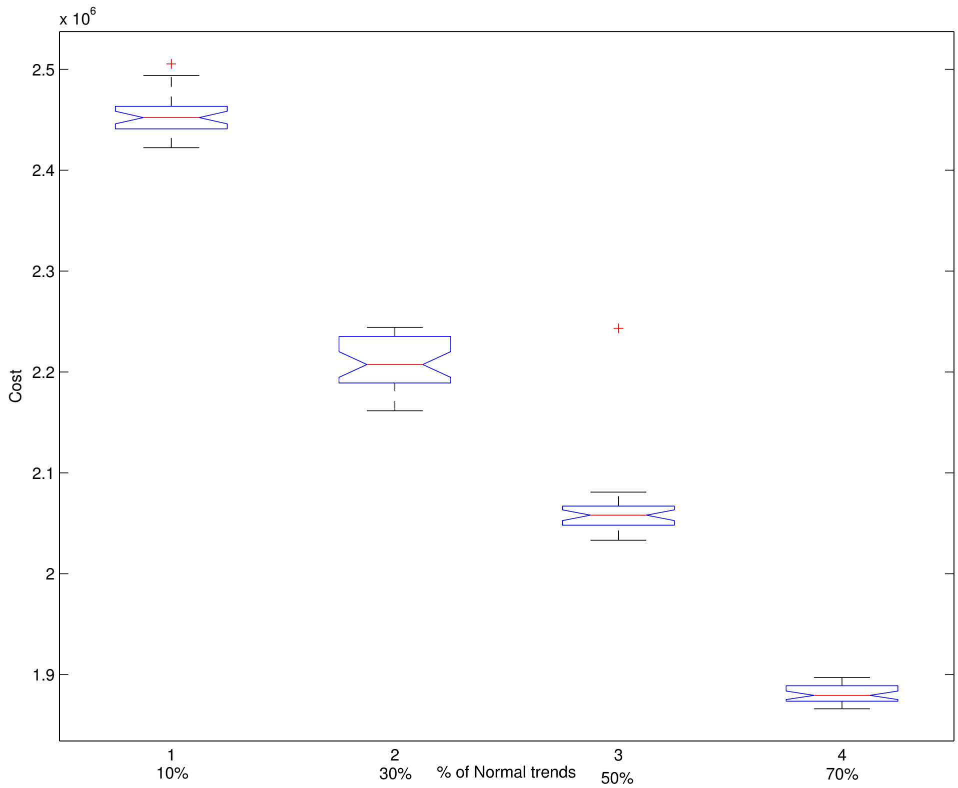

Figure 60. Box plot for significance of the normal patterns portion in the order data ($T.P.'$ model)

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	5.65343e+12	3	1.88448e+12	3156.89	7.15736e-117
Error	7.40206e+10	124	5.9694e+08		
Total	5.72745e+12	127			

Figure 61. ANOVA table for significance of the normal patterns portion in the order data (dynamic model model)

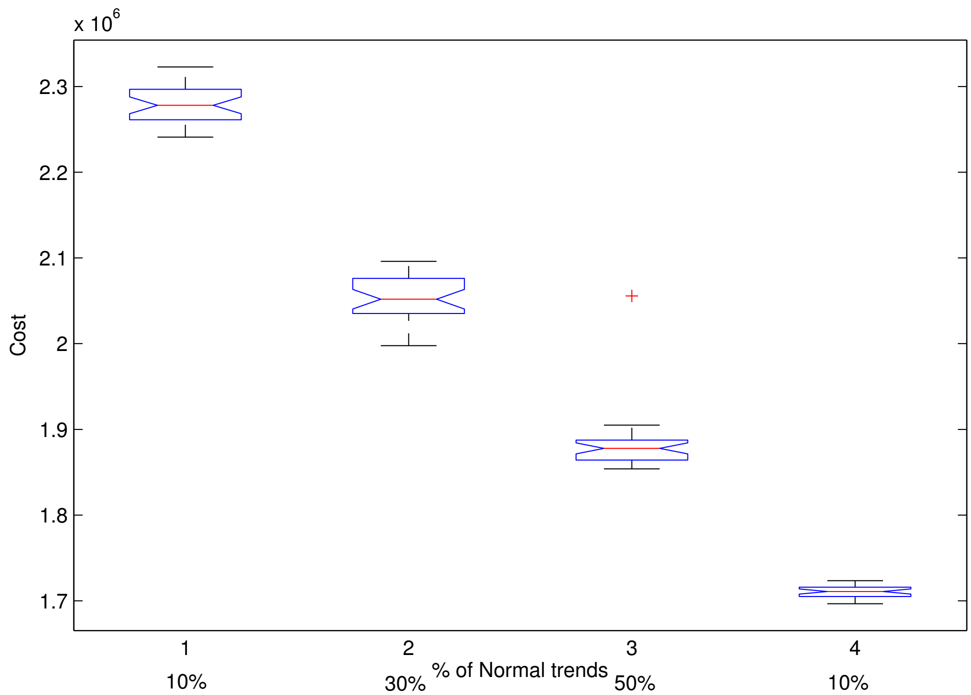

Figure 62. Box plot for significance of the normal patterns portion in the order data (dynamic model)

J Conclusion

Dynamic slotting of the forward area is a warehousing approach where the set of SKUs and the slots allocated to them are changed to continuously have an updated layout that will improve the picking and replenishment costs. The main contribution of this chapter is developing the first mathematical programming formulation for the dynamic slotting optimization with discrete slot allocation as well as the MIP formulations for the dynamic models with different replenishment strategies. We quantified the gap between the static and dynamic forward-reserve problems in terms of the total picking and replenishment costs. Two heuristics based on the threshold policies are proposed that closely perform as well as the dynamic model but have a shorter solution time. The second heuristic enforces one slot per SKU in the forward area. We showed that when the SKUs' demand patterns are highly volatile, the dynamic model significantly reduces the overall costs. Through our experimental design, we validated and compared the static, dynamic, and the threshold policies models. An exhaustive full factorial design was executed to check the sensitivity of the aforementioned models to the variety of the factors affecting the SKUs' demand patterns.

CHAPTER VI

SUMMARY

The forward area is a small area of the warehouse with a low picking cost. Therefore, the items of the warehouse compete to be located in this area rather than the reserve area, which has a higher picking cost. Two approaches for selecting the SKUs of the fast picking area and the allocated space were investigated: the static and dynamic approaches.

In the static forward-reserve problem, we developed the discrete assignment, allocation and sizing model for large size problems. Prioritizing to solve any of these three problems and then using the resulting solution as the input to others is not the best strategy. A heuristic for the discrete forward-reserve problem has been suggested (Walter et al., 2013), but it is not applicable for large problems due to the solution time. They also assume that the slots are always wider than the SKUs and do not solve the three mentioned problems together. We are first to solve the assignment, allocation, and sizing problems simultaneously with very small solution time for large size problems and no restriction on the SKUs and slots dimensions. We developed two heuristics for the situations: with or without SKUs and slots dimensions. We compared several scenarios for the SKU labor efficiency, which is a key component of our heuristics. The heuristics were tested with real data.

Additional contribution of the static FRP study to the literature also includes the proposed algorithm for both profiling and slotting optimization. The proper profiling significantly reduces the replenishment activities and picking costs, while at the same time, maximizing the space utilization within a slot type in the forward area. The proposed algorithm evaluates the slot types in the fast picking areas and determines the best size of each pick mode, along with the SKU assignment and slot allocation.

We introduced the concept of a dynamic forward-reserve problem to warehousing. Under the dynamic environment, different sets of SKUs are assigned to the forward area and the number of slots allocated to them is not fixed for different periods. Therefore, the fast picking area is updated over time with replenishment of the appropriate SKUs, as opposed to the traditional static model that periodically reslots the forward area to reach the target map. A proper slotting methodology not only considers seasonality, but also other types of demand shifts, trends, and frequencies. We explored the methods for demand pattern detection and demand forecasting before proposing the dynamic model.

We proposed the MIP mathematical model for the dynamic forward-reserve problem for the first time. This model relaxes the major implicit assumptions of the static model. Assignment, allocation, and sizing problems are highly dependent on the activity distribution of products. Considering a fixed demand over time adversely affects decisions made regarding this efficient area of the warehouse in terms of a low picking cost.

We quantified the effects of using a static versus a dynamic setting. In our experiments, the dynamic model for SKU assignment and slot allocation in the fast picking area always outperforms the static model, regardless of having or not having future orders. The lost savings that resulted from forecasting errors is negligible in the dynamic model. Results show that early reslotting of the forward area is not the best way to always have the most effective layout of the forward area. Updating the layout of the fast picking area can be very costly if it is done at an inappropriate time.

Different replenishment policies for the forward area were investigated. Choosing the option of partial replenishment or full replenishment of slots affects the total picking and replenishment costs. The slots get empty later in the full replenishment scenario, but the number of replenishments is lower than the quantity replenishment scenario. In both strategies, the slow movers can be deleted from the forward area any time by moving them to the reserve area. The results recommend the full replenishment over the quantity replenishment. Even if the slot is not empty but needs more inventory to meet the demand, it is suggested to replenish the slot up to the full capacity. The dynamic model assumes that the replenishment of the forward area can be promptly accomplished.

Compared to the dynamic model, the static model excludes a significant portion of SKUs to be stored in the forward area. Smaller forward areas require more moves to the reserve area in the dynamic strategy to stay tuned with changes. The benefits attained from the dynamic model over the static model is greater for more volatile warehouses because the dynamic model adjusts the forward area's layout quickly to the changes in the demand pattern by replenishing the new SKUs. This research provides insights for practitioners to choose the appropriate setting for updating their forward area.

Finally, we developed a simple threshold policy that performs almost as well as the dynamic model. The dynamic model gets significantly more computationally expensive for large problems. The suggested intuitive methodology delivers a near optimal solution within a reasonable computing time as well as a good performance for the sensible number of SKUs and the size of the forward area in practice. Another model based on our threshold policy was developed to optimize the dynamic SKU assignment, but not slot allocation, in the forward area. The static and dynamic models were compared with two threshold policies by several experiments. The robustness of the models were checked with designing the experiments on the factors impacting the models. These factors include the parameters used for the demand trends' generation. Thus we can generalize our conclusions with these experiments.

This study provides insights for the practitioners who aim to achieve the pick efficiency by applying the dynamic slotting approach. Based on our numerical tests, the dynamic strategy can improve the total picking and replenishment costs by 6% to 14%.

REFERENCES

Yousef Al-Assaf. Multi-resolution wavelets analysis approach for the recognition of concurrent control chart patterns. *Quality engineering*, 17(1):11–21, 2004.

John J Bartholdi and Steven T Hackman. Allocating space in a forward pick area of a distribution center for small parts. *IIE Transactions*, 40(11):1046–1053, 2008.

John J Bartholdi and Steven T Hackman. Warehouse & distribution science: release 0.92. *The Supply Chain and Logistics Institute, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA*, 30332, 2010.

Oded Berman and Eungab Kim. Dynamic order replenishment policy in internet-based supply chains. *Mathematical Methods of Operations Research*, 53(3):371–390, 2001.

Oded Berman and Eungab Kim. Dynamic inventory strategies for profit maximization in a service facility with stochastic service, demand and lead time. *Mathematical methods of operations research*, 60(3):497–521, 2004.

Gabriel R Bitran and Arnoldo C Hax. Disaggregation and resource allocation using convex knapsack problems with bounded variables. *Management Science*, 27(4): 431–441, 1981.

Yavuz Ahmet Bozer. Optimizing throughput performance in designing order picking systems. 1985.

Yossi Bukchin, Eugene Khmelnitsky, and Pini Yakuel. Optimizing a dynamic order-picking process. *European Journal of Operational Research*, 219(2):335–346, 2012.

J Do Croston. Forecasting and stock control for intermittent demands. *Operational research quarterly*, pages 289–303, 1972.

Harwin de Vries, Ruth Carrasco-Gallego, Taoying Farenhorst-Yuan, and Rommert Dekker. Prioritizing replenishments of the forward reserve. Technical report, Econometric Institute Research Papers, 2012.

Harwin de Vries, Ruth Carrasco-Gallego, Taoying Farenhorst-Yuan, and Rommert Dekker. Prioritizing replenishments of the piece picking area. *European Journal of Operational Research*, 236(1):126–134, 2014.

Edward H Frazelle, Steven T Hackman, U Passy, and Loren Kerry Platzman. *The forward-reserve problem.* John Wiley & Sons, Inc., 1994.

Jean-Philippe Gagliardi, Angel Ruiz, and Jacques Renaud. Space allocation and stock replenishment synchronization in a distribution center. *International Journal of Production Economics*, 115(1):19–27, 2008.

Susanta Kumar Gauri and Shankar Chakraborty. Feature-based recognition of control chart patterns. *Computers & Industrial Engineering*, 51(4):726–742, 2006.

Susanta Kumar Gauri and Shankar Chakraborty. Recognition of control chart patterns using improved selection of features. *Computers & Industrial Engineering*, 56 (4):1577–1588, 2009.

Neil A Gershenfeld. *The nature of mathematical modeling.* Cambridge university press, 1999.

Marc Goetschalckx and H Donald Ratliff. Shared storage policies based on the duration stay of unit loads. *Management Science*, 36(9):1120–1132, 1990.

Yeming Gong and Rene De Koster. A polling-based dynamic order picking system for online retailers. *IIE Transactions*, 40(11):1070–1082, 2008.

Jinxiang Gu, Marc Goetschalckx, and Leon F McGinnis. Solving the forward-reserve allocation problem in warehouse order picking systems. *Journal of the Operational Research Society*, 61(6):1013–1021, 2010.

Ruey-Shy Guh and Yi-Chih Hsieh. A neural network based model for abnormal pattern recognition of control charts. *Computers & Industrial Engineering*, 36(1): 97–108, 1999.

Rafael S Gutierrez, Adriano O Solis, and Somnath Mukhopadhyay. Lumpy demand forecasting using neural networks. *International Journal of Production Economics*, 111(2):409–420, 2008.

Steven T Hackman and Loren K Platzman. Near-optimal solution of generalized resource allocation problems with large capacities. *Operations Research*, 38(5): 902–910, 1990.

Steven T Hackman, Meir J Rosenblatt, and John M Olin. Allocating items to an automated storage and retrieval system. *IIE transactions*, 22(1):7–14, 1990.

Adnan Hassan, M Shariff Nabi Baksh, Awaluddin Mohd Shaharoun, and Hishamuddin Jamaluddin. Improved spc chart pattern recognition using statistical features. *International Journal of Production Research*, 41(7):1587–1603, 2003.

Warren H Hausman, Leroy B Schwarz, and Stephen C Graves. Optimal storage assignment in automatic warehousing systems. *Management Science*, 22(6):629–638, 1976.

Bradley K Hollingsworth. *Decision strategy to minimize replenishment costs in a distribution center with forward-reserve storage*. PhD thesis, Ohio University, 2003.

Stephanie A Jernigan. Multi-tier inventory systems with space constraints. 2004.

Mehmed Kantardzic. *Data mining: concepts, models, methods, and algorithms*. John Wiley & Sons, 2011.

Byung Soo Kim and Woon-Seek Lee. Meta-heuristic algorithms for a multi-product dynamic lot-sizing problem with a freight container cost. *Industrial Engineeering & Management Systems*, 11(3):288–298, 2012.

Byung Soo Kim and Woon-Seek Lee. A multi-product dynamic inbound ordering and shipment scheduling problem at a third-party warehouse. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 20(1-2), 2013.

Byung Soo Kim and Jeffrey S Smith. Slotting methodology using correlated improvement for a zone-based carton picking distribution system. *Computers & Industrial Engineering*, 62(1):286–295, 2012.

Byung Soo Kim, Woon-Seek Lee, and Shiegheun Koh. A genetic algorithm for dynamic inbound ordering and outbound dispatching problem with delivery time windows. *Engineering Optimization*, 44(7):859–874, 2012.

Nikolaos Kourentzes. Intermittent demand forecasts with neural networks. *International Journal of Production Economics*, 143(1):198–206, 2013.

Thomas L Landers, Melinda K Beavers, Malik Sadiq, and Don E Stuart. Software for dynamic reconfigurable order picking systems. *Computers & industrial engineering*, 27(1):245–248, 1994.

Peng Liu, Yaohua Wu, Chen Zhou, and Na Xu. Fluid-based slotting optimization for automated order picking system with multiple dispenser types. *Chinese Journal of Mechanical Engineering*, 24(4):529, 2011.

Ibrahim Masood and Adnan Hassan. An integrated mewma-ann scheme towards balanced monitoring and accurate diagnosis of bivariate process mean shifts. *Journal of King Saud University-Computer and Information Sciences*, 24(2):93–100, 2012.

Hiroshi Motoda and Huan Liu. Feature selection, extraction and construction. *Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol*, 5:67–72, 2002.

Marcus B Perry, Julie K Spoerre, and Tomas Velasco. Control chart pattern recognition using back propagation artificial neural networks. *International Journal of Production Research*, 39(15):3399–3418, 2001.

DT Pham and MA Wani. Feature-based control chart pattern recognition. *International Journal of production research*, 35(7):1875–1890, 1997.

Ussanee Purintrapiban and HW Corley. Neural networks for detecting cyclic behavior in autocorrelated process. *Computers & Industrial Engineering*, 62(4):1093–1108, 2012.

Vahid Ranaee and Ata Ebrahimzadeh. Control chart pattern recognition using a novel hybrid intelligent method. *Applied Soft Computing*, 11(2):2676–2686, 2011.

Malik Sadiq, Thomas L Landers, and G DON TAYLOR. An assignment algorithm for dynamic picking systems. *IIE transactions*, 28(8):607–616, 1996.

Xiaobing Shao. Recognition of control chart patterns using decision tree of multi-class svm. In *Advances in Intelligent Systems*, pages 33–41. Springer, 2012.

Alexander Shapiro. A dynamic programming approach to adjustable robust optimization. *Operations Research Letters*, 39(2):83–87, 2011.

Géraldine Strack and Yves Pochet. An integrated model for warehouse and inventory planning. *European Journal of Operational Research*, 204(1):35–50, 2010.

Sriram Subramanian. *Managing Space in Forward Pick Areas of Warehouses for Small Parts*. PhD thesis, Georgia Institute of Technology, 2013.

Ruud Teunter and Babangida Sani. On the bias of croston's forecasting method. *European Journal of Operational Research*, 194(1):177–183, 2009.

Ruud H Teunter and Laura Duncan. Forecasting intermittent demand: a comparative study. *Journal of the Operational Research Society*, 60(3):321–329, 2009.

Jeroen P Van den Berg, Gunter P Sharp, AJRM Noud Gademann, and Yves Pochet. Forward-reserve allocation in a warehouse with unit-load replenishments. *European Journal of Operational Research*, 111(1):98–113, 1998.

Uday Venkatadri, Vimalesh Vardarajan, and Biman Das. Product placement within a fast-picking tunnel of a distribution centre. *The International Journal of Advanced Manufacturing Technology*, 76(9-12):1681–1690, 2015.

Joseph L Walden. *Velocity management in logistics and distribution: lessons from the military to secure the speed of business*. CRC Press, 2005.

Peter Wallström and Anders Segerstedt. Evaluation of forecasting error measurements and techniques for intermittent demand. *International Journal of Production Economics*, 128(2):625–636, 2010.

Rico Walter, Nils Boysen, and Armin Scholl. The discrete forward–reserve problem–allocating space, selecting products, and area sizing in forward order picking. *European Journal of Operational Research*, 229(3):585–594, 2013.

Tony Wild. *Best practice in inventory management*. Routledge, 2007.

Thomas R Willemain, Charles N Smart, and Henry F Schwarz. A new approach to forecasting intermittent demand for service parts inventories. *International Journal of forecasting*, 20(3):375–387, 2004.

LI Yingdea and Jeffery S Smith. Dynamic slotting optimization based on skus correlations in a zone-based wave-picking system. 2012.

APPENDIX

Algorithm for Profiling and Slotting Optimization (PSO) of multi-mode Forward Area

Input: The SKU, order, rack, and facility data.

Output: Profiling and slotting optimization of the multi-mode forward Area.

* Import Data

Read the SKU, order, rack, and facility data.

* Fit test

For all pick modes $j > 0$
    For all SKUs
        Check if the SKU (eaches) fits the slot (=1) or not (=0).
        Check if the case fits the slot (=1) or not (=0).
    EndFor
EndFor

* Find the best case orientation, which results in space utilization

For all SKUs
    Find the optimal orientation in one Pallet and one lane of the Carton Flow Rack.
EndFor

* Find the parameters required for cost analysis

For all SKUs
    Find the SKU flow, the accumulated ordered quantity (demand) and order lines (picks) during the planning horizon.
    The No. of eaches/cases in Min No. of slots given to the SKU in the Carton Flow Rack.
    The No. of eaches/cases in one pallet of the Pallet Flow Rack mode.
EndFor
For all pick modes $j > 0$
    For all SKUs
        Find the No. of restocks during the planning horizon period, if Min No. of slot(s) is given.
    EndFor
    Find the volume of the shelf.
EndFor

* Cost analysis

For all i in Pallet Flow Rack Bays Range
    For all j in Carton Flow Rack Bays Range
        For all k in Bin Shelving Bays Range
            Find the Ave. picking cost from the reserve area.
            For all pick modes
                Find the Ave. picking and replenishment costs of the mode by travel distance calculation.

EndFor

If the total bays fit in the picking area

For all SKUs

Find the optimal number of slots given to the SKU in a Carton Flow Rack.

comment: (Refer to the algorithm for finding m_{ij} in next Appendix)

Find the picking cost from the reserve area.

EndFor

For all pick modes

Find the picking cost for the SKU if it is picked from the mode j.

Find the replenishment cost for the SKU if it is picked from the mode j.

Find the total cost for the SKU if it is picked from the mode j.

Find the savings by picking the SKU from the pick mode rather than the reserve area.

EndFor

While saving > 0

Find SKU x with the max savings by picking from mode y

If any slot(s) is available in the mode y

Assign the SKU x to the mode y

Exclude the allocated slot(s) to SKU x from the available slot of mode y

Get the associated costs of SKU x

Exclude SKU x

EndIf

EndWhile

Find the total cost of mode (i,j,k)

EndIf

EndFor

EndFor

EndFor

Find the optimal design (i*,j*,k*), which provides the Min total cost among all modes

* Export Data

Algorithm for finding m_{ij} (units of SKU i in mode j)

For all SKUs

For all pick modes

If j=1

$m_{ij} = \left\lfloor \frac{W_1}{w_i} \right\rfloor \varphi_{ij}\theta_{ij}b_i$

ElseIf j=2

$y_i = \frac{\sqrt{q_{ij}}}{\sum_{k\in A}\sqrt{q_{kj}}}V_2$

comment: (A is the set of SKUs)

$n_{1i} = \left\lceil \frac{w_i}{W_2} \right\rceil$ & $n_{2i} = \left\lceil \frac{y_i}{O_j} \right\rceil$

$n_i = \max(n_{1i}, n_{2i})$

comment: (n_i is the number of slots allocated to SKU i)

comment: (Find the number of lanes in carton flow rack given to SKU i)

$L_i = \left\lfloor \frac{W_j N_j^{SL}}{w_i} \right\rfloor$

$m_{ij} = L_i\varphi_{ij}\theta_{ij}b_i$

comment: (Find the number of cases of SKU i that is replenished in mode j)

ElseIf j=3

$m_{ij} = b_i$

EndIf

EndFor

EndFor

Dynamic SKU assignment and slot allocation in the forward area (heuristic $T.P.$)

Input: The generic MIP DFRP model's parameters.

Output: The dynamic SKU assignment and slot allocation in the forward area over time.

$r, p, k, E = 0$

For $(t = 1$ to $T)$

Read the initial inventory of the current layout of the forward area (I_{it});

Read the actual demand data for the current time;

Read the forecasted demand (d_{it}) of forecasting window ω;

For all SKUs in the forward area

$I_{it} = I_{i,t-1} - d_{it}$;

EndFor

$e_1, e_2, e, x = 0$;

For all SKUs in the forward area

If $I_{it} = 0$ & $I_{i,t-1} > 0$ **then** $e_1 = e_1 + 1$; **endIf**

If $I_{it} < 0$ **then** $e_2 = e_2 + 1$; **endIf**

EndFor

$e = e_1 + e_2$;

If $e > 0$ **then**

Run G_2 from chapter II to get SKU assignment and allocation;

comment: (Procedure A_4 was used in G_2 for ranking and space allocation:

A_4: $f_{4it} = \frac{d_{it}}{b_i}$ $le_{it} = \frac{p_{it}}{\sqrt{f_{4it}}}$ $y_{4it} = \frac{\sqrt{q'_{it}}}{\sum_{j \in A} \sqrt{q'_{jt}}} S$.)

Exclude all available SKUs in the forward area ($I_{it} > 0$) from the solution of G_2;

comment: (Replenish the empty slots with the first e allocated slots from the solution of G_2.)

For all SKUs in the forward area

If $I_{it} \leq 0$ **then** $I_{it} = I_{it} + a_i n_{it}$; **endIf**

EndFor

For all SKUs in the forward area

If $I_{it} < 0$ **then**

$I_{it} = 0$; $k = k + 1$;

comment: (k is the number of SKUs that leave the forward area and only the rest of order is replenished at time t.)

endIf

EndFor

EndIf

$E = E + e$; **comment:** (E calculates total number of emptied slots during T.)

EndFor

For $(t = 1$ to $T)$

For all SKUs in the forward area

If $I_{i,t+1} > I_{it}$

$r = r + 1;$ **comment:** (Find number of replenishments.)

EndIf

If $I_{it} > 0 \quad \& \quad I_{i,t+1} \quad \& \quad I_{i,t+1} < I_{it}$

$p = p + 1;$ **comment:** (Find number of picks from the forward area.)

EndIf

EndFor

EndFor

comment: (Calculate the picking and replenishment costs as below. P is the total picks during T)

Total cost=$c_1(p + E - k) + c_2(P - p) + c(r + k)$

TABLE 33

Costs for the data set with 10% Normal demand patterns

Static	T.P.	T.P.'	Dynamic
2481500	2315388	2440144	2284884
2402424	2311044	2438484	2282586
2517768	2306992	2464108	2277208
2436628	2299004	2441820	2272354
2449332	2340312	2477152	2304448
2702740	2279584	2455728	2253188
2430184	2305364	2452468	2277494
2474096	2305404	2435196	2276826
2482584	2348288	2462580	2312590
2721572	2338484	2493844	2300414
2463472	2337272	2453900	2306420
2485292	2324920	2480332	2288898
2694116	2290088	2451340	2262392
2497864	2320784	2505396	2285162
2733024	2307664	2457024	2278674
2718308	2297828	2449812	2270704
2522300	2307184	2455260	2278908
2413560	2333584	2461112	2298760
2416264	2335664	2467100	2300094
2550388	2324164	2452040	2290320
2698772	2286768	2422408	2260018
2693116	2279912	2428772	2253176
2731624	2267820	2438228	2240886
2688704	2290572	2435848	2259078
2707868	2292592	2445780	2259292
2724216	2290672	2445320	2257718
2702808	2302244	2443776	2270842
2723256	2329208	2457880	2294852
2674760	2298320	2445672	2265858
2709352	2283036	2431884	2256514
2468772	2357732	2483912	2322788
2736940	2334208	2485440	2300058

TABLE 34

Costs for the data set with 30% Normal demand patterns

Static	T.P.	T.P.'	Dynamic
2125212	2052092	2200612	2042148
2129772	2042088	2193208	2032772
2136280	2041940	2203532	2032656
2125688	2060604	2184272	2050874
2150548	2062272	2221596	2050590
2378284	2032332	2164024	2022964
2139260	2030876	2167736	2022056
2179524	2059868	2205816	2049722
2145128	2059396	2205676	2048932
2389248	2052392	2185124	2041314
2141212	2048308	2200180	2038624
2141212	2070236	2215972	2058046
2372600	2046748	2178292	2037496
2145116	2090800	2222988	2079024
2366632	2069400	2208896	2059480
2368680	2007248	2161576	1997724
2147432	2038776	2198376	2029680
2126596	2102876	2225944	2090920
2151696	2087480	2223652	2076570
2192436	2108168	2237488	2096134
2363780	2029612	2173632	2019712
2316004	2086552	2213936	2076434
2379132	2076504	2236384	2065332
2366500	2094684	2238680	2082014
2373224	2086388	2234072	2073842
2366028	2086944	2237408	2075910
2399964	2089768	2244208	2076904
2376912	2090832	2239400	2078682
2383340	2064136	2193256	2052898
2392788	2026884	2169468	2017052
2166756	2079680	2237260	2068588
2396012	2087292	2243236	2074240

TABLE 35

Costs for the data set with 50% Normal demand patterns

Static	T.P.	T.P.'	Dynamic
2396012	2087292	2243236	2055770
2031964	1888808	2049800	1865248
2025520	1886380	2036184	1862806
2014800	1877500	2033716	1853904
2008388	1910084	2063396	1885000
2158820	1902224	2056964	1878262
2007744	1880968	2038396	1857778
2019548	1879936	2037828	1857156
2019888	1915104	2063792	1889248
2157356	1925768	2080852	1899916
2169588	1906900	2065344	1882536
2023596	1914996	2073412	1889916
2165800	1911732	2061964	1887360
2021300	1887060	2042228	1863088
2155368	1900184	2053452	1876204
2153404	1891700	2057720	1868134
2028124	1878020	2033168	1855178
2007444	1908312	2047556	1883614
2020892	1897972	2048656	1873648
2030784	1901296	2058216	1877326
2163140	1899564	2057544	1875240
2169924	1885872	2039500	1862302
2131352	1880960	2048756	1857760
2158028	1911224	2067620	1885710
2144760	1904880	2066488	1880540
2169212	1929328	2075648	1903078
2172876	1912460	2069252	1887370
2160112	1920904	2073436	1896152
2184104	1929956	2079612	1904858
2158132	1894148	2054912	1870536
2012304	1908520	2058816	1884876
2153912	1898104	2058892	1873776

TABLE 36

Costs for the data set with 70% Normal demand patterns

Static	T.P.	T.P.'	Dynamic
1827840	1725976	1872908	1704174
1845084	1740048	1889936	1719126
1831956	1724644	1879392	1703908
1832564	1731508	1879412	1709632
1843696	1741960	1894932	1722018
1842600	1733692	1891140	1712196
1841488	1717028	1873148	1696646
1839588	1742596	1897052	1719848
1835548	1731820	1881308	1711838
1828532	1720408	1867260	1701112
1833904	1727128	1873580	1706806
1838532	1732636	1881284	1713480
1829852	1740816	1889100	1718386
1828372	1737912	1891456	1716990
1838248	1727880	1880492	1707938
1838688	1724456	1877764	1707488
1839128	1730024	1880884	1708428
1830088	1742532	1893376	1717664
1830088	1722584	1873612	1704696
1830576	1735032	1879388	1712898
1833064	1745336	1888596	1723554
1838048	1726416	1873012	1705120
1826044	1716796	1869812	1697060
1841788	1743976	1890572	1721548
1836156	1736252	1885064	1714664
1830304	1729724	1878644	1709482
1832020	1734588	1876584	1713946
1828328	1731344	1875592	1711896
1823660	1720952	1866492	1701050
1828152	1728572	1874812	1706036
1837036	1733396	1886116	1712980
1830680	1722032	1866092	1704144

CURRICULUM VITAE

NAME:	Parvaneh Jahani
ADDRESS:	Industrial Engineering Department University of Louisville, KY
DOB:	Tehran, Iran-8/24/1986
EDUCATION & TRAINING	- B.S., Textile Engineering Tehran Poly-technique, 2004-2009 - M.S., Industrial Engineering Sharif University of Technology, 2009-2012 - Ph.D., Industrial Engineering University of Louisville, 2012-2016
HONOURS AWARDS:	- Recipient of Material Handling Education Foundation, Inc. (MHEFI) scholarship, (March 2015). - Recipient of Grosscurth Ph.D. Fellowship, University of Louisville, Louisville, KY (Aug. 2012-Aug. 2014). - Recipient of Scholarship from American Society for Quality, (Feb. 2015). - Graduate Student Council (GSC) research scholarship, awarded by school of interdisciplinary and graduate study, University of Louisville, Louisville, KY (Nov. 2014).

PROFESSIONAL

SOCIETIES: - Research and Development Intern, Optricity Corporation, Research Triangle Park, NC (May. 2015-Sep. 2015).

- Operations and Solution Analyst, Honeywell Corporation, Mason, OH (June 2016-Current).