A STORAGE ASSIGNMENT PROBLEM WITH MULTI-STOP PICKING TOURS IN AN AUTOMOTIVE SPARE PARTS WAREHOUSE

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By
Esra Aybar
January, 2008

"A STORAGE ASSIGNMENT PROBLEM WITH MULTI-STOP PICKING TOURS IN AN AUTOMOTIVE SPARE PARTS WAREHOUSE" by Esra Aybar
Bilkent University – Institute of Engineering and Science
January 24, 2008

I certify that I have read this thesis and that in my opinion it is full adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

Prof. Dr. Barbaros Tansel (supervisor)

I certify that I have read this thesis and that in my opinion it is full adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

Prof. Dr. Ihsan Sabuncuoğlu

I certify that I have read this thesis and that in my opinion it is full adequate, in scope and in quality, as a dissertation for the degree of Master of Science.

Prof. Dr. Cevdet Aykanat

Approved for the Institute of Engineering and Science

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Science

ABSTRACT

A STORAGE ASSIGNMENT PROBLEM WITH MULTI-STOP PICKING TOURS IN AN AUTOMOTIVE SPARE PARTS WAREHOUSE

Esra Aybar
M.S. in Industrial Engineering
Supervisor: Prof. Dr. Barbaros Tansel
January, 2008

In this study, a storage assignment problem for an automobile spare parts warehouse is considered. Incoming items from the suppliers are located at storage slots in pallet loads with single stop storage tours while outgoing items requested from the company's service centers are collected on a daily basis with multi-stop pick tours. The problem involves seeking a layout (defined by an assignment of items to storage slots) to minimize the total distance items are moved from the receiving dock to storage slots and from storage slots to the shipping dock. The items requested each day are not the same. Consequently, the number and locations of pick stops in pick tours differ from day to day. This feature of the problem sufficiently complicates the structure to make analytical approaches difficult to use. Two simulation models are developed, one single-level and one multi-level model, to investigate factors that have some effect on the storage assignment problem under consideration. Various insights are obtained for a set of storage assignment alternatives for both models.

Keywords: storage assignment problem, warehouse, simulation models

ÖZET

ÇOK NOKTALI TOPLAMA YOLLARI OLAN OTOMOTİV YEDEK PARÇA DEPOSUNDA ÜRÜNLERİN DEPO RAFLARINA ATANMASI PROBLEMI

Esra Aybar
Endüstri Mühendisliği Yüksek Lisans
Tez Yönrticisi: Prof. Dr. Barbaros Tansel
Ocak, 2008

Bu çalışmada, otomobil yedek parça deposunda ürünlerin raflara atanması problemi ele alınmıştır. Tedarikçilerden gelen ürünler palet birimleri ile tek tek raflara yerleştirilmektedir. Rafa yerleştirme turları tek noktalı yollar ile tamamlanmaktadır. Yetkili servislerden gelen yedek parça siparişleri ise günlük olarak çok noktalı toplama turları ile toplanmaktadır. Burada gelen ürünleri raflara yerleştirme ve giden ürünleri raflardan toplama esnasında katedilen toplam mesafeyi minimize edecek ürün-raf atanması yapılmak isteniyor. Hergün sipariş edilen ürünler aynı değildir. Bundan dolayı, toplama turlarında uğranılan nokta sayısı ve yerleri günden güne değşmektedir. Bu özellik, probleme analitik çözümler getirmeyi güçleştiriyor. Böylelikle, biri tek kat raflı, diğeri çok kat raflı iki tane benzetim modeli kurulmuştur. Bu modellerle sözkonusu ürün-raf atanması problemini etkileyen faktörlere dikkat edilmiştir. Değişik atama alternatifleri incelenerek bir takım sonuçlar çıkarılmıştır.

Anahtar Kelimeler: ürün-raf ataması problemi, depo, benzetim modelleri

Acknowledgement

I would like to start by thanking my supervisor Prof. Dr. Barbaros Tansel for sharing his experience and deep knowledge with me throughout the study, and for motivating me at all times. I learned a lot from Dr. Tansel. Thank you!

I send my greatest appreciations to my thesis jury for sharing their time and thoughts with me.

Second, I would like to thank my parents for being right next to me at every moment, even when they were far away. They were either on the other end of the phone for hours, trying to calm me down at every break down I lived, or they were on the plane flying here to support me with a tight hug. I am lucky to have parents such as them!

My sister, Sera Aybar, was a great roommate during my last year of study and a best friend always. She was my sister, my friend, my mom, my dad for me here when I was happy, alone, ill, hopeless…She was my assistant when I panicked that nothing was going to finish on time. She is my little sister and do not know what I would do without her. I owe her the greatest thanks!

I would also like to thank my dearest and best office mates Konul Bayramoglu and Ipek Keles for their friendship, care, and love they gave every day in the office. I really

appreciate their cheering up, listening, and sharing experience and thoughts with me at out "chat corner" in the office. They were there with me at every step I took.

I would also like to thank my other office mates for cheering the office up, and making it a better place to study and to relax from time to time!

I would like to thank all my other friends from other offices for stopping by and motivating me throughout the day.

I would, next, like to thank each and every one of my friends in Kaytarikcilar, most of which are now abroad or working out of Ankara. I would especially like to thank Mehmet Mustafa Tanrikulu for organizing the best activities and bringing all of us together. We would not have been "Kaytarikcilar" if it was not for him. I would like to thank him for his kindness, help, and support for both my academic and social activities. He was there when I first learned to ski, he was there when I first coded my macro…He is a best friend everybody should have! I would like to thank Gulay Samatli for the cheer she always gave to me. She was with me from the very first step until my thesis submission, even when she moved all the way to the other side of the world. She is a real friend, and I am very happy to have her as one of my best friends. Last from Kaytarikcilar, I would like to thank Seda Elmastas, Banu Karakaya, Erdinc Mert, Mehmet Fazil Pac, Evren Kahramanoglu, Murat Kalaycilar, Nurdan Ahat for the wonderful organizations we did together and for sharing their thesis experiences with me when I was at that stage.

I would like to thank Dr. Ugur Yuregir for helping me believe that I can do this, for convincing me to get back to it at the time I had decided to quit, for helping me discover myself and listen to myself despite what others say, for giving me advice on people and the real world that I mostly could not see. I learned a lot from you! Thank you!

Finally, I would like to thank all the others who were with me in some part of the way.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

The problem in consideration for this thesis is a storage assignment problem. A simple but precise definition of the storage assignment problem can be stated as “The problem of determining an appropriate storage policy and assigning the items to storage slots accordingly”.

When looking at the storage assignment problem on its own, the warehouse layout, routing policy, and any other organizational issues are accepted to be known and therefore do not lie in the scope of the problem in consideration. More precisely, the storage assignment problem’s boundaries begin with the incoming of items to a warehouse for which a specific layout has been *a priori* defined and end at the point where the order pickers start their order picking routes to pick the located items within the warehouse according to a specified routing policy. Therefore, the results derived from the storage assignment problem affect the order picking process directly, which is the major efficiency performance measurement within a warehouse. This attaches a significant importance to the assignment problem.

The problem environment specifically in this thesis is a spare parts warehouse for an automotive company in Turkey serving the company’s own automobile repair\retail centers. The company’s identity is not disclosed due to a privacy contract signed between the company and the researcher. Given the warehouse layout, routing policy, number of storage slots required by each item type, incoming item data from suppliers, and outgoing item data according to customer orders, a storage assignment configuration will be determined under dedicated storage policy. With the warehousing overview obtained from the real problem environment two simulation models are developed where one is for single-level storage systems and the other for multi-level storage systems. Various storage assignment configurations with respect to dedicated storage assignment policy is implemented to both models and their performance is measured in terms of total distance traveled for the single-level system and total travel time for the multi-level system in a 30-days time period. An insight of the factors the storage assignment structure effects is derived from these results. Verification and validation is done for the simulation models in order to guarantee accuracy of the results and, so, the derivations.

The following section of this chapter overviews the warehouse terminology, its operations, and research areas. The next chapter gives a summary of the warehouse related problem areas in the literature and how different researchers have dealt with the problems. Following that, Chapter 3 gives a precise explanation of the problem environment in this study. Chapter 4 describes the two simulation environments and the models designed for them. This is followed by Chapter 5 where the different storage assignment configurations are explained and their results are stated together with related comments. Chapter 6 gives the verification and validation procedures for the simulation models and a general conclusion of the study is presented in Chapter 7.

1.1. General Overview of Warehouse Operations

As population grows and the world becomes a more global environment, companies are faced with a dramatically increasing demand for their products or services. They depend on warehouses in order to overcome loss of sales and reply faster to customer demand. De Koster et al. (2007) give a more detailed list of company missions warehouses contribute to.

These are

- achieving transportation economies (e.g. combine shipping, full-container load),
- achieving production economies (e.g. make-to-stock production policy),
- taking advantage of quality purchase discounts and forward buys,
- supporting the firm’s customer service policies,
- meeting changing market conditions and uncertainties (e.g. seasonality, demand fluctuations, competition)
- overcoming the time and space differences that exist between producers and customers,
- accomplishing least total cost logistics matching with a desired level of customer service,
- supporting just-in-time programs of suppliers and customers,
- providing customers with a mix of products instead of a single product on each order (i.e. consolidation)
- providing temporary storage of material to be disposed or recycled (i.e. reverse logistics)
- providing buffer location for trans-shipments (i.e. direct delivery, cross-docking)

Warehouses are usually large rectangular shaped buildings with products stored in them until requested by a customer. Different warehouses can serve for different purposes. Examples include finished products warehouse in a manufacturing company storing items for customers in the market, raw material warehouse for a manufacturing plant with material stored to be fed to the production environment, spare parts warehouse where parts of the finished product are stored until demanded by the company’s local service\repair centers. The storage assignment problem considered in this thesis is for warehouses of the third kind.

Indifferent from the product type it stores and the customers it serves, every warehouse has its own processes, resources, and organization. Rouwenhorst et al. (2000) defines the *processes* as the steps the products arriving at a warehouse are taken through.

They define *resources* as all means, equipment and people, needed to operate a warehouse and *organization* as all the planning and control procedures used to run the warehouse. In addition to Rouwenhorst et al (2007)'s definitions, a warehouse also has its *entities* that are the actual parts moving throughout the warehouse, going under processes, and using resources according to the organization. Each *item type* arriving, being stored and leaving the warehouse is an entity and a combination of item types in various quantities requested by a customer is an *order*.

It is convenient to define the warehouse resources first since they are used to carry out the processes. The *storage system* can be considered to be the main resource of a warehouse, as it serves the most important process (i.e. storage). It is the group of physical components such as shelves, racks, or automated systems to store the products in the warehouse. Each storage space available for item location on a storage system is referred to as a *storage slot*. The storage system may be either a *single-level system* where the storage slots are at the floor level or a *multi-level system* where the shelves are placed on top of each other to form storage slots higher than the floor level. This results in differences in the material handling equipment used during the picking process and also in the storage assignment decision in some cases where there are constraints on assigning an item to a specific level of the storage system (e.g. heavy items cannot be placed at top levels).

The storage\retrieval of items to\from the storage system is done with the *material handling devices*, mostly forklifts and hand trucks in manual systems but conveyors and AS\RS robots in automated systems. Other equipment supporting the order picking process can be referred to as *order pick equipment*, such as the barcode scanners used to input data to computer system.

A *computer system* is necessary for keeping the inventory data, along with assigning incoming items to storage slots, detecting storage locations of outgoing items for picking, and any other necessary data related to the warehouse.

Finally, the *workers* are one of the most important resources of a warehouse as they are the ones to perform and control the warehouse processes.

Warehouse processes begin with *receiving* which is the arrival of items to warehouse from international or local suppliers. This step includes the opening, checking, and repackaging of the items.

Followed by the receiving of items is the *storage* of them within the warehouse. This is the process where items are placed in storage slots in the storage area. In some warehouses, the storage slot the item will be placed in is not known *a priori* but is determined after receiving and so the storage slot assignment step can be included as a part of the storage process.

Order picking, another warehouse process, is the retrieval of items from their storage locations. This can be done either manually by material handling equipment such as forklifts, hand trucks, etc. or automatically by automatic storage and retrieval systems (AS\RS). The systems in which order picking is done manually by going to each item's location are referred to as *picker-to-parts systems* whereas systems where picking is done automatically and so the item's are retrieved and carried to the picker by robots are called *parts-to-pickers systems*. There are also systems running partly automatic where conveyors carry manually retrieved items from storage locations to the order preparation area of the warehouse.

The last process carried out in a warehouse is the *order preparation* where the items are boxed according to the orders received by customers. Loading these boxes on to trucks is also included in this phase.

The organization of a warehouse is important since it determines the run and control of the warehouse. Each process within the warehouse has its own organizational policies. One of the major ones is at the storage process stage where items must be assigned to storage slots depending on a specific *storage assignment policy*. One extreme of this policy is *randomized* *storage policy* where items can be assigned to any storage slot without any pre-defined criteria. The other extreme is the *dedicated storage policy* where a certain number of storage slots are reserved for a specific item type and cannot store any other item. *Class based storage policy* is in between these two extremes and can be described as the dedicated storage policy of item classes. This means that a number of storage slots are dedicated to a class of items rather than an individual item and each item of that class can be randomly assigned to a storage slot within the dedicated region. There are many other variations of these three policies described in the literature (e.g. Petersen & Aase (2003), Ruben & Jacobs (1999))

To regulate the order picking processes various decisions must be made previously. These decisions take part in the organization of the warehouse. One decision to be made is whether or not to divide the picking area into regions and assign an order picker to each region individually. The division into regions can be done by a *zoning policy*. One is the *pick-and-pass system* where the order picker, after his own picking process, passes the cart with the picked items and the order pick list to the next order picker in the next zone. The other is the *parallel picking system* where all order pickers in all zones pick their own part of the orders simultaneously. A second decision to be made is to determine whether the orders are picked one by one (*single order picking*) or in batches (*batch picking*). A third decision arises if batch picking is done, which is how to sort the orders for shipping. This can be done either when the orders are being picked (*sort while pick*) or after the orders are picked during the order preparation process (*pick and sort*). Finally, a *routing policy* must be determined to define the sequence of items to be picked and the route to be followed during the order picking process. Different routing policies have different inferences on the order pickers. The *S-shape heuristic*, for example, is the simplest heuristic for routing order pickers and states that any aisle containing at least one pick item is traversed entirely. With *the return method*, however, the order picker enters and leaves the aisle from the same side meaning that he does not have to traverse the entire aisle if not necessary. De Koster et al. (2007) give descriptions of other routing policies and a review of papers including them.

The effective determination of these policies results in a warehouse working with high performance. There are various performance measures for any warehouse and are listed in De Koster et al. (2007) as

- total travel distance during order picking,
- total warehouse operating cost,
- throughput time of an order,
- overall throughput time to complete all batch of orders,
- use of space,
- use of equipment,
- use of labor, and
- accessibility of items.

Over the years, warehousing has been of great interest to researchers and is widely studied in the literature. Due to its complex structure with many processes and organizational issues, solutions to different warehouse problems have been derived independently. The main warehouse problems studied in the literature are the *warehouse layout problem* (functional area size determination, picking and cross aisle orientation, dock locations, dwell-point selection), *storage assignment problem* (assignment of items to storage slots), and *routing problem* (optimal route selection for order pickers). Other problems such as *batching* (combination of orders for easier and smoother pick routes), *zoning* (division of the warehouse into pick zones), etc. have also been studied but have not received as much attention as the previously stated three. Different researchers have presented different solution methodologies for these problems. The simulation approach is the most common one, where the warehouse environment is presented as a simulation model and various organizational policies are implemented. Results are compared according to the simulation runs. Developing a mathematical model for the problem in consideration is another approach. However, most models presented are non-linear and cannot be solved optimally within reasonable CPU time.

For this reason, various algorithms have been generated by different researchers for reaching a near optimal solution.

Chapter 2

LITERATURE REVIEW

Warehouse design and control has been widely studied by many researchers for many years. Different studies have considered the topic from different perspectives, some with a very broad view while others focusing on a more specific problem of the general topic. Rouwenhorst et al. (2000) reviews papers related to warehouse design and control and present a framework for what must be done from the beginning when designing a warehouse. The authors state what decisions must be taken during the design stage of a warehouse. It is seen that many problems, each with a different objective, must be handled. The paper divides warehouse design problems into three (strategic level, tactical level, operational level) according to the time spam the decision will be kept in use and the investment level it requires. Details of specific problems under these levels are given in the paper. Following this framework of decisions regarding warehouse design, the authors name studies from the literature related to each level of decision groups. Another paper presenting framework for warehouse design is Hassan (2002) where a step-by-step procedure is given as to how to deal with the warehouse layout problem which includes the assignment of items to storage locations, arrangement of functional areas of the warehouse, determination of the number of and location of I\O points, determination of the number of aisles, their dimensions, and orientation, estimation of the space requirements, design of the flow pattern, and formation of the pick zones. Muller (1989) is another example for the class of papers dealing with the review and framework of warehouse design. It discusses modeling issues related to warehouses in general, and what are the important concepts that must be considered when a warehouse is being modeled in any matter. Modeling objectives and approaches are presented in his work.

To look at the warehouse design and control subject in more detail, many problem areas have been defined by different researchers under this concept. Problem areas included in warehouse design can be listed as:

- Equipment selection
- Process flow design
- Functional areas size determination
- Equipment and employee requirement specification
- Allocation of incoming goods to storage locations
- Batch formation or order sequencing
- Assignment of picking tasks to pickers
- Routing of pickers
- Dwell-point selection
- Aisle orientation
- Dock assignment
- Lane assignment

None of these problems can be handled alone, and it can be seen from the literature that most of the time three or four of them are considered together. For example, functional area size determination, dwell-point selection, and aisle orientation are mostly viewed as the “warehouse layout problem“. Another group involves the allocation of incoming goods to storage locations, batch formation (order sequencing), assignment of picking tasks to pickers, and routing of pickers. This group of problem is referred to as the “storage assignment problem” in general. The storage assignment problem can further be divided into two classes with one class assuming that only a single pick stop is present in a picking tour (i.e. single-command) while the other class assumes multiple pick locations in a picking tour (i.e. multi-command). Our research deals with the work on multi-command storage assignment problems.

Before looking at the literature of storage assignment problems, there are a couple of papers in the area of warehouse layout problems that are worth mentioning. The first is Caron et al. (2000a) where a simulation based method is proposed for the problem under consideration, aiming to minimize the total travel distance by defining an efficient layout. So, they present a simulation approach for defining an efficient layout design of the picking area in a picker-to-part system with COI-based storage policy. "Layout design", here, implies determining the number and orientation of aisles within the warehouse. The efficiency of the layout is measured by the picking travel distance. It is an important study, because it gives a nice overview of the types of layout that can be implemented in a warehouse and their efficiencies.

Roodbergen and Vis (2006) give a mathematical formulation for a warehouse where the order picking is done manually between a number of parallel aisles. The paper aims to give an approach to determine the layout for the order picking area of the warehouse, so that the average travel distance of a picking tour is minimized. First, an estimation for the travel distance is formulated which depends as parameters on the routing policy and number of products to be picked in a tour; and as variables on the number of aisles, length of an aisle, and depot location. Two formulations are given in the paper dependent on the routing policy, one for the s-shape policy and one for the largest gap policy. Then, using this formulation as an objective function, a non-linear mathematical model is presented to determine the optimal layout.

Another study in the field of layout design is Caron et al. (2000b) where an optimal layout is defined to increase the picking system efficiency (i.e. minimize total picking time). The main system parameters affecting the layout are listed as the total length of the picking aisles, number of picks per tour, and the shape of the COI-based ABC curve in the paper. According to this, a formula relating the optimal number of aisles to the listed parameters is presented. A travel distance formulation is derived and the impact of layout on the expected travel distance is examined by a simulation model.

When looking at the literature on multi-command storage assignment problems we see that there are a number of papers that compare different policies for the storage, routing, and picking operations. Hsieh and Tsai (2006) is one of them. They state the factors that affect order picking efficiency as warehouse layout design, storage assignment policy, picker routing policy, and combination of order. Their work provides a simulation model that can be used to see the affects of cross aisle quantity, storage assignment strategy, order combination, order picking policy, and picking density within an aisle on the order picking efficiency which is measured in terms of average order picking distance. Their work differs from most of the policy comparison papers in the sense that it combines the elements of the layout problem with the storage assignment problem.

Routing policies (traversal, return, largest gap, Z-pick, and etc.) are most frequently reviewed in the comparison papers, and their effects on the system efficiency are discussed. Some notable papers that review different types of the routing policies are Manzini et al. (2007), Petersen and Aase (2003), Caron et al. (1998), Petersen (1999), Petersen and Aase (2004), and Ruben and Jacobs (1999). Each of these papers includes the comparison of other operational polices as well and measures the system performance differently. Manzini et al. (2007) identifies and measures the principle impacts of alternative policies and configurations (routing, sequencing, scheduling, and order batching policies) on the response throughput time (i.e. total picking cycle time which consists of time at the I\O point at the start and end of each tour, processing time, and travel time between pick locations). Petersen and Aase (2003) examine the effects of applying different picking, storage, and routing policies on the current system. Together with this, sensitivity analysis are performed to see the effect of order size, warehouse shape, location of I\O point, and demand distribution on the system performance which is measured in terms of order picker travel distance. Caron et al. (1998) compares different routing strategies, namely traversal and return, and storage policies based on the COI index for low-level picker-to-part systems. The work presents analytical models which assess the expected travel distance required to fill an order for different routing policies and associated storage policies. Petersen (1999) compares various routing heuristics with the optimal routing strategy developed by Ratliff and Rosenthal (1983), compares the performance of volume based storage and random storage, and examines the impact of travel speed and picking rates on routing and storage policy performance. While doing so, routing and storage policies are compared in terms of total time required to complete a given pick list which includes travel time, time for identifying the storage location and product, time for picking the correct quantity from the pick location, time for confirming the pick on the pick list, and time for placing the items into the pick cart.

Petersen and Aase (2004) present another work in the area of policy comparison where they focus only on storage policies and evaluate the performance of class based storage policy in a warehouse with multiple picking aisles where picking is done manually in a sort-while-pick order picking fashion. The performance of class based storage policy is also compared with that of volume based and random storage policies.

Ruben and Jacobs (1999) is slightly different from the other examples since it compares alternative polices in a multiple server environment. They deal with a warehouse where batch picking is performed by multiple pickers traversing the aisles. The importance of batching is laid out as to improving the system performance and so different batch construction heuristics and storage assignment strategies from the literature are taken into consideration.

Besides policy comparison studies for multi-command storage assignment problems, many optimization approaches for a specific problem definition were studied in the literature. Different authors defined various problem settings and proposed different methods for the solution. The three most frequently used modeling approaches in the literature are simulation models, mathematical programming models, and queuing models. All of the models have travel time, travel distance, or order picking cost minimization as an objective.

A couple of examples can be named under the title of simulation approach to the multi-command storage assignment problem with travel time minimization as an objective. Dallari et al. (2000) considers different storage policies, namely class-based and shared storage, while applying two tour construction heuristics (two band and band insertion heuristics) where there are multiple picks points traveled by an automated storage-retrieval system (AS/RS). The objective is to minimize the AS/RS travel time which is defined as a function of the shape of the storage area, the number of picking points and the sequencing algorithm. Simulation runs are taken for each combination of these variables and the results show that the number of picks per tour is the most influencing factor on the AS/RS travel time followed in second place by the influence of the storage policy. Hsieh and Tsai (2001) is another study that can be grouped in this class, but, is also different since it presents a different approach to the class-based storage strategy and uses simulation as a method to justify the efficiency of the proposed approach. They identify a methodology that is based on the bill-of-material (BOM) of an item that allows the items to be grouped into classes. This is more of a database system where each item is assigned a code according to its BOM properties and so grouping is done by these codes. The proposed method, therefore, carries the properties of the class-based storage and also can be implemented into a computer integrated manufacturing (CIM) system where the system can define the classes on its own and locate the items to storage bays accordingly.

When we look at the work studying the multi-command storage assignment problem with a mathematical model, we see that no paper can provide a mathematical formulation that can be solved for all instances in polynomial time. Most of the authors, therefore, present an analytical formulation and propose a heuristic to solve the problem. A very early work on multi-command storage assignment problems is Malmborg and Krishnakumar (1989). They identify a storage assignment policy that will minimize the order picking cost in a warehouse system. Their system works under the dedicated storage policy with multi-command order picking done by man aboard vehicles each serving for a specific aisle and I/O point. The problem is solved in two stages where first, the items are assigned to aisles and then each item is located within its own aisle to a storage slot. While assigning the items to aisles, Malmborg and Krishnakumar propose to balance the workload of each man aboard vehicle and use the COI rule when locating the items of the shelves in aisles. Proofs are given as to why these proposed methods actually work for the problem in question. Le-Duc and De Koster (2005), for example, give a probabilistic formulation to estimate the average travel distance of a picking tour in a warehouse where there is a manually operated order picking system with multiple narrow aisles which store items under the class-based storage policy. Classes are formed according to their order frequencies where each item in a class has the same order frequency. The analytical formulation for the travel distance depends on the parameters of the layout (length of an aisle, total number of aisles, number of classes, width of a cross-aisle, and center-to-center distance between two consecutive aisles), the pick frequency, the storage assignment scheme, and the pick list size. Then, the problem of determining the optimal storage space (i.e. storage zone boundaries) for each class in an aisle is formulated by a mathematical formulation where the objective function is the average travel distance. A heuristic is proposed for solving the non-linear formulation.

One of the few multiple server multi-command storage assignment problem papers is Jewkes et al. (2004). It assumes to have a warehouse where there are multiple pickers in each aisle and a “pick and pass’ strategy is applied. Under this strategy, each picker is assigned to a region of the aisle and collects only the items that are in his\her own zone and passes the material handling container to the next picker who continues to collect the items of the picking list. The issues of interest are product location, allocating products to each picker, and picker home base location (each picker has a home base where the picking process begins in his\her own zone) with the objective of minimizing the expected order cycle time. The authors provide algorithms to solve these problems. For fixed product locations, a dynamic programming algorithm is developed which determines the optimal product allocation and server locations.

The last type of papers seen in the literature of multi-command storage assignment problem with a mathematical approach is the ones that aim to minimize the order picking cost. An attempt to develop mathematical models for both the warehouse layout and storage assignment problems simultaneously is performed by Heragu et al. (2005). The study considers warehouses with five functional areas: receiving, shipping, stages for cross-docking operations, reserve and forward areas. Their problem is to determine the functional area sizes and product allocation in order to minimize the total material handling cost. A mathematical formulation is provided for this problem and the model defines the flow to which each product must be assigned and as a result the size of the functional areas within the warehouse. A heuristic is presented to solve the model for large instances.

Some authors prefer to tackle the multi-command storage assignment problem under queuing model reasoning. An example for the papers in this area is Chew and Tang (1999). They assume a manual order picking system in a rectangular warehouse with a multi-command order picking policy where order batching is applied. The exact probability mass functions that define the tour of an order are given and their first and second moments are derived with an associated tour. The order picking system is then modeled as a queuing system and allows the system performance to be measured under various order batching and storage allocation strategies.

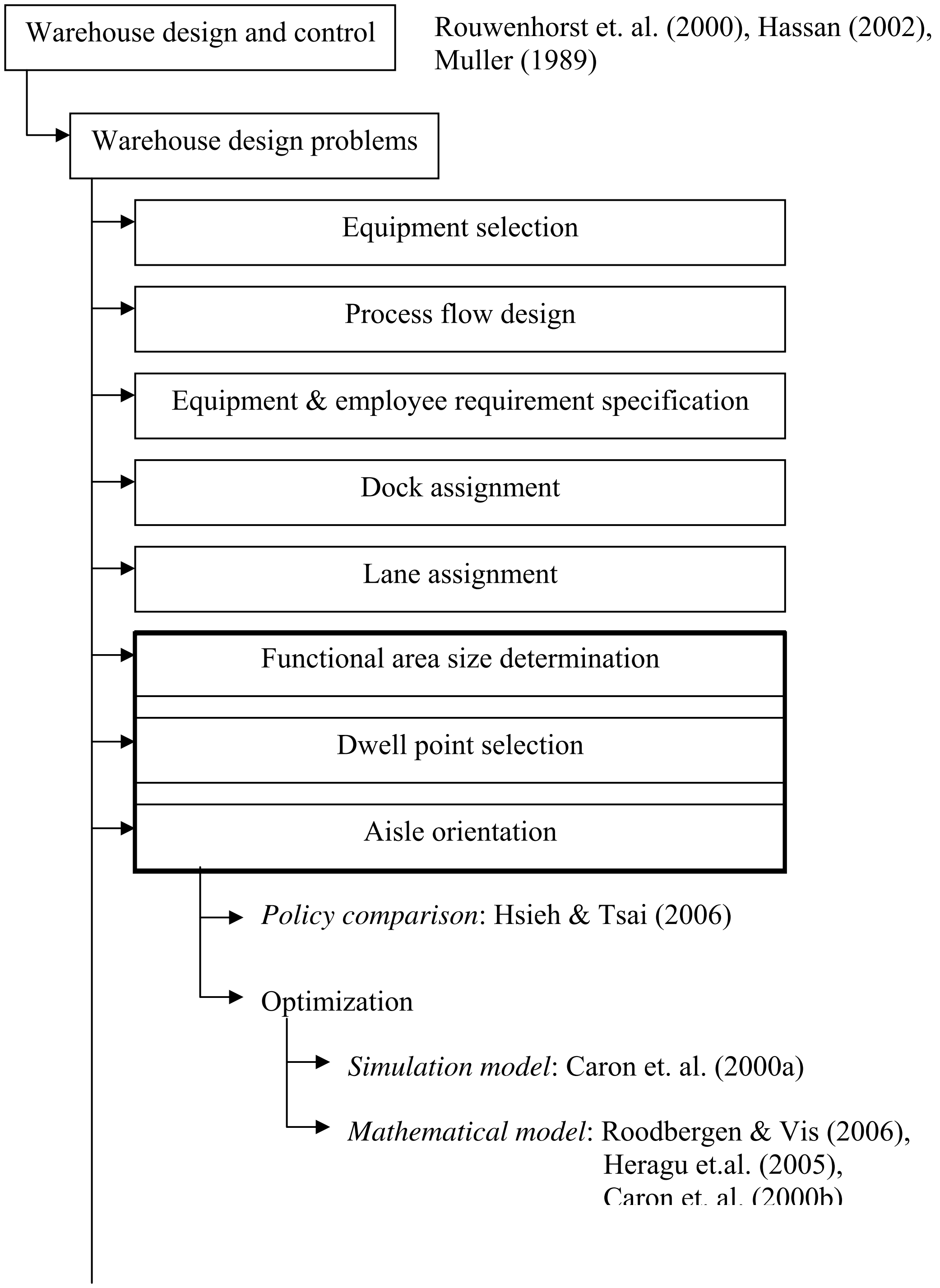
All papers in the warehouse design and control literature evaluate the warehouse performance efficiency in some measure of the “picking tour”. That is, the performance of a warehouse is determined either by the travel time, travel distance, or travel cost of a picking tour. Also, it was observed that the order picking process consumes nearly 60% of the labor activities in a warehouse (De Koster & Der Poort (1998)). Due to these reasons the routing of order pickers in a warehouse is worth studying. Many researchers have concentrated on this area only and presented work related to “routing optimization”. Literature in this field mostly tend to represent the warehouse as an undirected graph with vertices being the item locations and end points of aisles; arcs representing the time, distance, or cost of travel between nodes. Different papers present different algorithms for the order picking route.

Ratliff & Rosenthal (1983) has presented an algorithm for the routing problem in a rectangular warehouse with cross-aisles only at the ends of the aisles. Given an undirected graph as described above, they generate a sub-graph where a minimum length picking tour route can be generated. Following this, an algorithm for generating a minimum length tour from this sub-graph is given. The procedure computation effort defined by Ratliff & Rosenthal (1983) is linear in the number of aisles. This paper is a widely accepted one in the literature as it gives an optimal solution to warehouse routing problem. Many researchers use Ratliff & Rosenthal's algorithm to compare their own algorithms.

De Koster & Der Poort (1998) defines the problem by the following question: "Given that the order picker has to collect a number of products in specified quantities at known locations, in what sequence should the order picker visit these locations in order to minimize the distance traveled?". They consider three types of warehouses differing in aisle width (narrow or wide aisles), shelf height (single store shelves or multi-store shelves), and depot positions (centralized or decentralized depositing). An undirected graph similar to the one defined above is presented by the authors. They extend the optimal algorithm of Ratliff and Rosenthal (1983) for warehouses with a central depot to an algorithm working for warehouses with decentralized depositing as well. So they guarantee an efficient algorithm for all three warehouse cases they present. The performance of the new algorithm is compared with that of the S-shape heuristic (for all three warehouse structures) since it is the most commonly used one in practice.

A similar graph representation is given in Roodbergen & De Koster (2001) for a warehouse with a middle cross aisle (i.e. 3 cross aisles warehouse). Again a routing algorithm is generated for the warehouse structure and using this algorithm warehouses with 3 cross aisles are compared with those with 2 cross aisles.

A review of what has been done mostly in the area of warehouse order picking has been presented by De Koster et al. (2007). The paper includes a revision of warehouse missions and functions, and an overview of order-picking systems. It includes detailed literature review on layout design, storage assignment, zoning, batching, routing methods, and order accumulation and sorting. *Figure 2.1* shows the classification for the literature presented in this chapter.



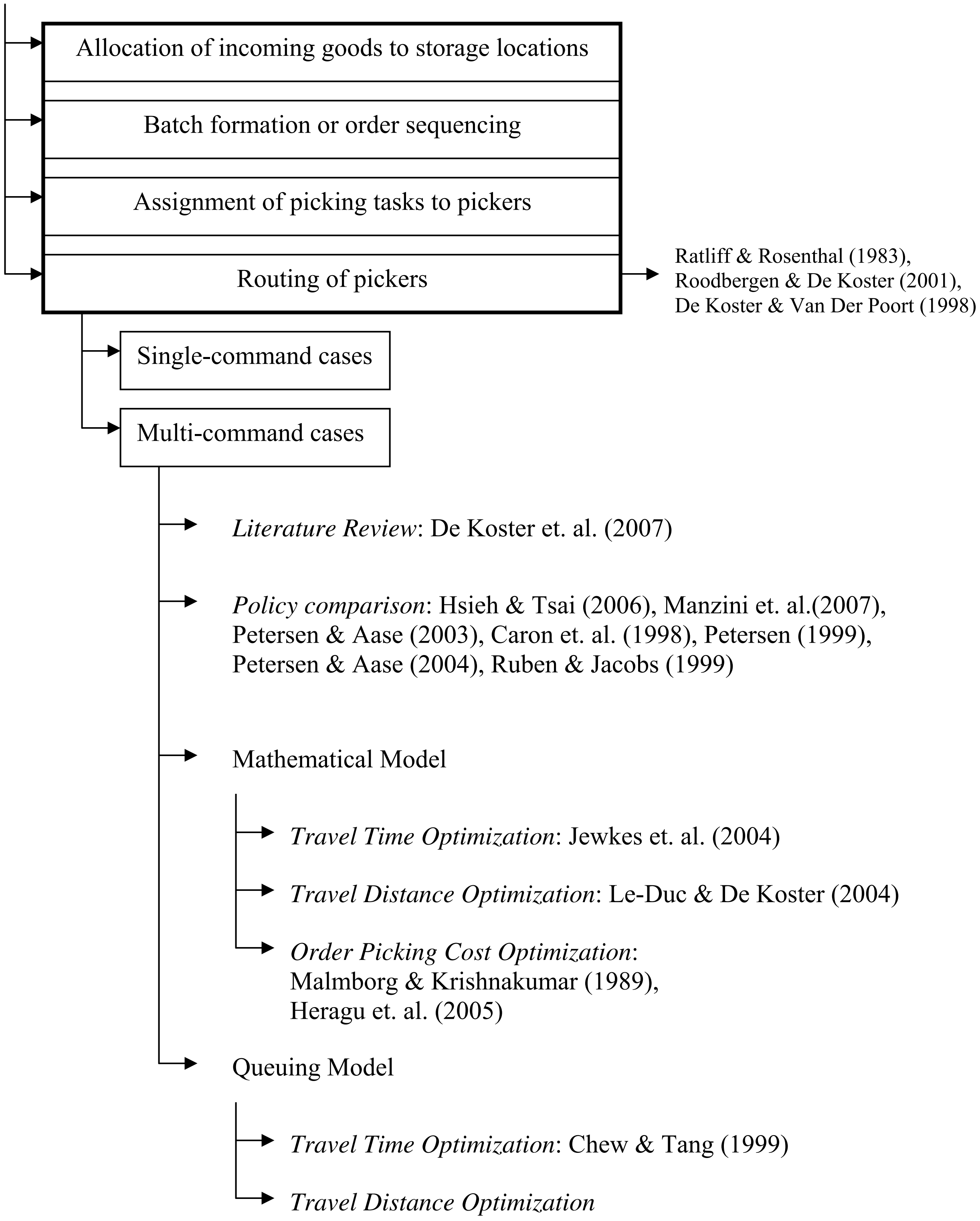


Figure 1.1 Literature classification

Chapter 3

PROBLEM DEFINITION

The problem considered for this study is a real world problem taking place in an automotive spare parts warehouse in Turkey and is brought to our attention by the company.

The problem environment, the spare parts warehouse, is a rectangular building located in the same region as the company's local manufacturing plant. The building has three main functional areas, namely the *item receiving area*, *storage area*, and *order preparation area*. There are other areas such as offices, resting area, etc. that are not included under the functional areas since they do not have a direct impact on the warehouse processes. *Figure A1*, in the Appendix – Section A, shows the block plan of the warehouse including all areas. Different processes are carried out in each functional area using different resources under specified organizational policies. *Table 3.1* shows a summary of the resources, processes, and organization of each functional area for which details are given in the following paragraphs.

An item arriving from either a local or international supplier is unloaded from the truck at the *incoming item dock* and is first placed in the item receiving area until the receiving process for the item is complete. This area is an empty space reserved for the receiving process at the front of the warehouse next to the incoming item dock. Here, items are stored on the warehouse floor. Different suppliers deliver their products in different packages and unit loads. Dependent on the quantity they supply to the warehouse, some bring the items in pallet loads having a number of one-item packages; some bring only a couple of

Table 3.1 Resources, processes, and organization of each functional area in warehouse

Functional Area	Resources	Processes	Details of Processes	Organization
Item Receiving Area	Hand Computer	Receiving	Opening	-
			Checking&Counting	
			Data input to computer system	
			(re)packaging	
	Hand Pallet Truck		Storage card formation	
			Transfer to temporary storage point	
Storage Area	Cartex Storage System (vertical carousel)	Storage	Placing item to storage slot	Random Storage Policy
	AA-AI Storage System (shelves)			
	B-Z Storage System (shelves and containers)		Making any changes in storage location if necessary	
	Forklift			
	Storage Card			
	Hand Truck	Order Picking	Routing of order pickers for order pick	Order Batching (parallel picking) & Return Routing Policy
	Pick Card			
	Hand Computer			
Order Preparation Area	Computer System	Order Card Preparation	Print order data for each customer	-
		Pick Card Preparation	Accept orders from customers, batch orders into zones, print cards for each zone	
		Order Preparation	Sorting	
			Boxing for each customer	
			Loading to trucks	

unit boxes containing 1-item packages, while some do not even package their products and deliver them in open carts. According to the delivery and storage form of the item, each item undergoes a combination of operations in the receiving area, namely opening, checking and counting, data entering (into the computer system), (re)packaging, storage card formation, and transfer to temporary storage point. For example, an item arriving in pallet loads with each pallet containing boxes with one unit of item inside will be opened, checked (for any damages) and counted (for quantity verification with supplier). Data of the item (item number, arrival date, quantity) will be entered into the computer system which will print a *storage card* (a card showing the item name and quantity, and the storage slot it should be placed at). The item pallets, together with the storage card, will be carried by hand pallet trucks to the *temporary storage point* where they will wait to be picked up by forklifts for the storage process.

The temporary storage point is in the storage area, which is the largest functional area of the warehouse. It has a storage system consisting of three different sub-systems. The first is referred to as the *Cartex System* which is a vertical carousel. Vertical carousels are automated storage systems where the shelves rotate up and down to deliver the required item to an ergonomically positioned pick window. The same system can be with shelves rotating left and right, which is called the horizontal carousel. The Cartex System stores small sized (as small as nuts and screws, for example) items in small quantities. Picking in this system is done only by entering the item number into the system's computer and waiting for the item to arrive at the pick window. No traveling is done as in traditional order picking processes (i.e. parts-to-picker system).

The second storage system is the *AA-AI storage system* which is a three-floor high-rack storage system. Different from the traditional high-rack storage systems, each floor of the shelf rack contains its own aisles meaning that the order picker can walk along each aisle on the upper floors with a hand cart and pick items. An elevator is present on the storage system to allow the hand carts to move up and down between the floors, while the order picker uses the stairs of the system. The AA-AI storage system is divided into 9 zones (AA, AB, AC, AD, AE, AF, AG, AH, AJ) with three on each floor. The first floor (i.e. bottom floor) is AA, AB, AC; the second floor (i.e. middle floor) is AD, AE, AF; and the third floor (i.e. top floor) is AG, AH, AJ. All storage slots in this system are relatively small in size when compared to the slots of the B-Z storage system and can store only small, or long but narrow parts. Specifically, there are 11 different storage slot types differing in width, length, and height dimensions. Items are stored directly on the storage slot without any *storage auxiliaries* (containers, pallets, and etc. used for storing items on shelf slots). Note that a *shelf slot* and a *storage slot* differ from each other. A shelf slot is the hollow space on the shelf for storing items whereas a storage slot is the precise space the items are stored in the shelf slot. To clear up the definitions, storage auxiliary is placed into shelf slots to form storage slots and so for cases where there is no storage auxiliary than the storage slot is the shelf slot itself. Order picking is done manually with hand carts in each zone simultaneously (i.e. parallel batch picking in a picker-to-parts system with pick-and-sort policy).

The largest storage system is the third one which is referred to as the *B-Z storage system*. It is a traditional high-rack shelf system where the shelf lines are placed parallel to each other with pick aisles, in between. This system has 15 parallel *pick aisles* and three *cross aisles* (aisles that are perpendicular to the pick aisles used for traveling between them; in the B-Z storage system there is one at the front, one at the back, and one in the middle of the pick aisles). Each pick aisle has shelves on both of its sides except for the first and last aisles which are only one-sided. There are a total of 53 shelf lines with each aisle, except the two end aisles, defining two shelf lines – one on each side of the aisle. A simple layout representation of the shelf lines and aisles is presented in *Figure A2,* in the Appendix – Section A. However, the shelf types on each line are not identical, implying that all the shelf slots of this system are not necessarily of the same size. The B-Z storage system has two types of shelves differing in their physical properties. One is the type where the shelf lines are fixed to the warehouse floor and cannot be moved to a different place. This means that these shelf lines have a specific position and height. Nearly 70% of the B-Z storage system is of this type of shelves where there a total of 78 shelf slot types differing in width, length, and height dimensions, and storage auxiliaries used. The rest of the B-Z storage system consists of portable storage units.

These are one unit storage slots that can be placed on top of each other, if desired, to form high level storage shelves and are not fixed to the warehouse floor. This implies that the location and height of storage systems formed with these portable storage units can be changed. However, it is evident that such changes are highly unlikely except perhaps in the long run. Items to be stored in the B-Z storage system are placed in shelf slots either on pallets (mostly for the items arriving from the supplier in large quantities with pallets), in containers owned by the warehouse, or directly on the shelf storage slot. Currently the warehouse has five different containers in various quantities. *Table 3.2* shows the dimensions and current quantities of the storage auxiliaries in the warehouse.

Table 3.2 Current storage auxiliaries used in the warehouse

Storage auxiliary	**Width (cm)**	**Length (cm)**	**Height (cm)**	**Quantity in warehouse**
Small Container	80	124	121	3202
Large Container	120	160	145	1026
Wooden Case	-	-	-	830
Metal Case	-	-	-	204
Pallet	-	-	-	1043

Other than the normal shelf slots where items are placed directly in the slot or with storage auxiliaries, there are shelf slots with metal dividers. These slots store flat and thin items and each item is placed in one division of the shelf slot. Given all these details related to the shelf slot dimensions and storage auxiliary used, the types of storage slots within the warehouse for storage purposes can be defined by the storage auxiliary type used due to the fact that they determine the actual space for storage. Currently, there are 38 different types of storage slots in the B-Z storage system. Independent from which shelf slot storage auxiliary is placed, it is counted as one type of storage slot. However, each shelf slot with a different dimension (whether it is a permanent type of shelf slot or a portable type) is considered to be a different type of storage slot. All storage slot types in this storage system are presented in *Table 3.3*.

Table 3.3 Storage slot types in the B-Z Shelf System

Storage Slot Code	Width (cm)	Length (cm)	Height (cm)	Storage Auxiliary	# storage slots of this type in B-Z region	% of this storage slot type in B-Z system
SC				Small Container	3028	35,23
LC				Large Container	999	11,62
PLT				Pallet	1041	12,11
PS1				Portable Shelf 1	64	0,74
MC				Metal Case	114	1,33
PS2				Portable Shelf 2	675	7,85
PS3				Portable Shelf 3	1224	14,24
WC				Wooden Case	831	9,67
DS1	160	290	275	Divided Shelf	2	2,44
DS2	160	290	148	Divided Shelf	9	
DC3	160	290	125	Divided Shelf	31	
DC4	160	290	119	Divided Shelf	30	
DC5	160	290	73	Divided Shelf	138	
SS1	168	200	19	Shelf Slot	20	4,75
SS2	160	290	306	Shelf Slot	3	
SS3	160	270	306	Shelf Slot	1	
SS4	160	250	104	Shelf Slot	2	
SS5	160	250	22	Shelf Slot	30	
SS6	160	251	48	Shelf Slot	18	
SS7	160	270	30	Shelf Slot	4	
SS8	160	270	25	Shelf Slot	4	
SS9	200	169	104	Shelf Slot	18	
SS10	153	169	104	Shelf Slot	36	
SS11	153	169	48	Shelf Slot	11	
SS12	160	250	22	Shelf Slot	30	
SS13	200	169	103	Shelf Slot	134	
SS14	200	169	75	Shelf Slot	68	
SS15	200	169	47	Shelf Slot	9	
SS16	160	285	169	Shelf Slot	1	
SS17	160	285	157	Shelf Slot	3	
SS18	160	218	197	Shelf Slot	1	
SS19	160	218	186	Shelf Slot	2	
SS20	160	194	198	Shelf Slot	2	
SS21	160	218	169	Shelf Slot	1	
SS22	160	194	186	Shelf Slot	4	
SS23	160	180	198	Shelf Slot	1	
SS24	160	218	157	Shelf Slot	3	
SS25	160	180	186	Shelf Slot	2	

The B-Z storage system accommodates a high majority of the items to be stored in the warehouse and so the temporary storage point where the items are placed after all operations of the receiving process is determined to be within this system's area. A man-on-board forklift picks the item to be stored from this point; takes it to the storage location specified on the storage card (the computer system gives the last location of the item); if the slot is empty, places it there; if not, places it to the nearest empty slot and inputs the new location data into the computer system via his hand-held computer. Then, the picker returns to the temporary storage point to pick the next item to be stored. Due to the single stop the man-on-board forklift makes in a typical storage tour, it can be referred to as a *single stop storage tour system* where storage is done in bulks. *Figure A3*, in the Appendix – Section A, shows an example of a single-stop storage tour in the B-Z shelf system. The current storage assignment policy for this warehouse is a randomized storage policy, based on the "independent" location decision the operator can take when the storage slot determined by the computer system is not empty. There is no restriction on assigning a storage slot to an item, resulting in a randomized storage.

Order picking, in this area, is done by traditional order picking tours where the order picker travels along and between the picking aisles with a 4-wheeled hand truck for low shelves and a forklift for high shelves to collect the ordered items (i.e. picker-to-parts system). This storage system is divided into 11 zones (B, C, D, F, G, H, K, M, S, T, and Z), so batch order picking is done with a return routing policy between zones simultaneously (i.e. parallel batch picking with pick-and-sort policy). Each order pick tour begins at the order preparation area, stops at multiple item locations, and finishes again at the order preparation area. So these routing tours are called *multi-stop picking tours* where picking is done in units, rather than in bulk. *Figure A4*, in the Appendix – Section A, shows an example of a multi-stop picking tour completed in one zone of the B-Z shelf system.

The last functional area in the warehouse is the order preparation area where order cards, pick cards, and customer boxes are prepared. Orders from customers arrive at the warehouse via the online computer system daily. Management has divided orders into groups according to the importance of the order and the customer it comes from. Items that must be quickly sent to a customer are grouped in the "emergency" category and are picked up and sent within a day independent from the customer's priority ranking. Any order arriving from a customer with high priority is also shipped out within the same day or the day that follows. Every afternoon the card preparation team prints order cards having the customer name and location, ordered items' names and quantities and the pick cards for each zone within the warehouse including item name, pick quantity, and storage slot number data. The items on the pick card are sorted according to the route they will be picked up at and so the order picker only follows the directions on the card to complete the picking tour in his own zone. Once all items picked from each zone have been delivered to the order picking area, the order preparation teams box each customer's order separately (according to the order cards prepared previously) and load them on to the trucks waiting outside the *outgoing item dock* of the warehouse. The same order picking and order preparation procedures are carried out for orders arriving from lower priority customers twice a week (a week consists of five working days).

The problem the company faces in this warehouse is the randomness in the item storage processes. This process is done on a traditional and know-how basis, without any standard rules and regulations. This causes a high dependency on the experienced workers of the system, and no control over them by any means. However, if a storage assignment policy is set specifically (i.e. something other than the random storage policy) and the items are pre-assigned to their storage slots accordingly, then the dependency for workers would be minimized and the warehouse storage process transactions could even be controlled by a well designed computer system.

In the presence of the problem environment and the problem itself, our aim is to determine the storage assignment policy for this warehouse, given its processes, resources, and organization, and locate the items to the storage slots accordingly. While doing so, it is assumed that any past year data collected from the warehouse is enough to see the problem environment's general characteristic and behavior.

Chapter 4

SIMULATION APPROACH

The problem environment considered has many parameters each related to a different organizational issue of the warehouse. That is, conditions change when a different policy is applied for storage, routing, order batching, picking, and etc. Therefore, mostly, researchers dealt with the efficiencies of each policy under a specified warehouse environment rather than assuming certain policies and optimizing their implementation in the system. To see the performance of each policy, they design simulation models and compare the results of different (in terms of organization policies) runs with the system's predetermined performance measure.

Simulation models are also useful for viewing the system as a whole while determining the importance of each activity on the system performance. So, they can be used as a tool for determining an efficient solution to a problem among different solution methodologies implemented to the model. In the case of warehouses, this can be, for example, trying different item-storage slot assignments under specific organizational policies. As it is difficult to obtain an optimal solution via an analytical approach, simulation models present a manageable way of evaluating alternative solutions. The simulation approach for this study aims to obtain best solution among reasonable alternatives.

Two different environments, motivated by and approximating the real problem environment, are considered and results for both models are derived. The first simulation environment is the low-level warehouse system where the number of items and storage slots are relatively small. The second environment only differs from the first in the storage system configuration as it is a multi-level system. Accordingly, it stores a larger number of item types in a larger number of storage slots.

4.1 The Single-Level Simulation Environment

The simulation environment is different from the real problem environment since it is a "created" environment by considering many assumptions different from the original problem to simplify the problem while capturing its key characteristics. Any detail of the real problem environment that is not directly related to the problem in question (i.e. the storage assignment problem) is excluded from the simulation environment. Therefore, the simulation model will only include processes related to the storage assignment policy determined, namely the item storage and order picking processes. Overall, this simulation environment can be taken as a "simplified version of the original problem environment".

The warehouse, in consideration, is an automobile spare parts warehouse with a rectangular building where incoming items are spare parts supplied by a number of suppliers. The one-store shelves placed parallel to each other on the warehouse floor form parallel aisles. Within this layout structure there are 10 shelf lines and 6 aisles, with two of them being the cross aisles at the ends of the pick aisles. *Figure A5*, in the Appendix – Section A, shows the picking area floor plan. Each shelf line has 10 storage slots, so there are a total of 100 storage slots for the items to be stored. The storage slots are uniform in size. However, the number of units of an item type a storage slot can hold depends on the physical properties of the item. Therefore, depending on the annual inventory level of the item and the storage slot capacity for it, each item requires a different number of storage slots.

Items arriving from suppliers are stored in the warehouse and are depleted in time as dependent on their demands. There are 40 different types of items in consideration. These items, arriving in batches, are carried to storage slots by forklifts. The storage slot(s) for each item is predetermined under a dedicated storage assignment policy. The storage tours are single-stop tours beginning and ending at the incoming item dock.

Outgoing items are collected in smaller quantities and so can be handled with a four-wheeled hand truck. Item demands arrive at the warehouse one-by-one as single demand quantities for each item. That is, each item requested by the service centers arrives to the warehouse online database independent from the others. This implies that these item requests cannot be referred to as *orders* since an order was defined as “a combination of item types in various quantities requested by a customer” previously. Rather than requesting a *combination of items types in various quantities*, each item is demanded independently and the pick list (item number, demand quantity, and storage slot number(s)) is driven from these requests. Therefore, these single item orders will be called *item requests* instead of “orders”. Pick lists are formed at the end of each day and items on these pick lists are picked by a multi-stop pick tour beginning and ending at the outgoing item dock. The routing policy for a pick tour can be called “next closest location routing policy” where the order picker moves to the next closest storage slot of the items on the pick list from his current position.

Under the projection of what has been said up to now for the simulation environment, *Table 4.1* summarizes the simulation warehouse according to its entities, resources, processes, and organization.

Table 4.1 The single-level simulation environment summary according to its entities, resources, processes, and organization

Entities	Resources	Processes	Organization
Incoming items (40 item types)	Single-Level Storage System	Storage (single-stop tours)	Dedicated Storage
Outgoing items	Forklifts	Order picking (multi-stop tours)	Picker-to-parts order picking with no zoning
	Four-wheeled hand trucks		Next closest point routing policy

4.2 The Single-Level Simulation Model

The aim of the simulation model is to determine an efficient configuration of item-slot assignment so as to minimize the total material handling cost, based on one month's data. Various storage assignment configurations are implemented into the simulation model for each run without deviating from the assignment policy. At the end of each run the system's efficiency according to the implemented assignment is determined. The performance measure for the system will be the *total distance traveled* in 30-days.

The simulation model requires various input data. The most important two are the *incoming item data* and the *outgoing item data*. The incoming item data (i.e. data related to items arriving from the suppliers to the warehouse) is presented as an Excel file (ArrivalData.xls) with item number (defines the item) and arrival quantity (number of units of item arriving) columns. This data is generated randomly for one month. The outgoing item data (i.e. data related to items demanded from customers), randomly generated, is also presented as an Excel file (OrderData.xls) with item number and demand quantity (number of units of item demanded) columns. These two data files generate the attributes of the entities of the simulation system.

For storage transactions, the storage slot assignment for each item type must also be input to the model. The storage slots assigned to each item type are listed in an Excel file (StorageAssgData.xls – "Items" worksheet) with one column as the item number and the next column the number of units of item in storage slot, and the last column as the storage slot number, where each item type has A_i number of rows. A_i indicates the number of storage slots item *i* requires.

To calculate the single-stop storage tour distance and the multi-stop picking tour distance, a distance matrix must be input to the simulation model. This is a symmetric (100 x 100) matrix, namely *D*, with $d_h(i,j)$ indicating the rectilinear distance between storage slot *i* and storage slot *j* along the warehouse floor (i.e. horizontal distance). The formulation for $d_h(i,j)$ is as follows

$$d_h(i,j) = \begin{cases} \alpha_{kl} + \min\left[r(i,10k) + r(j,10l); r(i,(k-1)10+1) + r(j,(l-1)10+1)\right] & (a) \\ r(i,j) & (b) \end{cases}$$

where

α_{kl} : distance along the horizontal axis between shelf lines *k* and *l*

$r(i,10k)$: rectilinear distance between storage slot *i* and storage slot *10k* (i.e. the last slot on the shelf line when slots are numbered in increasing order from left to right) (note that each shelf line has 10 storage slots)

$r(i,(k-1)10+1)$: rectilinear distance between storage slot i and storage slot (k-1)$10+1$ (i.e. the first slot on the shelf line when slots are numbered in increasing order from left to right)

(a) : if shelf lines k and l are not on the same aisle

(b) : if shelf lines k and l are on the same aisle

The minimum operator within the distance function

$$\min\left[r(i,10k)+r(j,10l);\, r(i,(k-1)10+1)+r(j,(l-1)10+1)\right]$$

determines whether it is closer to travel from storage slot i to storage slot j from the right end of the shelf line or the left end.

It is known that every system takes in input data, processes it, and gives an output. Our simulation model does exactly the same; it takes the input data stated previously through a series of processes to derive an output. The model generates three Excel output files; the storage tours file (StorageAssgData.xls – "Storage Transactions" worksheet), the pick tours file (PickList.xls), and the inventory level file (StorageAssgData.xls – "Items" worksheet). The item storage transactions are given in the file as item number in one column, quantity stored in storage slot in the next with the distance traveled for each transaction as a separate column. The pick lists for each batch of item demand on each day are listed with item number, demand quantity, and storage slot number the item will be picked from as three separate columns, together with the total distance traveled for each pick tour in the fourth column of the file. As well as seeing the travel distance of each pick tour and storage tour separately, we can also calculate the total distance traveled for picking transactions and storage transactions in one month. Our main system performance measure, as stated previously, is the total distance traveled in a 30-days period, and this can also be calculated from the output data of the simulation model. The last output file is the inventory transactions file which can also be stated as an input file as well because this is updated throughout the simulation run as items are added and depleted from the storage slots. This file has three columns of data, namely item number, storage slot number, and quantity in slot. The "quantity is slot" column is updated within the processes of the model and the last version of the file at the end of the simulation run shows the final state of the warehouse inventory. It is an input file in the sense that the model checks if there are enough items in a certain slot during the picking transactions.

Now that the input and output data are stated clearly, it is important to define what exactly happens in between the input and output of data. Recall that only the two most important processes among the warehouse processes, namely the item storage and item picking, are included in the model. These two processes run simultaneously in a warehouse and so do they in the simulation model. Therefore, the simulation model can be assumed to be a combination of two sub models, the storage and the picking sub models.

The storage sub model begins with creating entities representing the item arrivals from suppliers. Entities are created with constant rate of every 10 hours, assuming that generally each day (each work day is 24 hours) 2-3 item types arrive at the warehouse from suppliers (*Step 1s* in *Simulation Manual* in Appendix – Section B). Once an entity is created with the CREATE module, *item number* (indicating which item it is) and *arrival quantity* (number of units of item sent from supplier to the warehouse) values, taken from ArrivalData.xls are assigned to its "ITEMNO" and "ARRIVALQUANTITY" attributes, respectively (*Step 2s* in Simulation *Manual* in Appendix – Section B). These assignments are done by the two READ\WRITE modules placed after the create module. According to this arrangement, each entity created will be a different item with a different arrival quantity; however the interarrival times between each pair of items is constant and independent from the item type arriving. Also, the interpretation of the fact that the model does not create a new entity before the readily created one is disposed, is that the storage process of arriving item types will be done

one-by-one (i.e. no two item type lots will be processed together in one storage transaction). Recall that the item receiving process is not modeled and so it is assumed that an incoming item lot is immediately sent to storage area after arriving at the warehouse.

Assigning incoming items to previously determined item dedicated storage slots is done according to the slot's capacity and the emptiness at the current time of item arrival. In order to find a suitable storage slot for the newly arrived item lot among the pre-assigned slots of the item, the model runs through a series of steps in the VBA module where Visual Basic codes are run. From the StorageAssgData.xls – StorageSlots file the capacity of a slot dedicated to the item (recall that each slot is unique in size; however, the number of units of item a slot can hold depends on the item type and so the capacity of a slot dedicated to an item is actually the maximum number of items the slot can carry for that certain item type) and from StorageAssgData.xls – Items the number of items in that slot currently is retrieved. These values are assigned to SLOTCAP and SLOTQUAN variables, respectively (*Step 3s* in *Simulation Manual* in Appendix – Section B). Next, the capacity the slot in question has remaining is calculated by REMAINCAP = SLOTCAP - SLOTQUAN (*Step 4s* in *Simulation Manual* in Appendix – Section B). If the slot has enough space to store the newly arrived lot (i.e. REMAINCAP ≥ ARRIVALQUANTITY), then this lot is added to that slot (i.e. SLOTQUAN = SLOTQUAN + ARRIVALQUANTITY) and these storage transaction details (time of transaction, item number, arrival quantity, slot number item was placed to, number of units of item in slot, and the distance traveled from incoming item dock to storage slot) are written to the StorageAssgData.xls – StorageTransactions file. Also, the corresponding row in StorageAssgData.xls – Items is updated for the current number of units of item in the slot (*Step 5s-True* in *Simulation Manual* in Appendix – Section B). However, if the slot does not have enough space left to store the whole lot arrived (i.e. REMAINCAP ≤ ARRIVALQUANTITY and REMAINCAP ≠ 0), then the slot is filled up to its capacity limit (i.e. SLOTQUAN = SLOTCAP) and the rest of the lot of the incoming item is placed at the next suitable slot of the item. To continue searching for a suitable slot first what is left to be stored from the incoming lot must be determined. So, LEFTQUAN = ARRIVALQUANTITY – REMAINCAP. Then, the same procedure of determining the slot capacity, quantity in slot, and the remaining slot capacity values; and assigning what fits from the incoming item lot to the slot in consideration and moving to the next storage slot for the remaining item units is revised (*Step 5s-False* in Simulation Manual in Appendix – Section B). Notice that every time the model runs this same procedure, it does it with different SLOTCAP, SLOTQUAN, REMAINCAP, LEFTQUAN values (because each time a different storage slot is considered) and the loop is ended when there is no unit of incoming item left to store (i.e. LEFTQUAN = 0). All transactions and updates are written to the files StorageAssgData.xls – StorageTransactions and StorageAssgData.xls – Items files, respectively. The entity created to run through these modules is then disposed at the end with the DISPOSE module (*Step 6s* in *Simulation Manual* in Appendix – Section B).

The picking sub model also begins with a CREATE module where entities are created randomly with an average of 5 hours interarrival time between two entities (i.e. EXPO(5)). The entities created in this module represent the item demands arriving from a customer (*Step 1p* in *Simulation Manual* in Appendix – Section B). After an entity is created, it runs through the two READ\WRITE modules where the item number and demand quantity values are read from the OrderData.xls file and assigned to the ORITEMNO and ORDERQUANTITY attributes of the entity (*Step 2p* in *Simulation Manual* in Appendix – Section B). So, by this, the entity has gained an "identity" and is not just any order demand but an order demand of a specific item with a specific demand quantity. Similar to the approach in the storage sub model, every entity created will gain a different identity. This matches with the requirement that item type orders arrive one-by-one with their own demand quantities. It is assumed that one pick tour is done per day. To determine each day's pick tour the item demands arriving in the same day must be batched. To allow the model to do so, a signal is released every 24 hours to the system by the CREATE, SIGNAL modules and disposed right afterwards by the DISPOSE module (*Step 3p* in *Simulation Manual* in Appendix – Section B). Meanwhile, entities arriving one-by-one to the system are counted by an ASSIGN module where a variable COUNT1 is defined and the value of COUNT1 is increased by 1 each time an entity passes the module (i.e. COUNT1 = COUNT1+ 1) (*Step 4p* in *Simulation Manual* in Appendix – Section B). The HOLD module placed afterwards is to hold the entities until the signal is released to the system (*Step 5p* in *Simulation Manual* in Appendix – Section B). This allows the entities created in the first 24 hours to separate from the entities created in the next 24 hours. The interpretation for the warehouse is that the item demands of every 24 hours are picked separately. With the VBA module following the HOLD module, the order details (item number, order quantity, time item order had arrived to system, and time of pick tour) are written to the Picklist.xls – Picklist file with the order they had arrived to the warehouse (*Step 6p* in *Simulation Manual* in Appendix – Section B). Each entity entering the VBA module provides data for each row of the picklist forming in the file. As entities leave the VBA module and enter the following ASSIGN module the COUNT2 variable defined in the ASSIGN module is increased by 1 (i.e. COUNT2 = COUNT2 + 1) for each entity (*Step 7p* in *Simulation Manual* in Appendix – Section B). This is to count the number of items released after the HOLD module because the upcoming DECISION module divides the entity's path into two according to the values of COUNT1 and COUNT2. If the entity arriving is not the last entity of the HOLD module's queue (i.e. the item is not the last item on the pick list of the day), then it goes directly to the DISPOSE module to leave the system (*Step 8p* in *Simulation Manual* in Appendix – Section B). However, when the end of the pick list is reached (i.e. COUNT1 = COUNT2), the entity goes into the VBA module on the "true path" of the DECISION module where the picking transactions are executed. First, an extra line on the pick list in Picklist.xls – Picklist is added indicating the end of the day's item pick orders (*Step 9p* in *Simulation Manual* in Appendix - – Section B). After initializing the current position of the order picker to point 0 (i.e. CURLOC = 0), then the two data columns in Picklist.xls - Picklist, namely 'item number' and 'order quantity', are assigned to the two arrays ARRITEM and ARRQUAN, respectively (*Step 10p* in *Simulation Manual* in Appendix –– Section B). The slots dedicated to each item type are also entered to an array (ARRSLOTNO) together with an array indicating the corresponding item number (ARRITEMNO) where the item and slot numbers are taken from the file StorageAssgData.xls – Items (*Step 11p* in *Simulation Manual* in Appendix – Section B). Finally, an array with elements as "distance between current location and slot *i*" named ARRDIST is constructed where slot *i*'s are the elements of the ARRSLOTNO array. The distance data is retrieved from the distance matrix in the file Picklist.xls - Distances (*Step 12p* in *Simulation Manual* in

Appendix – Section B). After all arrays are filled with the appropriate values, ARRDIST is sorted in ascending order of distance. While doing so, ARRSLOTNO and ARRITEMNO are also sorted with respect to ARRDIST (*Step 13p* in *Simulation Manual* in Appendix – Section B). The first elements of the ARRITEMNO, ARRSLOTNO, and ARRDIST arrays give the item in the closest slot to the current position of the order picker. Therefore, these values are written to the Picklist.xls – Pickroute file together with the demanded quantity of the item (i.e. the element in ARRQUAN corresponding to the required item number in ARRITEM) (*Step 14p* in *Simulation Manual* in Appendix – Section B). Notice here that the closest item storage slot to the current location may not have enough number of units of item demanded on the pick list. In this case the order picker picks all units in that slot and moves on to the next closest slot of that specific item to retrieve the remaining units demanded. Each time units are retrieved from a storage slot of the item, their corresponding data is written to the Picklist.xls-Pickroute file and the remaining demanded quantity is calculated as LEFTQUAN = ORDERQUAN-SLOTQUAN for the first pick, and LEFTQUAN = LEFTQUAN - SLOTQUAN for the next picks. This loop ends when the demand for the item is picked completely from the storage slots of the item (i.e. LEFTQUAN = 0), or when all storage slots are checked and no units are left for picking the remaining demand (i.e. LEFTQUAN > 0 but no other slot for picking). Then the row with the item's data is deleted from the Picklist.xls – Picklist file so that it will not be included in the picking process anymore, as it has already been picked (*Step 15p* in *Simulation Manual* in Appendix – Section B). The current location of the order picker is then updated (i.e. CURLOC = ARRSLOTNO[1]) and the loop returns to assigning the data to the ARRITEM, ARRQUAN, ARRITEMNO, ARRSLOT, and ARRDIST arrays. Notice that this time, rows with value 0 (zero) will not be added to the arrays as there is an IF statement in the code controlling for non-negativity before assigning. The loop will end when all rows are zero for the day's pick list (recall that the end of the day's pick list was indicated with a mark in the file) (*Step 16p* in *Simulation Manual* in Appendix – Section B).

4.3 The Multi-Level Simulation Environment

Assume that the number of item types increases in the single-level system described previously. Then, to be able to store all items within the warehouse, the number of storage slots must be increased accordingly. This can be done in either of the two ways:

1. Keep the system as a low-level storage system and so change the building of the warehouse where the floor plan can fit all shelves, without facing any sizing problems. This can be referred to as a "horizontal expansion".

2. Keep the warehouse building as it is, and increase the number of storage slots by turning the storage system into a multi-level system where shelves are put on top of each other to form levels. This can be called a "vertical expansion".

As it is not convenient to change the warehouse building, we prefer to apply the second approach and change the single-level system to a multi-level system.

The multi-level warehouse system is very similar to the single-level system in terms of entities, resources, storage system, and organization. These are given in *Table 4.2*. The only difference is that the storage system has four levels of shelves (multi-level) rather than only one level (single-level) on the given layout as shown previously on *Figure A5,* in the Appendix – Section A. Each level of shelves has 100 storage slots so there are a total of 400 storage slots in this new 3-dimensional warehousing system. Parallel to this, the warehouse is to store 140 different item types, each of which has a known A_i (i.e. the number of storage slots required by item i).

Table 4.2. The multi-level simulation environment summary according to its entities, resources, processes, and organization

Entities	Resources	Processes	Organization
Incoming items (140 item types)	Multi-Level Storage System	Storage (single-stop tours)	Dedicated Storage
Outgoing items	Forklifts	Order picking (multi-stop tours)	Picker-to-parts order picking with no zoning
	Four-wheeled hand trucks		Next closest point routing policy

As stated previously, this warehouse applies dedicated storage policy, order batching policy, no zoning (i.e. single order picker picking the items all over the warehouse) with “next closest point” routing policy. One main assumption is that items are not split to different levels of shelves during the storage assignment process. That is, all storage slots assigned to an item will be on the same shelf level on the multi-level storage system. Another assumption, a result of the “no zoning” policy, is that the single order-picking will pick from all levels of shelves without any restriction. In other words, all items on all shelf levels are on the same pick list and are picked one by one, so the order picker travels between levels, if necessary, during an order pick route.

4.4 The Multi-Level Simulation Model

The simulation model for the multi-level storage system is, too, very similar to the one of the single-level storage system. The model works in the exact same manner except for the performance measure it gives as an output. Given that the input data files are of the same format, but with different entries now which reflect suitable data for the multi-level environment, the processes carried out are the same but generating both “travel time” and "travel distance" as outputs, which will be written to a new output file PicklistHL.xls - Pickroute. The system performance measure, however, will be *total travel time* in a 30-days time period in this case as oppose to total travel distance in the single-level case. An important reasoning lies behind this change.

Recall that in the single-level system the man-on-board picking vehicle travels only along the warehouse floor (i.e. the horizontal direction) with constant speed of $v_h = 1\ m/s$. Therefore, travel time is proportional to travel distance (t = d\v). Minimizing time is directly proportional to minimizing distance since velocity is constant. So, travel distance can be used as a system performance when there is a low-level storage environment. However, when the system rises to a multi-level system, the man-on-board order picking vehicle travels along the warehouse floor (i.e. horizontal travel) and up to the level of the storage slot (i.e. vertical travel). Given that the vehicle travels the horizontal direction at "unit speed", $v_h = 1\ m/s$, it goes up or down twice as slow implying that $v_h = 1/2\ m/s$. So, the travel time between two storage slots will be determined according to both the vertical and horizontal travel times independently.

This implies that total travel time is not directly proportional to total travel distance since the velocities of the horizontal and vertical directions are not the same. The inference behind this is that traveling 1m horizontally is not equal to traveling 1m vertically in terms of "work" and so their performances must be degreed separately. Therefore, minimizing time and minimizing distance cannot be used interchangeably.

The travel time between two storage slots can be determined in two ways, basically according to the assumption considered for the order picking vehicle (i.e. forklift in this case) movements. That is, it can be assumed that

(i) the forklift can only move in one direction at a time (i.e. sequential moves in the horizontal and vertical directions), or

(ii) the forklift can move in both directions at the same time (i.e. simultaneous moves in the horizontal and vertical directions).

Depending on the assumption made for the forklift moves, the calculation of the travel time between two storage slots changes. If the first assumption is considered, then the travel time between two storage slots will be of rectilinear measure as given below

$$\begin{aligned} t(i,j) &= t_h(i,j) + t_v(l_i,l_j) \\ &= \frac{d_h(i,j)}{v_h} + \frac{d_v(l_i,l_j)}{v_v} \qquad \forall i,j \end{aligned}$$

where

$t(i, j)$: total travel time between storage slots i and j
$t_h(i,j)$: horizontal travel time between storage slots i and j
$t_v(l_i,l_j)$: vertical travel time between shelf levels of storage slots i and j
$d_h(i,j)$: horizontal travel distance between storage slots i and j
$d_v(l_i,l_j)$: vertical travel distance between shelf levels of storage slots i and j
v_h : horizontal travel speed (m\s)
v_v : vertical travel speed (m\s)

If the second assumption is taken into consideration, then the travel time between two storage slots is based on a Chebyshev measure as the following

$$
\begin{aligned}
t(i,j) &= \max\left(t_h(i,j), t_v(l_i,l_j)\right) \\
&= \max\left(\frac{d_h(i,j)}{v_h}, \frac{d_v(l_i,l_j)}{v_v}\right) \qquad \forall i,j
\end{aligned}
$$

where the notation is the same as above.

In both cases, the Visual Basic code for the high-level system will retrieve $d_h(i,j)$ from the distance matrix presented for the low-level system, namely D, located in Picklist.xls – Distances; and the $d_v(l_i,l_j)$, v_h, v_v values are defined as constants.

The multi-level simulation model is capable of calculating each case independently, and so the output printed to the PicklistHL.xls - Pickroute file will include item number, order quantity, storage slot number, rectilinear travel distance between the storage slot and previous storage slot, rectilinear travel time of the distance traveled, Chebyshev travel time of the distance traveled. This will allow a comparison to be done among the two assumptions taken.

Chapter 5

STORAGE ASSIGNMENT ALTERNATIVES AND THEIR RESULTS

Recall that there are two different simulation models presented previously and each one's results will be evaluated separately. It is not possible to compare the results of the two models since they serve the purpose of viewing each system on its own rather than comparing a single-level and a multi-level system. However, it can be said that the single-level model is an exact approximation of the multi-level model when forklifts are allowed to move in the horizontal and vertical directions simultaneously (i.e. Chebyshev measure) and a very close approximation when this movement is done sequentially (i.e. rectilinear measure). This is because the travel distance/time in the horizontal direction is much larger than that of the vertical direction and when Chebyshev measure is used the maximum value is always of the horizontal movement, decreasing the two-dimensional movement to single dimension. When rectilinear measure is used, the vertical distance/time is added but this contributes to a very small portion and so makes only a small change in the total value. Therefore, it can be said that the single-level is a close approximation of the multi-level in the rectilinear measure case.

Each simulation model is run with a minimum of six different storage slot assignment configurations, staying strict to the dedicated storage assignment policy, and the results of each system are compared separately.

The simulation models are run with fictitious data that resemble real world data to some extent. The values generated for the performance measures are much smaller than those of real warehouses; however the magnitude of results does not make too much difference in relative performance of different layouts. The fictitious data is suitable for comparing various storage assignments implemented. Overall, the comparisons of results are more important than absolute magnitudes.

5.1. Single-Level System Simulation Results

For the single-level system, ten different storage assignment configurations were implemented where total distance traveled for picking and storage transactions are generated separately. From these, average distance (in meters) per picking tour and average distance per storage tour values are derived. The same incoming item and outgoing item data files were input to each run, which consists of a 30-days replication.

5.1.1. Storage Assignment Configuration #1: Arbitrary Assignment

To start with, an arbitrary storage slot assignment with no certain assignment rule is generated. For simplicity, though, item types and storage slots were ordered in increasing order and matched together. Therefore, items with smaller index numbers are located at smaller indexed storage slots. Simulation runs results can be seen in *Table 5.1*, below. *Figure*

A6 in the Appendix – Section A shows the picture representation of the arbitrary configuration.

Table 5.1. System performance measures obtained from single-level simulation model for Configuration #1

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2316	3923	6239
Average Values (per tour)	**Distance (m\tour)**	80	62	-
Average Values (per day)	**Distance (m\day)**	80	131	-

5.1.2 Storage Assignment Configuration #2: Arbitrary Assignment

This is the second arbitrary storage assignment implemented to see the behavior of the system under a different assignment variation. Different from Configuration #1, here item types are not split to more than more shelf line. That is, every item type in assigned to the slots of a single shelf line. Results are shown below in *Table 5.2*. *Figure A7* in the Appendix – Section A shows the picture representation of the second arbitrary configuration.

Table 5.2. System performance measures obtained from single-level simulation model for Configuration #2

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2564	4144	6708
Average Values (per tour)	**Distance (m\tour)**	92	61	-
Average Values (per day)	**Distance (m\day)**	92	138	-

5.1.3 Storage Assignment Configuration #3: Arbitrary Assignment

The third configuration is again an arbitrary assignment with the following performance measure values as shown in *Table 5.3*. This assignment has only one item split into two shelf lines, better than Configuration #1 but worse than Configuration #2 when compared in this. The most frequently demanded items, namely item 6, 5, 4, 7, and 8 are located closer to the picking dock when compared to their locations in configuration #2 where they are spread throughout the warehouse. *Figure A8* in the Appendix – Section A shows the picture representation of the third arbitrary configuration.

Table 5.3. System performance measures obtained from single-level simulation model for Configuration #3

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2452	4023	6475
Average Values (per tour)	**Distance (m\tour)**	88	60	-
Average Values (per day)	**Distance (m\day)**	88	134	-

5.1.4. Storage Assignment Configuration #4: Decreasing A_i

For this configuration, items are sorted in decreasing A_i while storage slots are sorted in increasing index order. Item-slot matching is done with these ordered lists. That is, aisles closer to the picking dock will store items requiring more storage slots. This implies that these aisles will accommodate less variety of items and as one goes further away from the picking dock the number of different item types the aisles hold will increase. This is an A_i parameter dependent configuration which is not directly related to the demand of the items. For example, an item requiring 5 slots may be demanded less frequently than an item requiring 2 slots. The first item in question may be large in volume and so requires 5 slots for storage whereas the more frequently demanded second item may be small in size and so the item lot fits into 2 slots easily. However, one cannot say that there is totally no relationship between an item's demand frequency and quantity and it's A_i. As more item units are demanded, more of that item must be stored and so more storage slots are required for the item. Simply, it can be said that when comparing two different demand statuses of the same item type the state with larger A_i is demanded more whereas this cannot be said when two different item types are compared (i.e. the item type with larger A_i may not be the one with higher demand frequency and quantity).

This assignment configuration was implemented in the simulation model to deviate from the random configuration and to see if it has any effect on improving the system efficiency. Results do show that since it is not directly demand related it does not decrease system performance measure as much as a "demand related" configurations, presented in the next sections. *Table 5.4*, below, gives the summary results of the configuration. *Figure A9* in the Appendix – Section A shows the picture representation of the decreasing A_i configuration.

Table 5.4. System performance measures obtained from single-level simulation model for Configuration #4

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2636	3772	6408
Average Values (per tour)	**Distance (m\tour)**	94	58	-
Average Values (per day)	**Distance (m\day)**	94	126	-

5.1.5. Storage Assignment Configuration #5: Picking Oriented (S-shape)

It is very logical to place items that are more frequently demanded to slots closer to the picking dock in order to decrease the distance traveled during a picking tour. By this, the order picker will mostly be traveling between aisles and slots near the picking dock and less frequently going to further slots to pick rarely demanded items.

Once the item order frequencies are derived from the 30-day item order data, items are sorted in decreasing frequency order. Then, they are assigned to storage slots in the order of decreasing demand frequency. For this configuration, the most frequently demanded item is assigned to the first shelf line starting from the slots closest to the picking dock. The second item on the frequency order list is assigned to the next set of slots following the first item. Once this shelf line is full, the slot assignment of the next item continues starting from the other end of the next shelf line. Therefore, if the operator were to place all 40 items sequentially with multi-stop tour, he would be starting from slot #1 and walking in an S-shape between the aisles until he reaches the last point which is slot #91. For simplicity, this configuration can be referred to as the *S-shape storage slot assignment*. An important inference about the S-shape configuration is that it only considers assigning frequently demanded items to close shelf lines disregarding its position within the line. That is, odd numbered shelf lines will have decreasing item frequency as one goes further away from the picking dock in the aisle, while even numbered shelf lines will have increasing item frequency along the same direction. Simulation results are shown in *Table 5.5*, below. *Figure A10* in the Appendix – Section A shows the picture representation of the S-shape picking oriented configuration.

Table 5.5. System performance measures obtained from single-level simulation model for Configuration #5

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2382	3935	6317
Average Values (per tour)	**Distance (m\tour)**	85	61	-
Average Values (per day)	**Distance (m\day)**	85	131	-

5.1.6. Storage Assignment Configuration #6: Picking Oriented (I-shape)

This is the second picking oriented configuration implemented to the low-level system. Here, the items sorted in decreasing demand frequency order are assigned one by one from the list to shelf lines starting at the same end of each line every time a new line is entered. In other words, items are placed to the closest end of the shelf line first and moved away from the picking dock to fill the whole line with items. Therefore, every shelf line in the system will have a decreasing trend in demand frequency as one goes along an aisle starting from the end close to the picking dock. This also implies that frequently demand items are placed to both near shelf lines and near positions within the lines. This configuration can be called the *I-shape storage slot assignment* since the assignment of items is similar to an "I". *Table 5.6*, below, gives the summarized results of the run. *Figure A11* in the Appendix – Section A shows the picture representation of the I-shape picking oriented configuration.

Table 5.6. System performance measures obtained from single-level simulation model for Configuration #6

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2284	3947	6231
Average Values (per tour)	**Distance (m\tour)**	82	64	-
Average Values (per day)	**Distance (m\day)**	82	132	-

5.1.7. Storage Assignment Configuration #7: Picking Oriented (H-shape)

The last picking oriented storage assignment configuration implemented to this low-level system is where items sorted in decreasing demand frequency order are assigned one by one to storage slots on the front half (i.e. the half closer to the picking dock) of the shelf lines. More specifically, the items on the demand frequency order list are assigned to shelf lines by starting at the same end of each line entered and the slots up to the mid point of the line are filled. Once the front half of the system is assigned, the relatively less demanded items are assigned to storage slots at the rear half of the shelf lines. This implies that the order picker will mostly be traveling between slots in the front half region of the storage system during order picking. This will be useful to see if it is efficient to travel only half way along each aisle. *Table 5.7* shows the results. The similarity of the assignment of items to the letter "H" gives the name of the configuration, *H-shape storage slot assignment. Figure A12* in the Appendix – Section A shows the picture representation of the H-shape picking oriented configuration.

Table 5.7. System performance measures obtained from single-level simulation model for Configuration #7

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2983	3848	6831
Average Values (per tour)	**Distance (m\tour)**	107	62	-
Average Values (per day)	**Distance (m\day)**	107	128	-

5.1.8. Storage Assignment Configuration #8: Supply Oriented (S-shape)

Remember that the Configurations #5, #6, and #7 are picking oriented as they aim to decrease the total distance traveled during picking. However, a warehouse also has its storage transactions where a considerable amount of distance is traveled. Also, the storage and picking docks in the warehouse are in opposite corners which derives the fact that a picking oriented assignment will result in high storage tours for items frequently demanded (i.e. these items will be located close to the picking dock but far from the storage dock). For this reason, a storage oriented storage assignment was generated by calculating the incoming frequencies of each item and then sorting in decreasing order. Items with high incoming frequency are tended to be assigned to storage slots closer to the storage dock. The first of these configurations is where the assignment procedure is done in an S-shape manner very similar to that of Configuration #5. This time, assignment will start from slot #91 with the most frequently arriving item and will continue throughout the whole warehouse in an S-shape. According to this, even numbered shelf lines will have decreasing incoming frequency when traveled from the front slot to rear slot, whereas odd numbered shelf lines will have an increase in item income frequency in the same direction. See *Table 5.8*, below, for performance measure values. *Figure A13* in the Appendix – Section A shows the picture representation of the S-shape storage oriented configuration.

Table 5.8. System performance measures obtained from single-level simulation model for Configuration #8

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2564	3476	6040
Average Values (per tour)	**Distance (m\tour)**	92	52	-
Average Values (per day)	**Distance (m\day)**	92	116	-

5.1.9. Storage Assignment Configuration #9: Supply Oriented (I-shape)

This is the second of the storage oriented assignments where items most frequently demanded are assigned to slots closer to the storage dock very similar to the I-shape manner of Configuration #6. *Table 5.9* gives the summary of the results of the run. *Figure A14* in the Appendix – Section A shows the picture representation of the I-shape storage oriented configuration.

Table 5.9. System performance measures obtained from single-level simulation model for Configuration #9

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2516	3504	6020
Average Values (per tour)	**Distance (m\tour)**	90	52	-
Average Values (per day)	**Distance (m\day)**	90	117	-

5.1.10. Storage Assignment Configuration #10: Supply Oriented (H-shape)

The final storage oriented assignment is the H-shape assignment. Here, frequently arriving items to the warehouse from suppliers are assigned to slots on the front half of the storage system, closer to the storage dock. The rear half of the system will have items arriving less frequently to the warehouse. *Tables 5.10*, below, gives the summarized results. *Figure A15* in the Appendix – Section A shows the picture representation of the H-shape storage oriented configuration.

Table 5.10. System performance measures obtained from single-level simulation model for Configuration #10

		Picking Tour	Storage Tour	Total Travel
Total Values (in 30 days)	**Distance (m)**	2510	3548	6058
Average Values (per tour)	**Distance (m\tour)**	90	53	-
Average Values (per day)	**Distance (m\day)**	90	118	-

5.1.11. Comments on the Single-Level System Storage Assignment Configurations

The previously described 10 configurations are compared with their total distance traveled (i.e. total picking distance + total storage distance) because it represents the total work done in the warehouse. Different assignments are compared with each other in different comparison groups, shown in *Table 5.11*.

The following comments were derived from the results:

1. The performance of an arbitrary assignment may sometimes be better than a ruled assignment because the random assignment implemented may turn out to be a good one. For example, here, Configuration #1 is better than most assignments. However, it does not imply that storage assignment can be done arbitrarily because not all randomly generated storage assignments will show good performance.

Table 5.11. Storage Assignment Configurations' Performance Summary (Single-Level System)

Comparison Type	Performance Measure	Configuration Order	Performance Measure Value
Arbitrary Assignments	Total Distance Traveled (m)	#1, #3, #2	6239 < 6475 < 6708
Picking Oriented Assignments	Total Distance Traveled (m)	#6, #5, #7	6231 < 6317 < 6831
Storage Oriented Assignments	Total Distance Traveled (m)	#9, #8, #10	6020 < 6040< 6058
Picking vs. Storage	Total Distance Traveled (m)	#9, #8, #10, #6, #5, #7	6020 < 6040< 6058 < 6231 < 6317 < 6831
All Assignments	Total Distance Traveled (m)	#9, #8, #10, #6, #1, #5, #4, #3, #2, #7	6020 < 6040< 6058 < 6231 < 6239 < 6317 < 6408 < 6475 <6708 < 6831

2. When the three picking oriented configurations are compared according to the slot assignment priorities they have, we see that the S-shape assignment assigns items according to a "right/left" strategy. That is, items are first located to slots on the right half of the warehouse, and then to the left half. Conversely, the H-shape assignment follows a front/rear strategy where slots on the front half of the warehouse are filled first. I-shape assignment is a middle strategy of these two extremes. *Table 5.12* summarizes this.

Table 5.12. Key Properties of the Storage Assignment Rule for the Picking Oriented Configurations

Configuration	Assignment Priority	
	Right / Left	Front / Rear
# 5	✓	✗
# 6	✓	✓
# 7	✗	✓

Take into consideration #5 and #7; placing items to close shelf lines is more efficient, even if the order picker has to traverse the whole aisle (i.e S-shape assignment - #5). Having more frequently demanded items at the closer half of the aisle results in the order picker visiting more aisles in a single pick tour which is not efficient (i.e. H-shape assignment - #7). Configuration #6, however, assigns the item according to both the right/left and front/rear strategies and so results in the lowest total travel distance among the three.

3. Similarly, when the storage oriented configurations are compared, the I-shape configuration gives the best performance.

4. When the picking and storage oriented assignment are compared it may seem reasonable to implement a storage oriented assignment to the warehouse however, this is not the case in practice. In all configurations, the total distance traveled for storage is larger than the total distance traveled for picking. This is mainly because each storage transaction is done with a single-stop tour where the operator travels to and from the slot during each one. This duplicates the distance traveled. Picking, on the other hand, is done with multi-stop tours where transactions are batched and so the unnecessary to-from travel between each slot and picking dock is eliminated. Therefore, the total picking distance in smaller when compared to the total storage distance. As a result of this, when a storage oriented assignment is implemented, a large decrease in the total storage distance is achieved, and so decreasing the total travel distance. This, however, does not imply that storage oriented assignment must be implemented to a warehouse because 65% of the transactions in this warehouse are picking transactions and need a considerable amount of effort. This percentage was calculated by adding all storage and picking transactions done in a month and deriving the ratio of picking transactions to this total value where a storage transaction is a single item storage tour and a picking transaction is a single item pick process (it cannot be defined as a “tour” since a pick tour includes a number of item picks; therefore, a pick tour consists of a number of pick transactions). This percentage is relevant to what has been presented in the literature. Ruben & Jacobs (1999) states that 65% of operating expenses are consumed by order picking. Petersen & Aase (2003) gives this percentage as a range of 50-75%. Also, efficient picking results in high customer satisfaction since fast shipping is directly related with fast and easy picking. In the presence of this information, warehouse managers tend to implement a picking oriented assignment.

5.2. Multi-Level System Simulation Results

Six different storage slot assignment configurations were generated for the multi-level system and three different performance measures, namely total distance, total time with rectilinear measure, and total time with chebyshev measure are evaluated for each one. The same incoming item and outgoing item data files were input to each run, which consisted of a 30-days replication.

5.2.1. Storage Assignment Configuration #1: Arbitrary Assignment

This is very similar to Configuration #1 of the single-level system simulation where the items are assigned to storage slots in increasing item number order. *Table 5.13* shows the system performance values obtained from this run.

Table 5.13 . System performance measures obtained from multi-level simulation model for Configuration #1

		Picking Tour	Storage Tour	Total Travel
Total Values (in 30 days)	**Distance (m)**	2667	3148	5815
	Time (rec.) (sec)	2819	3182	6001
	Time (chb.) (sec)	2542	3114	5656
Average Values (per tour)	**Distance (m\tour)**	99	67	-
	Time (rec.) (sec\tour)	104	68	-
	Time (chb.) (sec\tour)	94	66	-
Average Values (per day)	**Distance (m\day)**	99	105	-
	Time (rec.) (sec\day)	104	106	-
	Time (chb.) (sec\day)	94	104	-

5.2.2. Storage Assignment Configuration #2: Decreasing A_i

For the multi-level system, the second storage assignment configuration implemented was very similar to that of the single-level system where items are ordered in decreasing A_i order and are assigned to storage slots accordingly. In this configuration, items requiring more storage slots are placed to low level shelves and higher level shelves store items requiring a smaller number of storage slots. *Table 5.14* gives the result summaries of the run.

Table 5.14. System performance measures obtained from multi-level simulation model for Configuration #2

		Picking Tour	Storage Tour	Total Travel
Total Values (in 30 days)	**Distance (m)**	2849	2902	5751
	Time (rec.) (sec)	3035	3032	6067
	Time (chb.) (sec)	2687	2772	5459
Average Values (per tour)	**Distance (m/tour)**	106	64	-
	Time (rec.) (sec/tour)	112	67	-
	Time (chb.) (sec/tour)	100	62	-
Average Values (per day)	**Distance (m/day)**	106	97	-
	Time (rec.) (sec/day)	112	101	-
	Time (chb.) (sec/day)	100	92	-

5.2.3. Storage Assignment Configuration #3: Picking Oriented – Level-by-level

This configuration is the first of the picking oriented assignment configurations. Items are sorted according to decreasing demand frequency (obtained from the 30 days outgoing items data) and matched to storage slots sorted in increasing order of index. This results in highly demanded items being on the lower levels of the racks. As you go higher up the racks the demand frequency decreases. This assignment increases travel among the low level slots. The implementation of this configuration will allow seeing how the demand frequency effects the picking operation and how a single-level dominated picking influences the system performance. *Table 5.15*, below, shows the performance measures of the system with this storage assignment.

Table 5.15. System performance measures obtained from multi-level simulation model for Configuration #3

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2469	3490	5959
	Time (rec.) (sec)	2658	3604	6262
	Time (chb.) (sec)	2281	3376	5657
Average Values (per tour)	**Distance (m/tour)**	91	74	-
	Time (rec.) (sec/tour)	98	77	-
	Time (chb.) (sec/tour)	84	72	-
Average Values (per day)	**Distance (m/day)**	91	116	-
	Time (rec.) (sec/day)	98	120	-
	Time (chb.) (sec/day)	84	113	-

5.2.4. Storage Assignment Configuration #4: Picking Oriented – Line-by-line

This is the second picking oriented storage assignment configuration where items are again ordered according to their demand frequencies. This time more frequently demanded items are placed to aisles closer to the picking dock on all levels. So, the assignment is done aisle-by-aisle in decreasing item demand frequency order as opposed to the previous configuration where assignment was done level-by-level for the same ordering of items. When compared to the previous configuration, this assignment will decrease horizontal movement while increasing vertical movement. The order picker will pick more items from a single aisle while visiting a smaller number of aisles. Results of this run will show the influence of this movement strategy. Comparing configurations #3 and #4 will give whether smooth movement mostly on the lower levels or moving between levels is more efficient. More technically, this comparison will show whether horizontal or vertical movement in more effective on the system performance. *Table 5.16*, below, gives the result summaries of the simulation run.

Table 5.16. System performance measures obtained from multi-level simulation model for Configuration #4

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2713	3115	5828
	Time (rec.) (sec)	2843	3197	6040
	Time (chb.) (sec)	2580	3088	5668
Average Values (per tour)	**Distance (m/tour)**	100	66	-
	Time (rec.) (sec/tour)	105	68	-
	Time (chb.) (sec/tour)	96	65	-
Average Values (per day)	**Distance (m/day)**	100	104	-
	Time (rec.) (sec/day)	105	107	-
	Time (chb.) (sec/day)	96	101	-

5.2.5. Storage Assignment Configuration #5: Supply Oriented – Level-by-level

This configuration is storage oriented as opposed to the previous two assignments which were picking oriented. Here, similar to Configuration #3, items frequently arriving from suppliers are assigned to slots closer to the storage dock. Assignment of storage slots is done level-by-level to the list of items sorted in decreasing incoming frequency. That is, the lower level slots throughout the warehouse have items with higher incoming frequencies. *Table 5.17*, below, gives the results of this configuration.

Table 5.17. System performance measures obtained from multi-level simulation model for Configuration #5

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2870	2629	5499
	Time (rec.) (sec)	2998	2643	5641
	Time (chb.) (sec)	2751	2615	5366
Average Values (per tour)	**Distance (m/tour)**	103	56	-
	Time (rec.) (sec/tour)	107	56	-
	Time (chb.) (sec/tour)	98	56	-
Average Values (per day)	**Distance (m/day)**	103	88	-
	Time (rec.) (sec/day)	107	88	-
	Time (chb.) (sec/day)	98	87	-

5.2.6. Storage Assignment Configuration #6: Supply Oriented – Line-by-line

For comparison purposes, another storage oriented assignment was implemented where items frequently arriving from suppliers are assigned to storage slots located close to the picking dock in a line-by-line manner very similar to the assignment done in Configuration #4. The table below shows the results of this assignment.

Table 5.18. System performance measures obtained from multi-level simulation model for Configuration #6

		Picking Tour	**Storage Tour**	**Total Travel**
Total Values (in 30 days)	**Distance (m)**	2703	1956	4659
	Time (rec.) (sec)	2890	2090	4980
	Time (chb.) (sec)	2531	1822	4353
Average Values (per tour)	**Distance (m/tour)**	100	42	-
	Time (rec.) (sec/tour)	107	44	-
	Time (chb.) (sec/tour)	94	39	-
Average Values (per day)	**Distance (m/day)**	100	65	-
	Time (rec.) (sec/day)	107	70	-
	Time (chb.) (sec/day)	94	61	-

5.2.7. Comments on the Multi-Level System Storage Assignment Configurations

For the multi-level system, the configurations will be compared according to the total rectilinear travel time in seconds. It is assumed that currently the forklift movements are sequential and so rectilinear measures are used. Comments about the previously stated simulation runs for the multi-level warehouse are given after the table showing the performance measure orders of the storage assignment configurations under different comparison types.

Table 5.19. Storage Assignment Configurations' Performance Summary (Multi-Level System)

Comparison Type	Performance Measure	Configuration Order	Performance Measure Value
Picking Oriented Assignments	Total Rectilinear Travel Time (sec)	#4, #3	6067 < 6262
Storage Oriented Assignments	Total Rectilinear Travel Time (sec)	#6, #5	4980 < 5641
Picking vs. Storage	Total Rectilinear Travel Time (sec)	#6, #5, #4, #3	4980 < 5641 < 6040 < 6262
All Assignments	Total Rectilinear Travel Time (sec)	#6, #5, #1, #4, #2, #3	4980 < 5641 < 6001 < 6040 < 6067 < 6262

1. Among the picking oriented storage assignment configurations, the line-by-line assignment (i.e. #4) is the best. The reasoning behind this conclusion is that in Configuration #4 all frequently demanded items are placed at the aisles close to the picking dock on all levels of the shelves. This minimizes the horizontal distance traveled, which is the larger portion of the total distance traveled during a pick tour.

2. An average save of 9sec/tour is saved when the system switches from Configuration #3 to #4. Storage assignment Configuration #4 tends to minimize the number of aisles visited in a pick tour while increasing the travel between levels in an aisle whereas the 3rd configuration decreases travel between levels but increases the number of aisles visited in a pick tour. Therefore, it can be concluded that visiting less aisles to the extent possible is more efficient than staying on the same level during a pick tour.

3. Configuration #6 shows better performance than Configuration #5, due to the very same reasoning of the picking oriented case. Assigning items frequently arriving at the warehouse from suppliers to close aisles without any level restriction is better than placing them at the lower level slots.

4. Similar to the case in the low-level system, no generalization can be made about random storage assignments in high-level systems. Here, too, an arbitrary assignment of items to slots (i.e. #1) shows good performance when compared to the other assignments but this is only because the arbitrary choice happens to be a good one.

5. The results and order placement of Configuration #2 shows that an A_i dependent configuration is not as efficient as a picking oriented storage assignment configuration. However, it, likely due to coincidence, shows better performance than Configuration #3.

6. Supply oriented assignments show better performance according to the values given in the previous tables. However, the same reasoning as that of the single-level system lies behind the unsuitability of the implementation of storage oriented assignments in warehouses.

7. Allowing the order picker to move simultaneously in the horizontal and vertical directions during picking tours minimizes the total travel time by 8.22%.

Chapter 6

SIMULATION MODEL VERIFICATION AND VALIDATION

In general, simulation models are developed for various purposes and serve management as a decision tool. To be able to make the right decision, simulation results are analyzed. At this point, the accuracy of the results and the validity of the model are of importance. The model's validity must be measured with respect to its aim or purpose. A simulation model will be considered "valid" if its output variables are in the required accuracy level of the model's purpose. For a model to be valid, both *verification* and *validation* must be done.

Validation gives whether or not the real system is presented accurately in the computerized simulation models whereas verification determines whether or not the computerized model accurately represents the model envisioned in the mind of the analyst. Banks et al (2001) define these two concepts with simple key words as "Verification is building the model right" and "Validation is building the right model". Sargent (2001) gives more exact definitions for them. "Model verification is ensuring that the computer program of the computerized model and its implementation are correct". "Model validation is substantiation that a computerized model within its domain applicability processes a satisfactory range of accuracy consistent with the intended application of the model".

Model verification and validation can either be done during the model development phase or after it. Sargent (2001) summarizes four ways as to how a model can be determined as valid.

1. The model development team, itself, makes the decision of the model validity. They make this decision based on the results of various tests and evaluations conducted during the model development phase.

2. The users of the model decide whether it is valid or invalid. These users are usually those who are in communication with the model development team during the development phase, and so can view this process and see the results during the development of the model.

3. *"Independent Verification & Validation" (IV&V)* Here a third (independent) person makes the validity decision. In this approach, since the third person is not one of the model developers or one of the users, he/she must have a good understanding of what the model does and its purpose. He/she can conduct IV&V either during the development phase, or after the model has been developed.

4. The last approach is to use a scoring model. Various aspects of the model are scored during the validation process, and then combined to determine category scores and an overall score for the simulation model. The simulation model is said to be valid if its overall and category scores are greater than some passing score.

6.1. Model Verification

Banks et al. (2001) gives seven common sense ways of verifying a simulation model. These are:

1. Have the computerized representation checked by someone other than its developer.

2. Make a flow diagram which includes each logically possible action a system can take when an event occurs, and follow the model logic for each action for each event type.

3. Closely examine the model output for reasonableness under a variety of settings of the input parameters. Have the computerized representation print out a wide variety of output statistics.

4. Make the computerized representation print the input parameters at the end of the simulation, to be sure that these parameter values have not been changed inadvertently.

5. Make the computerized representation as self-documenting as possible. Give a precise definition of every variable used and a general description of the purpose of each major section of the code.

6. If the computerized representation is animated, verify that what is seen in the animation initiates the actual system.

7. The use of the interactive run controllers or debuggers of the computer program used for simulation.

8. Graphical interfaces are recommended for accomplishing verification and validation. The graphical representation of the model is essentially a form of self-documentation. It simplifies the task of model understanding.

In this study, both for the single-level and multi-level simulation models, verification is done partly during the model development phase and partly after the development. An important characteristic of both models is that they are deterministic simulation models where no probabilistic values are present. Therefore, this simplifies the verification of the models since the results can be generated *a priori* by calculations or simply hand simulations. It is a long process, but can be done if desired. The verification steps for both single-level and multi-level models are as:

1. The first way to verify that the model is working without any problem is to debug the system with the computer program's debugger while building the model. This ensures that there are no programming related errors in the model. However, it cannot detect logic errors. This step corresponds to suggestion # 7 stated by Banks et al. (2001).

2. To detect logic errors, the model is developed and controlled part-by-part. Once each part is built, it is run with some small sample data for a small replication time period to give a small output which can be easily computed and controlled by hand simulation. During this phase of model development, some of the input parameters and some control variables (variables that are not output data but are used in the model to run things smoothly and accurately, e.g. counters) of the system are printed to the screen to be able check their values throughout the simulation run. Any inconsistency in these values will give the logic errors of the computerized model. This is a similar action as to what was suggested by Banks et al. (2001) in #4.

3. When each sub model is verified to be working logically (i.e. as intended) on its own, they are integrated to form the main model. Then, again, the model is run with the small sample data set and the output generated is controlled with the results of the hand simulation. Suggestion #5 by Banks et al. (2001) is applied to all sub models of the simulation model and so no confusion occurs when integrating them. The definitions are added to the Visual Basic codes as comment lines.

4. Lastly, the large data set generated for a 30-days replication is input to the simulation model. To verify the results of this run, random output values are taken from the output set and are controlled by hand simulation.

During any one of these steps, once a syntax or logic error is detected, the necessary sections of the model are revised and corrected. The simulation model is, then, run again with the same sample set to check if the error is removed.

6.2 Model Validation

For operational validation, whether or not the real system is observable (i.e. possible to collect data on the operational behavior of the system) is important. When the real system is observable, it is possible to compare the output performance values of the real system and the simulation system to determine if the simulation model was developed accurately. A widely accepted three-step approach was presented by Naylor & Finger (1967) to aid in model validation. Banks et al. (2001) explains this approach in detail. The approach is as:

1. *Build a model that has high face validity*: This means to construct a model that appears reasonable on its face to model users and others who have knowledge about the real system being simulated.

2. *Validation of model assumptions*: There are two types of model assumption, namely structural assumptions and data assumptions. Structural assumptions deal with how the system works and include simplifications from the real system. These assumptions must reflect the real system with as little deviation as possible. Data assumptions are related to the collection of data. The reliability of data can be tested by statistical tests.

3. *Validating input-output transformations*: In this phase, the transfer of input data to output parameters is validated. One way to do this is to use past input data from the real system, and see if the model gives the correct output data (i.e. does it match the value from the past data)

Face validity and structural assumptions validity were done with the help of people knowing the system well. Input-output transformations validity was done to see if the system gives reasonable output data from the input data. However, past data was not available and so the models could not be compared with the real warehouse.

Chapter 7

CONCLUSION

The problem environment, in this study, is a spare parts warehouse for an automotive company in Turkey. The warehouse has a multi-level storage system where incoming items are stored in unit loads (i.e. bulks) and outgoing items are picked in numbers of units and sent to the company's automobile service centers around Turkey. Currently, incoming items are assigned to storage slots randomly based on the operator's own experience. Management is aware of the inefficient assignment procedure being followed within the warehouse, and so, has brought this problem to our attention.

As the project is currently proceeding, the first step is to derive an insight of *what* is going on at a warehouse in general. That is, what are the performance-related operations within a warehouse (i.e. which operations have a direct impact on the warehouse efficiency performance), which factors must be taken as performance measures of a warehouse, and what types of storage assignments show better performance. For this reason, after the real warehouse was explored and some insight was gained from the literature, two simulation models were constructed, one single-level and one multi-level model, where warehouse operations were run for a 30-days time period to obtain a total travel distance measure. It was noticed that only the picking and storage processes in a warehouse have direct impact on the warehouse efficiency and so only these two operations were included in the simulation models. Related to this fact, the performance measures of the single-level and multi-level systems were assumed to be total travel distance and total travel time, respectively.

Alternative storage assignment configurations, all based on a dedicated storage assignment policy, were implemented to both systems and their results were compared. The assignment alternatives included arbitrary, picking oriented, and storage oriented assignments.

For both single-level and high-level storage systems it was observed that arbitrarily chosen assignment structures do not always show good performance. This is obvious, as they are called "arbitrary" and so may or may not turn out to be "good". Therefore, management of the real warehouse in consideration is right about remarking their concerns. Second, assigning items incoming/outgoing frequently to storage slots located closer to the storage/picking dock with an I-shape manner results in higher efficiency in a single-level system. A similar case is present in multi-level systems where assigning items incoming/outgoing frequently to storage slots close to the storage/picking dock by a line-by-line manner shows better performance. The third observation, in general for both systems, is that storage-oriented assignments show better performance in warehouses where even 65% of the total transactions are picking transactions since they decrease the total distance traveled during multiple single-stop tours, therefore decreasing the total distance traveled within the warehouse. It may, however, be expected that if this percentage rises to, say, 80% a picking-oriented assignment will be better. At what point this shift will occur is an open question.

BIBLIOGRAPHY

1. Banks J., Carson J.S., Nelson B.L., Nicol D.M. (2001), “Discrete-event system simulation”, 3rd ed., *Prentice Hall*, Upper Saddle River, NJ

2. Caron F., Marchet G., Perego A (1998), “Routing policies and COI-based storage policies in picker-to-part systems”, *International Journal of Production Research*, Vol. 36, No. 3

3. Caron F., Marchet G., Perego A. (2000a), “Layout Design in Manual Picking Systems: a simulation approach”, *Integrated Manufacturing Systems*, Vol. 11, No. 2

4. Caron F., Marchet G., Perego A. (2000b), “Optimal Layout in Low-Level Picker-To-Part Systems”, *Int. J. Prod. Res.*, Vol. 38, No. 1

5. Chew E.P.; Tang L.C. (1999), “Travel time analysis for general item location assignment in a rectangular warehouse”, *European Journal of Operational Research*, Vol.112, No.3

6. De Koster R.,Van Der Poort E. (1998), “Routing order pickers in a warehouse: a comparison between optimal and heuristic solutions”, *IIE Transactions*, Vol.30, No.5

7. De Koster R., Le-Duc T., Roodbergen K. J. (2007), “Design and control of warehouse order picking: a literature review”, *European Journal of Operational Research*, Vol.182, No.2

8. Dallari F., Marchet G., Ruggeri R (2000), “Optimization of man-on-board automated storage\retrieval systems”, *Integrated Manufacturing Systems*, Vol.11, No.2

9. Hassan, M.M.D. (2002), “A framework for the design of warehouse layout”, *Facilities*, Vol. 20, No.13-14

10. Heragu S.S., Du L., Mantel R.J., Schuur P.C. (2005), “Mathematical model for warehouse design and product allocation”, *International Journal of Production Research*, Vol. 43, No. 2

11. Hsieh S, Tsai K-C. (2001), “A BOM Oriented Class-Based Storage Assignment in an Automated Storage/Retrieval System”, *The International Journal of Advanced Manufacturing Technology*,Vol.17, No. 9

12. Hsieh S., Tsai K-C. (2006), “The optimum design of a warehouse system on order picking efficiency”, *International Journal of Advanced Manufacturing Technology*, Vol.28

13. Jewkes E., Lee C., Vickson R. (2004), “Product location, allocation and server home base location for an order picking line with multiple servers”, *Computers and Operations Research***,** Vol.31, No. 4

14. Le-Duc T., De Koster B.M. (2005), “Travel distance estimation and storage zone optimization in a 2-block class-based storage strategy warehouse”, *International Journal of Production Research*, Vol. 43, No. 17

15. Malmborg, C.J. Krishnakumar, B. (1989), “Optimal storage assignment policies for multiaddress warehousing systems”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 2

16. Manzini R., Gamberi M., Persona A., Regattieri A. (2007), “Design of a class based storage picker to product order picking system”, *The International Journal of Advanced Manufacturing Technology,* Vol.32, No.7-8

17. Muller D.J. (1989), “AS/RS and Warehouse Modeling”, *Proceedings of the 1989 Winter Simulation Conference*

18. Naylor T.H., Finger J.M. (1987), “Verification of Computer Simulation Models”, *Management Science*, Vol.2, ppB92-B101

19. Petersen C.G., (1999), “The impact of routing and storage policies on warehouse efficiency”
International Journal of Operations & Production Management, Vol. 19 No. 10

20. Petersen C.G., Aase G. (2003), “A comparison of picking, storage, and routing policies in manual order picking” *Int. J. Production Economics*

21. Petersen C.G., Aase G.R. (2004), “Improving order-picking performance through the implementation of class-based storage” *International Journal of Physical Distribution & Logistics*

22. Ratliff, H.D. and Rosenthal, A.S. (1983) “Order picking in a rectangular warehouse: a solvable case of the traveling salesman problem”, *Operations Research*, Vol.31, No.3

23. Rouwenhorst B., Reuter B., Stockrahm V., Houtum G.J., Mantel R.J., Zijm W.H.M. (2000), “Warehouse design and control: Framework and literature review” *European Journal of Operational Research*, Vol.122, No. 3

24. Roodbergen K.J., De Koster R. (2001), “Routing order pickers in a warehouse with a middle aisle”, *European Journal of Operational Research*, Vol.133, No.1

25. Roodbergen K.J., Iris F., Vis A. (2006), “A model for warehouse layout”, *IIE Transactions*, Vol. 38, No.10

26. Ruben R. A.; Jacobs F.R. (1999), "Batch Construction Heuristics and Storage Assignment Strategies for Walk/Ride and Pick Systems", *Management Science*, Vol. 45, No. 4

27. Sargent R.G., (2001), "Some Approaches and Paradigms for Verifying and Validating Simulation Models", *Proceedings of the 2001 Winter Simulation Conference*

APPENDIX

Section A

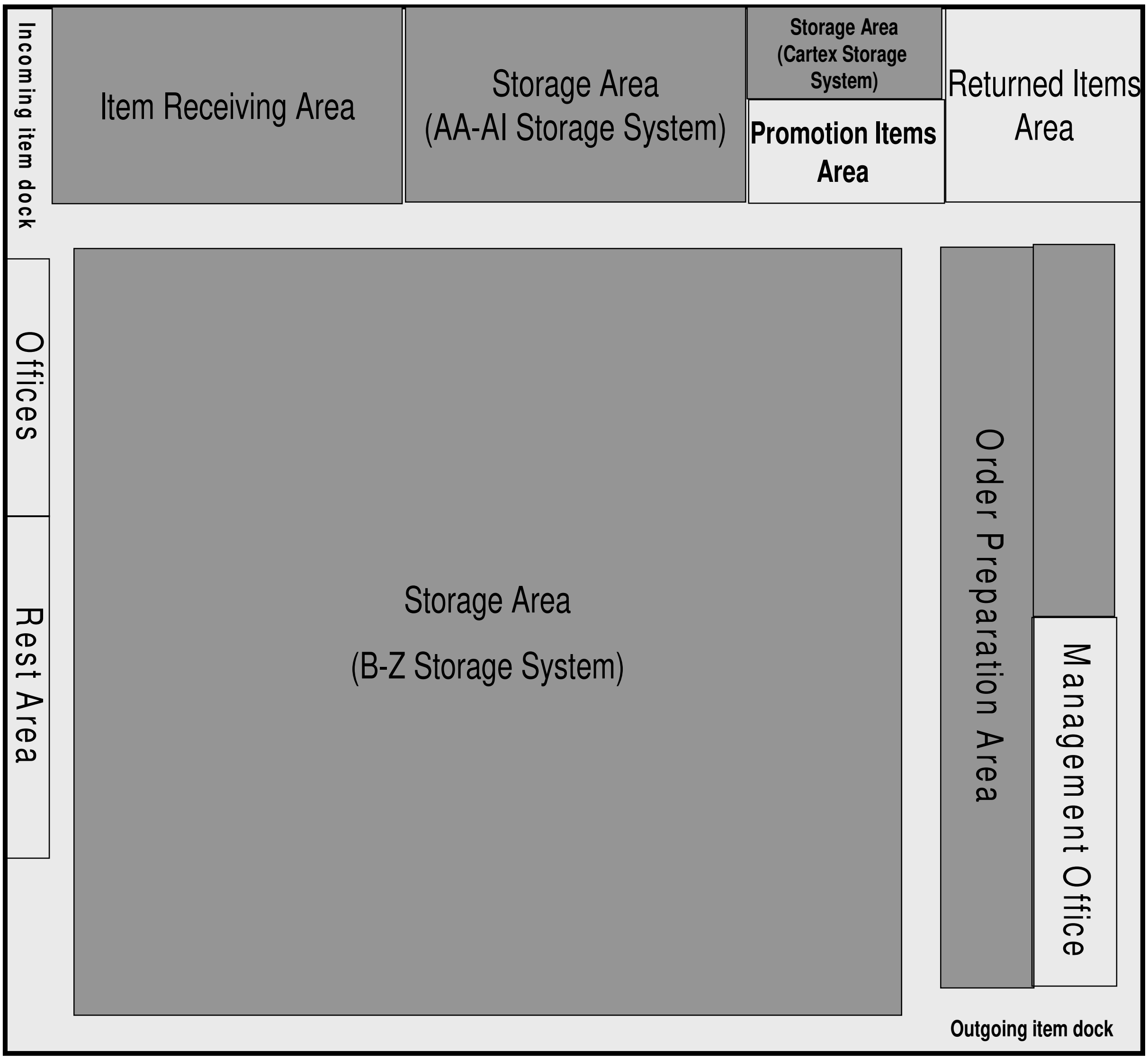


Figure A1. Block Plan of Warehouse (block sizes are not of accurate ratio)

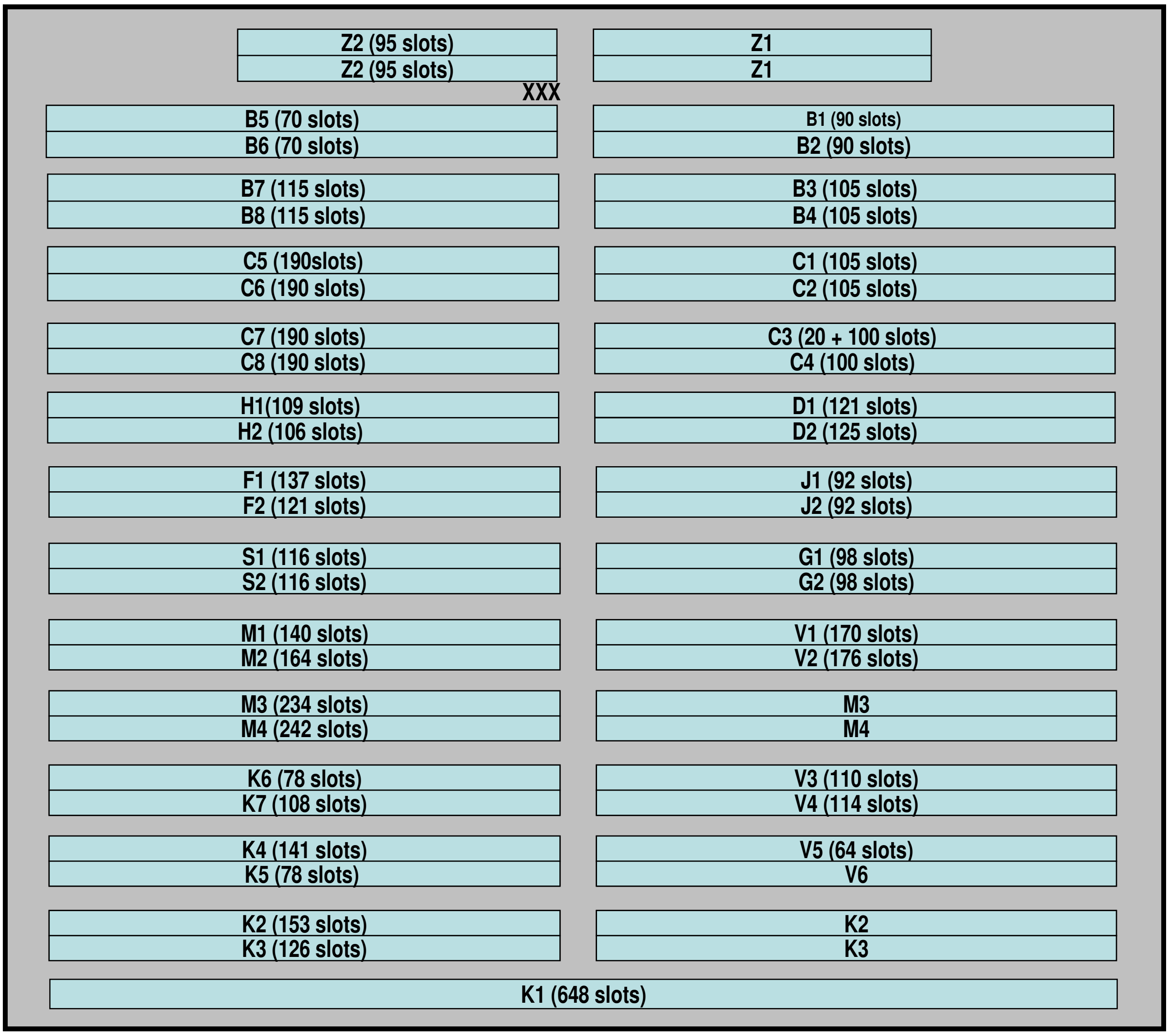


Figure A2. Aisle Layout of B-Z Storage System (XXX indicating the temporary storage point)

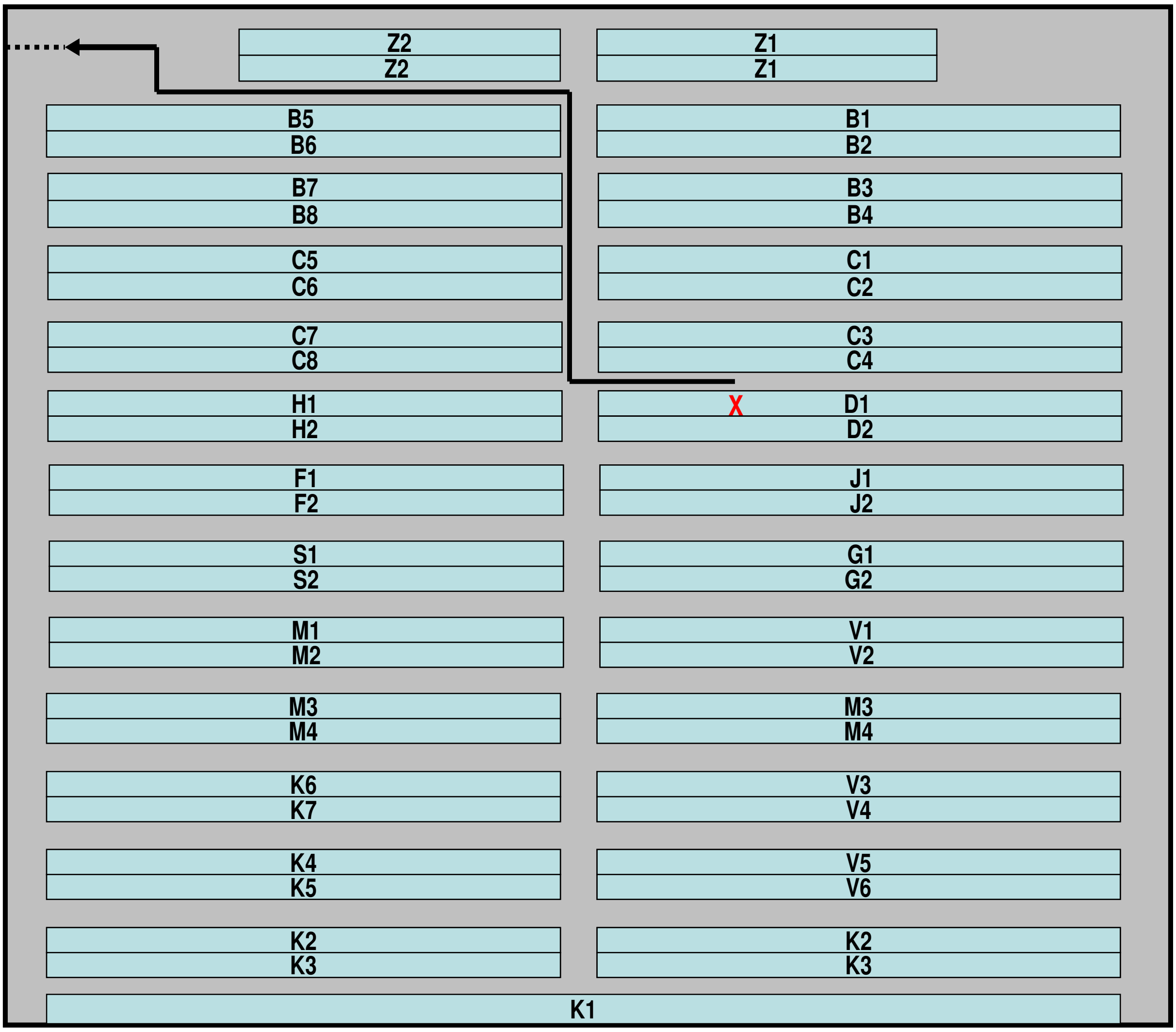


Figure A3. Example of a Single-Stop Storage Tour in B-Z Shelf System Area

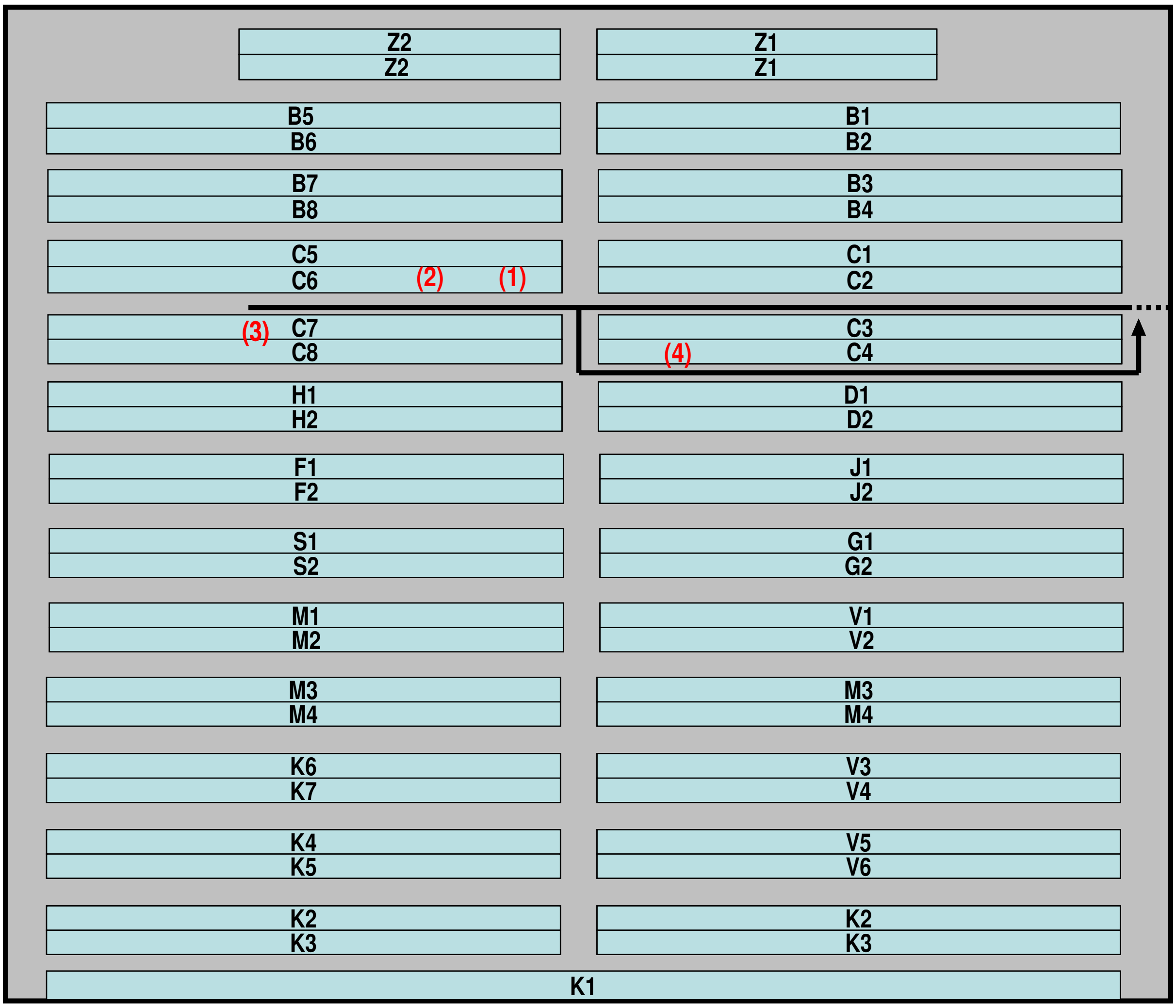


Figure A4. Example of a Multi-Stop Storage Tour in B-Z Shelf System Area

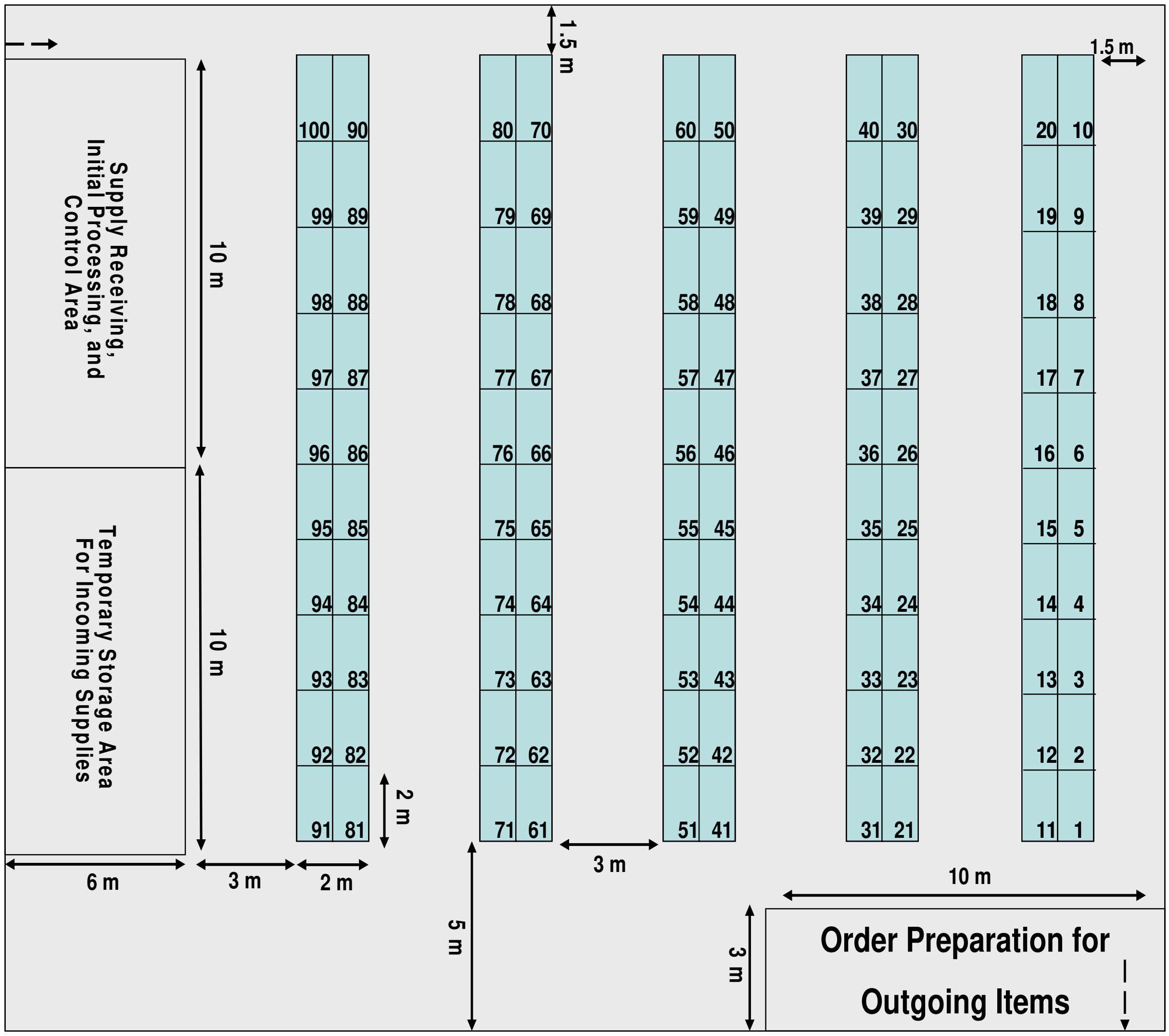


Figure A5. Single-Level Warehouse Layout

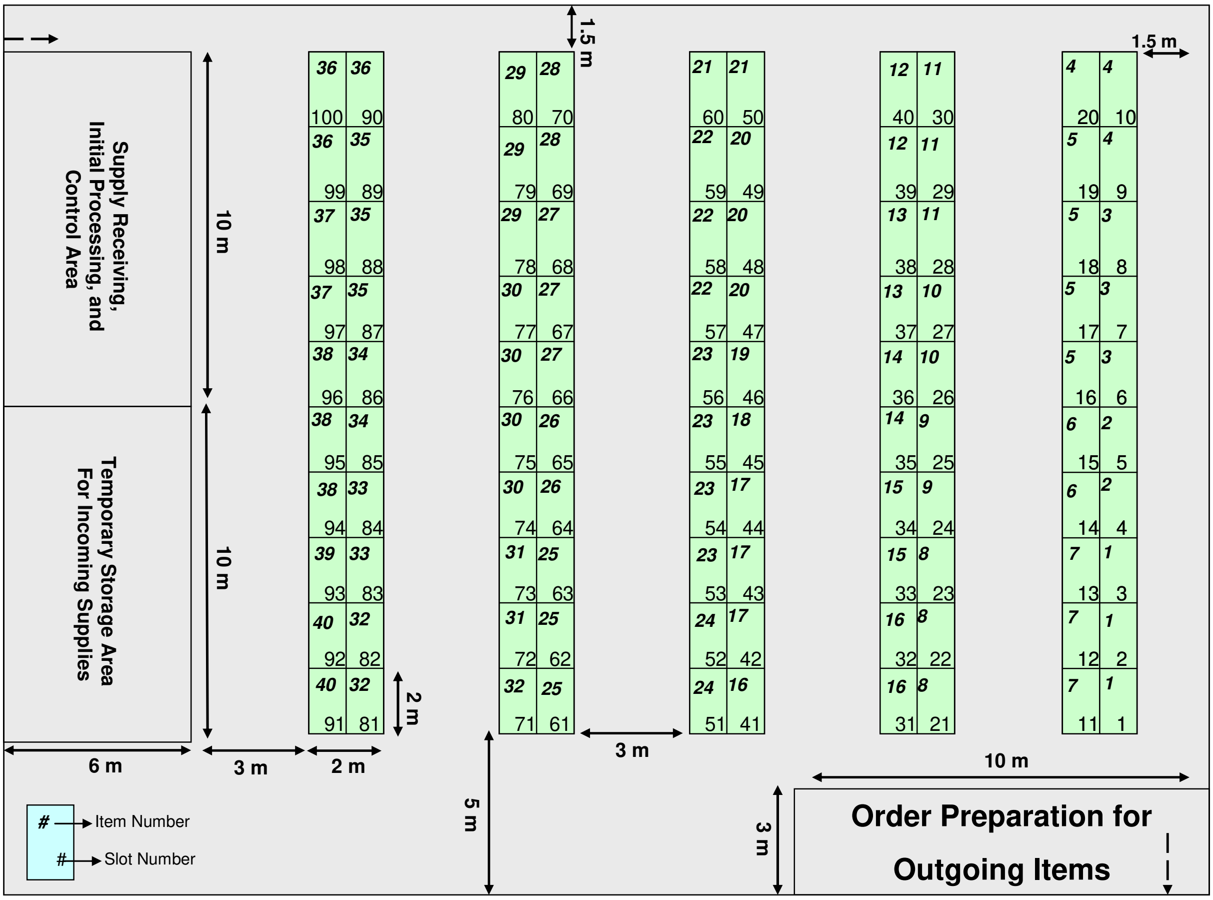


Figure A6. Single-Level Storage System: Configuration #1

Figure A7. Single-Level Storage System: Configuration #2

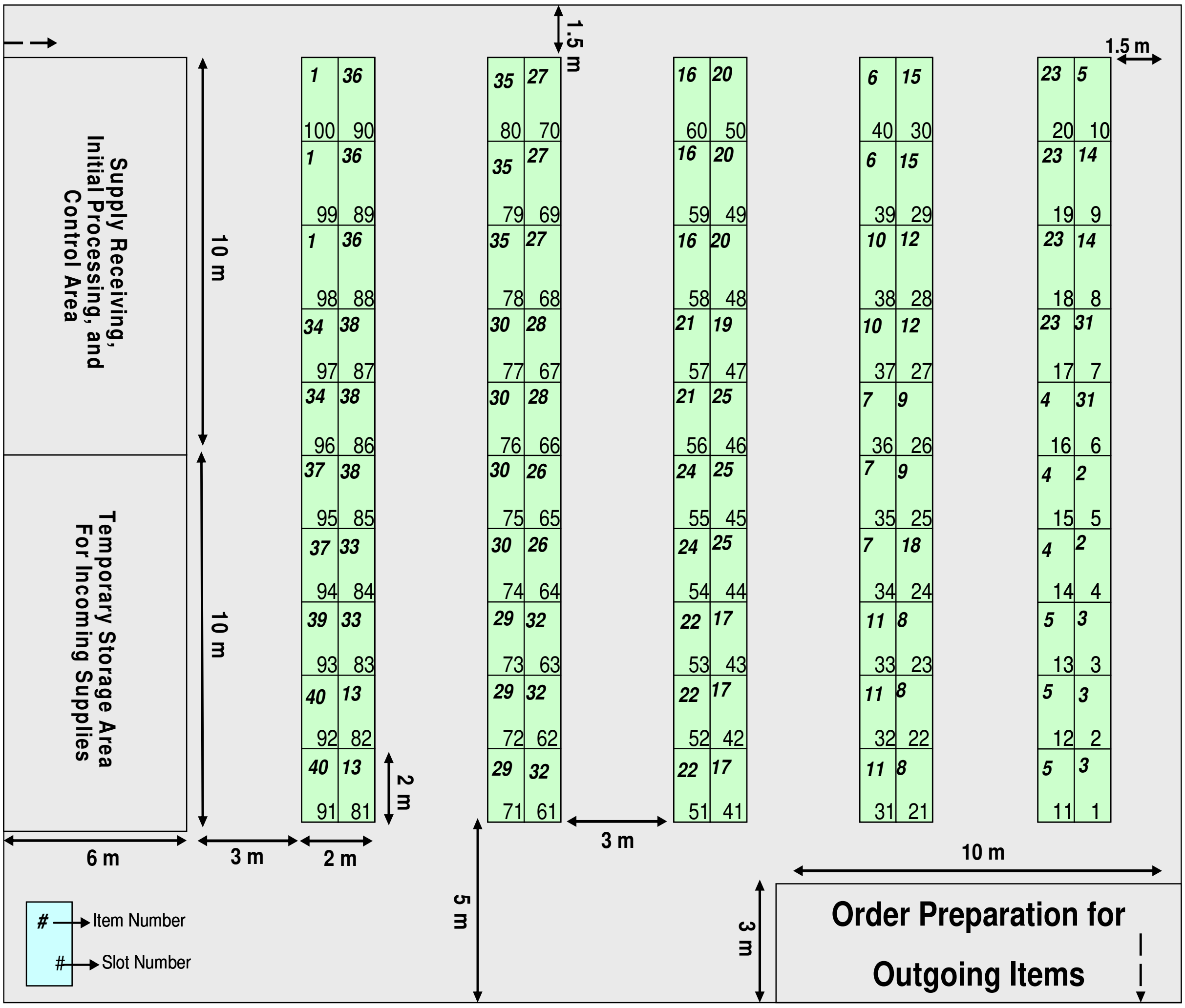


Figure A8. Single-Level Storage System: Configuration #3

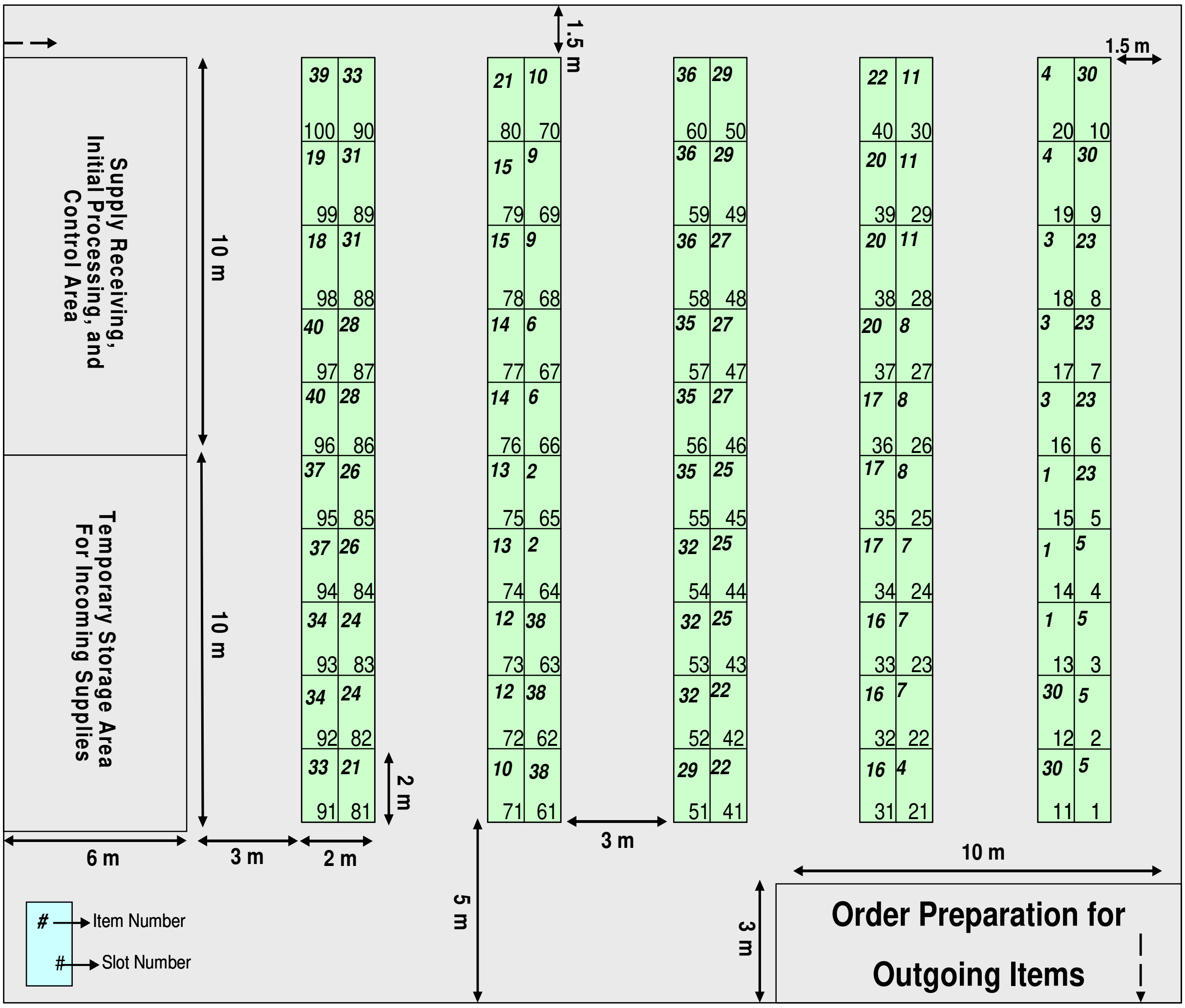
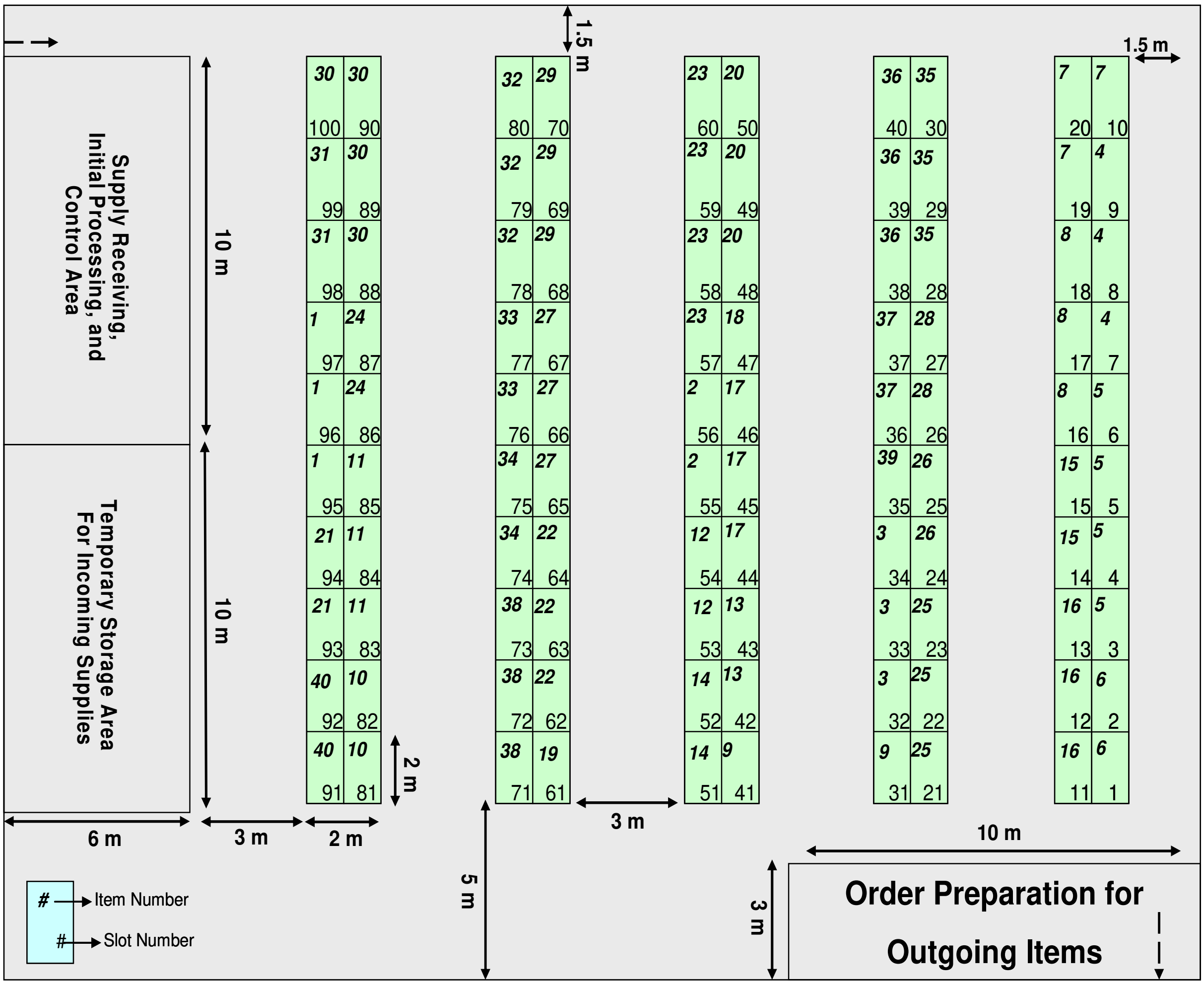


Figure A9. Single-Level Storage System: Configuration #4



30 100	30 90	32 80	29 70	23 60	20 50	36 40	35 30	7 20	7 10
31 99	30 89	32 79	29 69	23 59	20 49	36 39	35 29	7 19	4 9
31 98	30 88	32 78	29 68	23 58	20 48	36 38	35 28	8 18	4 8
1 97	24 87	33 77	27 67	23 57	18 47	37 37	28 27	8 17	4 7
1 96	24 86	33 76	27 66	2 56	17 46	37 36	28 26	8 16	5 6
1 95	11 85	34 75	27 65	2 55	17 45	39 35	26 25	15 15	5 5
21 94	11 84	34 74	22 64	12 54	17 44	3 34	26 24	15 14	5 4
21 93	11 83	38 73	22 63	12 53	13 43	3 33	25 23	16 13	5 3
40 92	10 82	38 72	22 62	14 52	13 42	3 32	25 22	16 12	6 2
40 91	10 81	38 71	19 61	14 51	9 41	9 31	25 21	16 11	6 1

Figure A10. Single-Level Storage System: Configuration #5

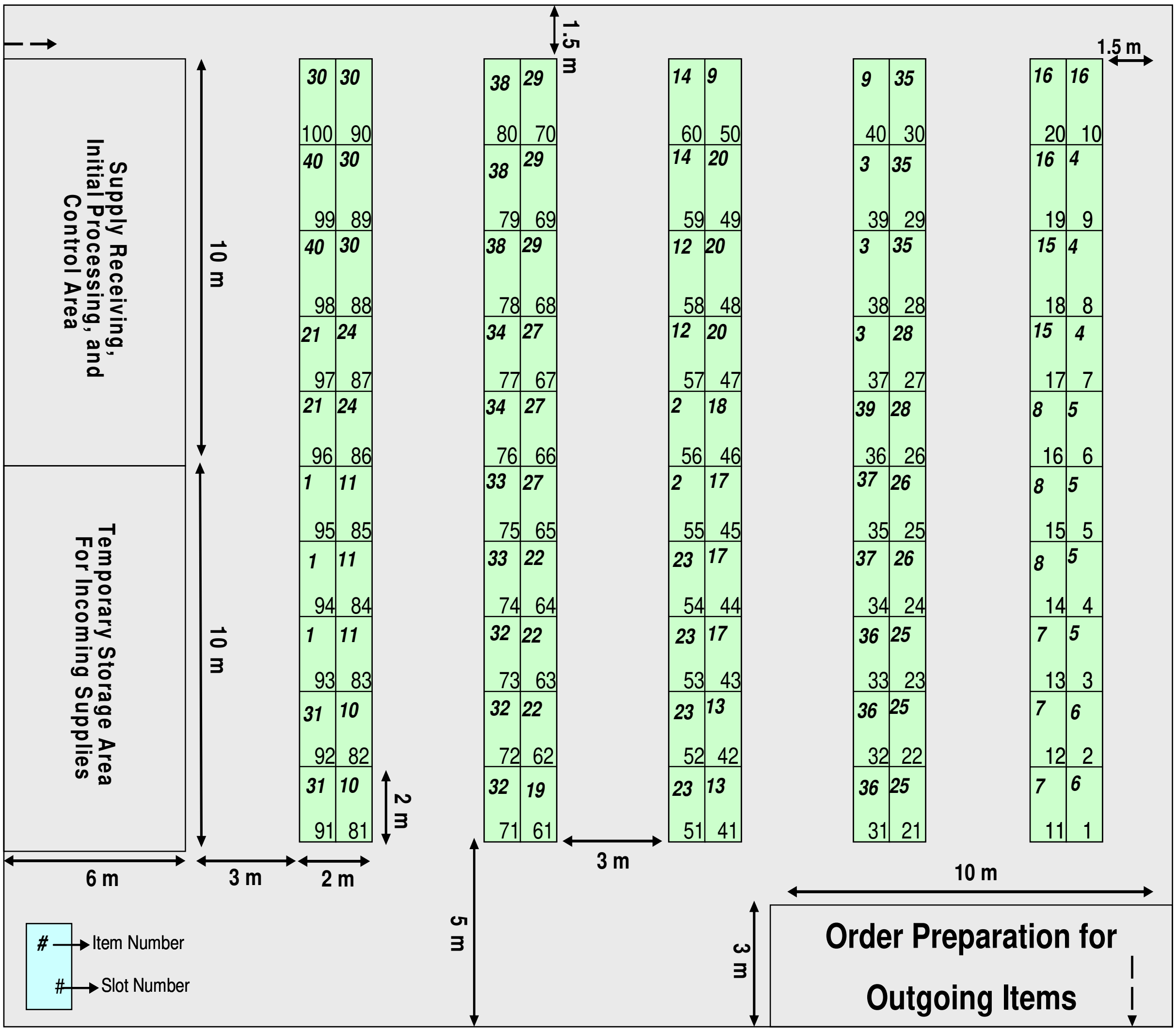


Figure A11. Single-Level Storage System: Configuration #6

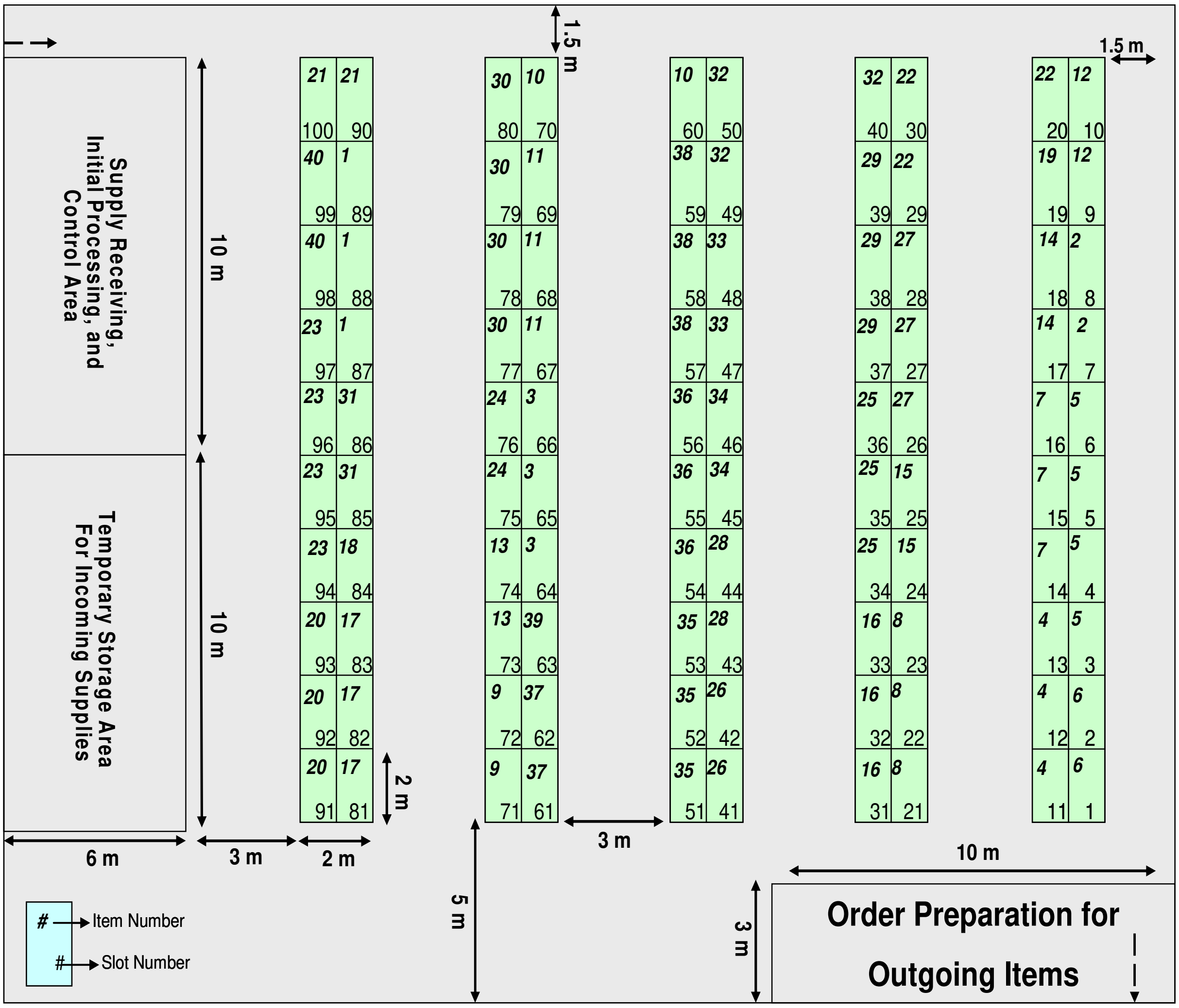


Figure A12. Single-Level Storage System: Configuration #7

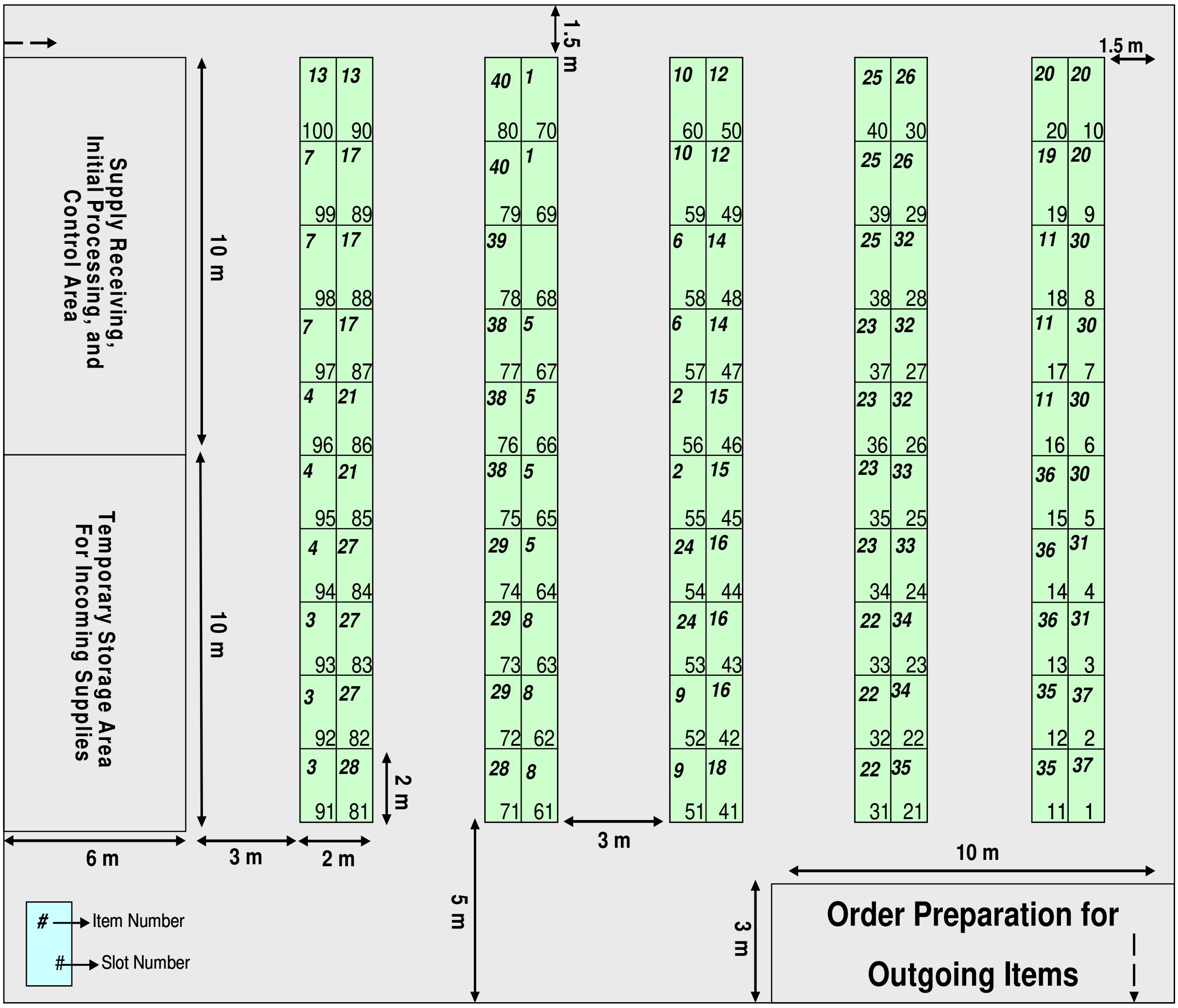


Figure A13. Single-Level Storage System: Configuration #8

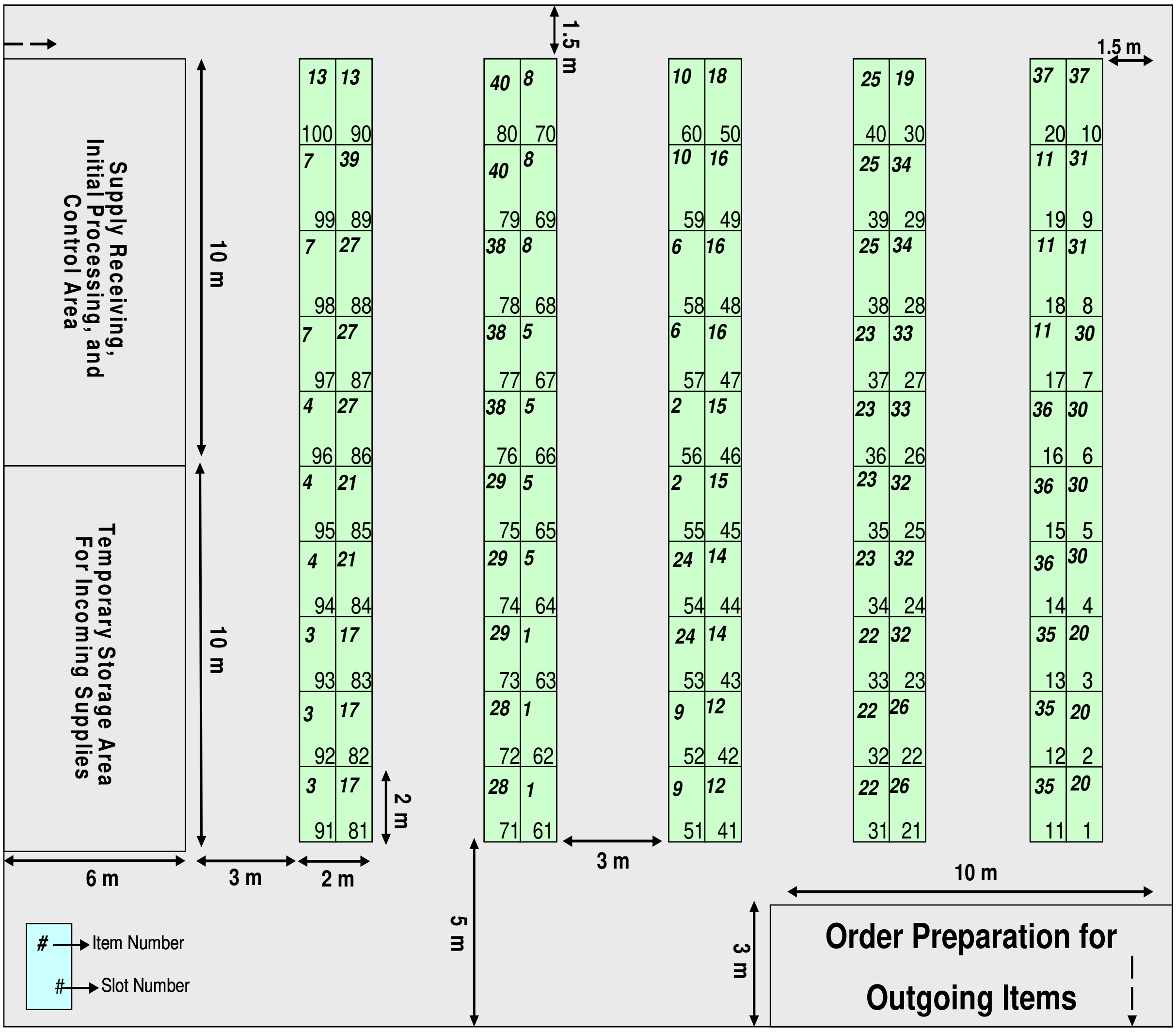


Figure A14. Single-Level Storage System: Configuration #9

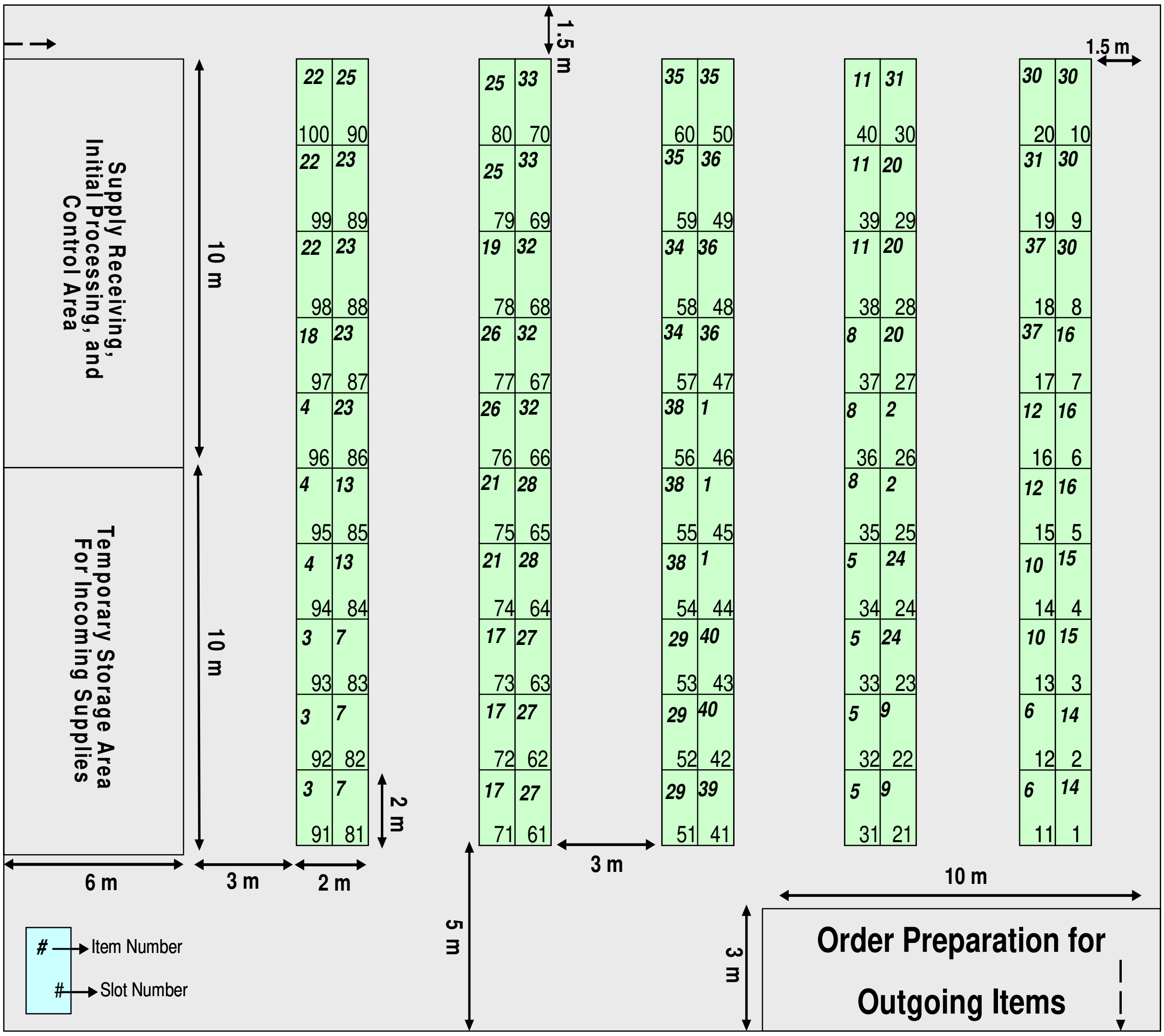


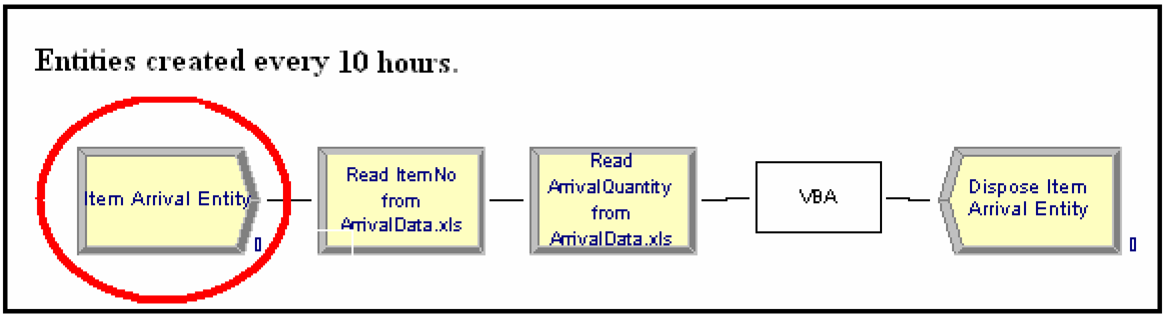
Figure A15. Single-Level Storage System: Configuration #10

APPENDIX

Section B

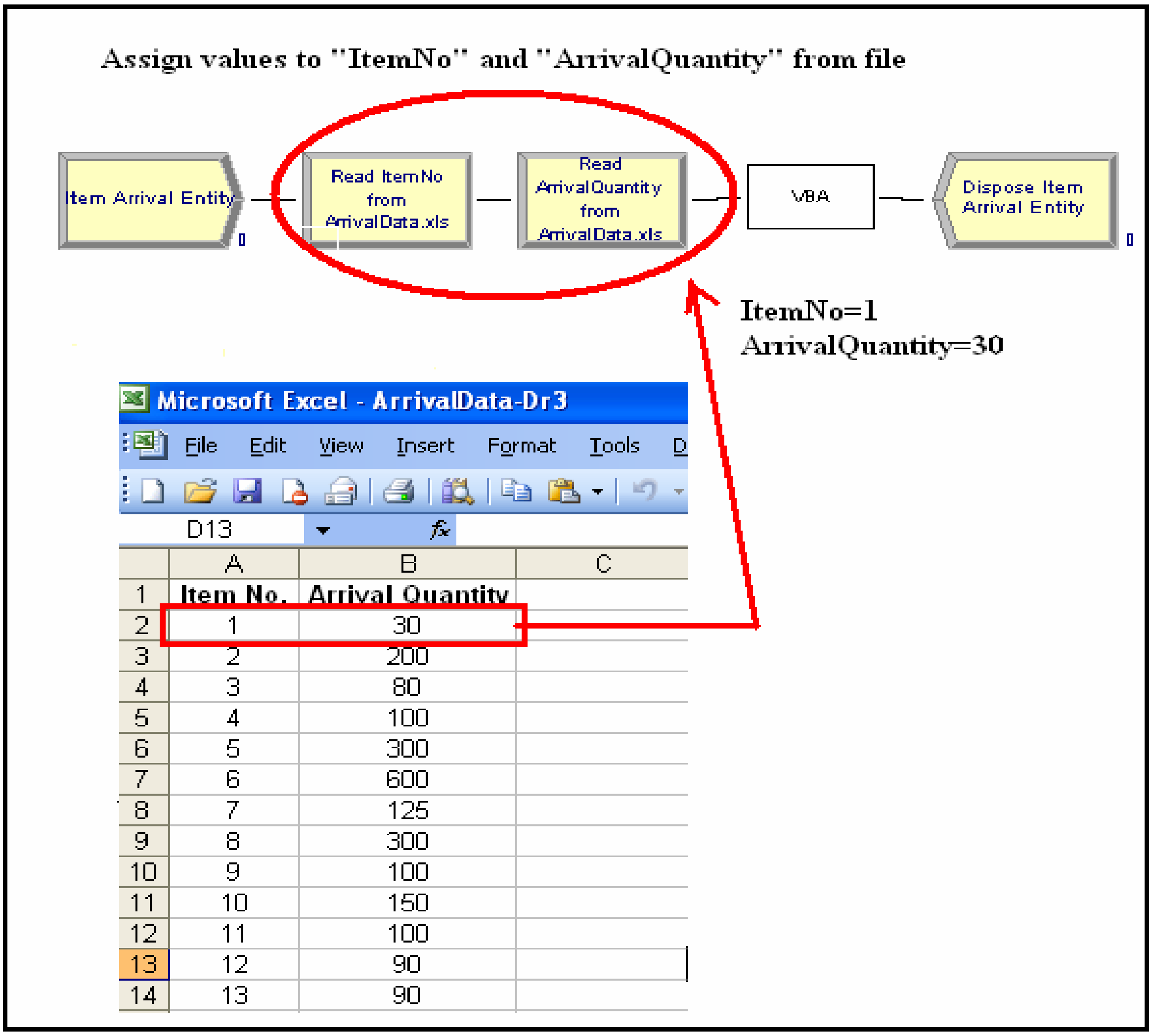
Simulation Manual for the Storage Sub-model

Step 1s:



Entities created every 10 hours.

Step 2s:

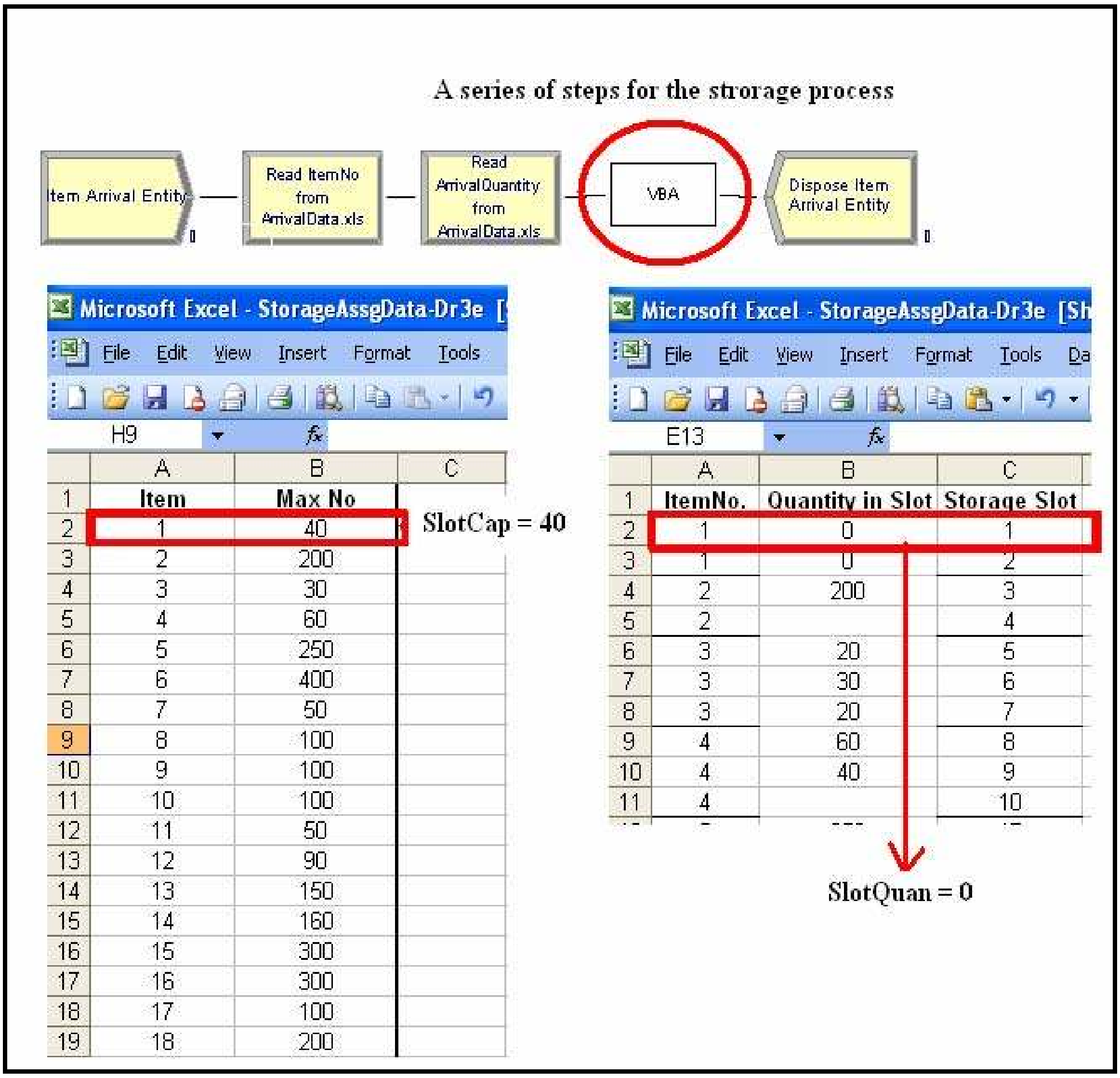


Assign values to "ItemNo" and "ArrivalQuantity" from file

ItemNo=1
ArrivalQuantity=30

	A	B	C
1	Item No.	Arrival Quantity	
2	1	30	
3	2	200	
4	3	80	
5	4	100	
6	5	300	
7	6	600	
8	7	125	
9	8	300	
10	9	100	
11	10	150	
12	11	100	
13	12	90	
14	13	90	

Step 3s:



H9

	A	B	C
1	Item	Max No	
2	1	40	
3	2	200	
4	3	30	
5	4	60	
6	5	250	
7	6	400	
8	7	50	
9	8	100	
10	9	100	
11	10	100	
12	11	50	
13	12	90	
14	13	150	
15	14	160	
16	15	300	
17	16	300	
18	17	100	
19	18	200	

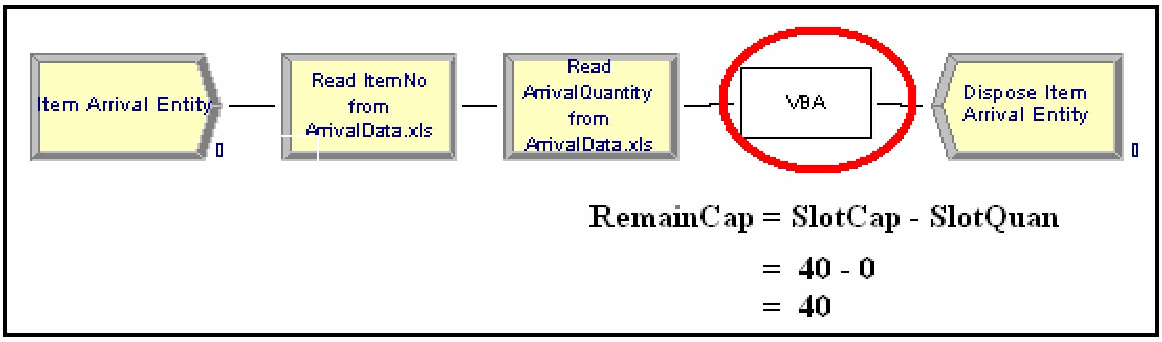
SlotCap = 40

E13

	A	B	C
1	ItemNo.	Quantity in Slot	Storage Slot
2	1	0	1
3	1	0	2
4	2	200	3
5	2		4
6	3	20	5
7	3	30	6
8	3	20	7
9	4	60	8
10	4	40	9
11	4		10

SlotQuan = 0

Step 4s:



RemainCap = SlotCap - SlotQuan

= 40 - 0

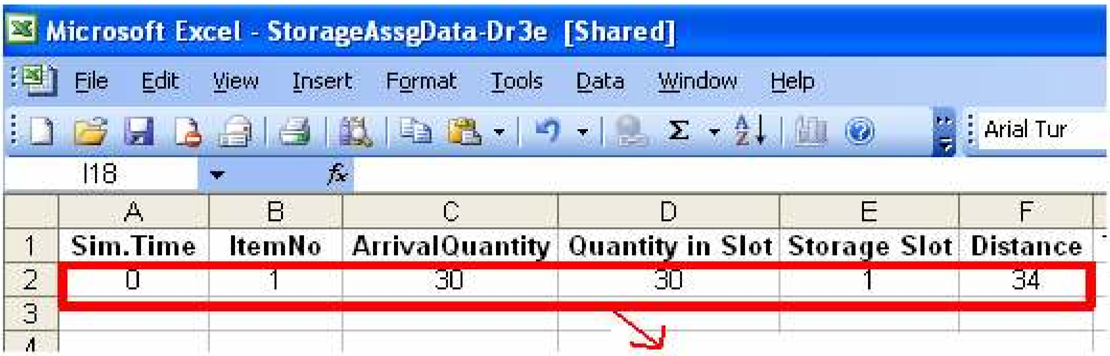
= 40

Step 5s-True:

RemainCap >= ArrivalQuantity

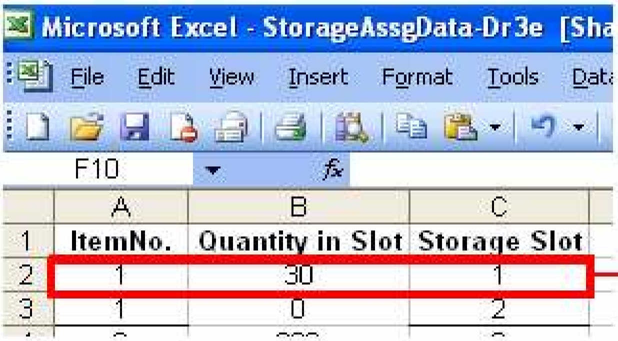
40 >= 30 → TRUE

Then, SlotQuan = SlotQuan + ArrivalQuantity
= 0 + 30
= 30



	A	B	C	D	E	F
1	Sim.Time	ItemNo	ArrivalQuantity	Quantity in Slot	Storage Slot	Distance
2	0	1	30	30	1	34
3						

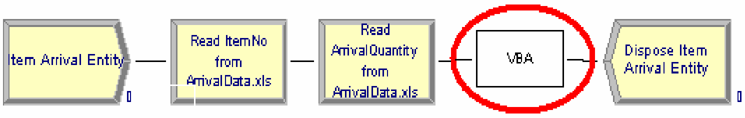
Add transaction details



	A	B	C
1	ItemNo.	Quantity in Slot	Storage Slot
2	1	30	1
3	1	0	2

Update "Quantity in Slot" from 0 to 30

Step 5s-False:



Assume that SlotQuan = 25 and so RemainCap = 40 - 25 = 15
from previous steps

RemainCap >= ArrivalQuantity

$15 \quad >= \quad 30 \longrightarrow$ FALSE

RemainCap > 0

So, SlotQuan = SlotCap

= 40

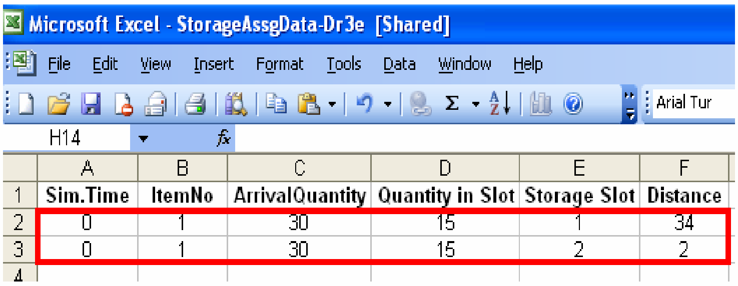
and LeftQuan = ArrivalQuantity - RemainCap

= 30 - 15

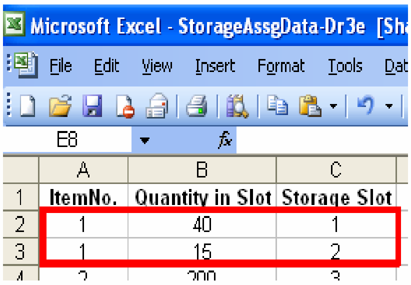
= 15

Revise steps 3s, 4s, 5s, 6s until LeftQuan = 0

and the following transactions and updates are made to the files



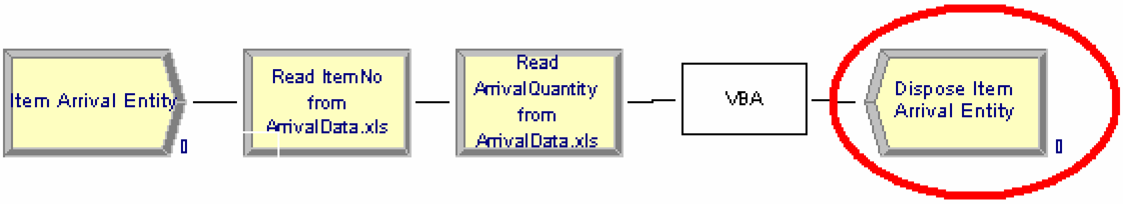
	A	B	C	D	E	F
1	Sim.Time	ItemNo	ArrivalQuantity	Quantity in Slot	Storage Slot	Distance
2	0	1	30	15	1	34
3	0	1	30	15	2	2
4						



	A	B	C
1	ItemNo.	Quantity in Slot	Storage Slot
2	1	40	1
3	1	15	2
4	2	200	3

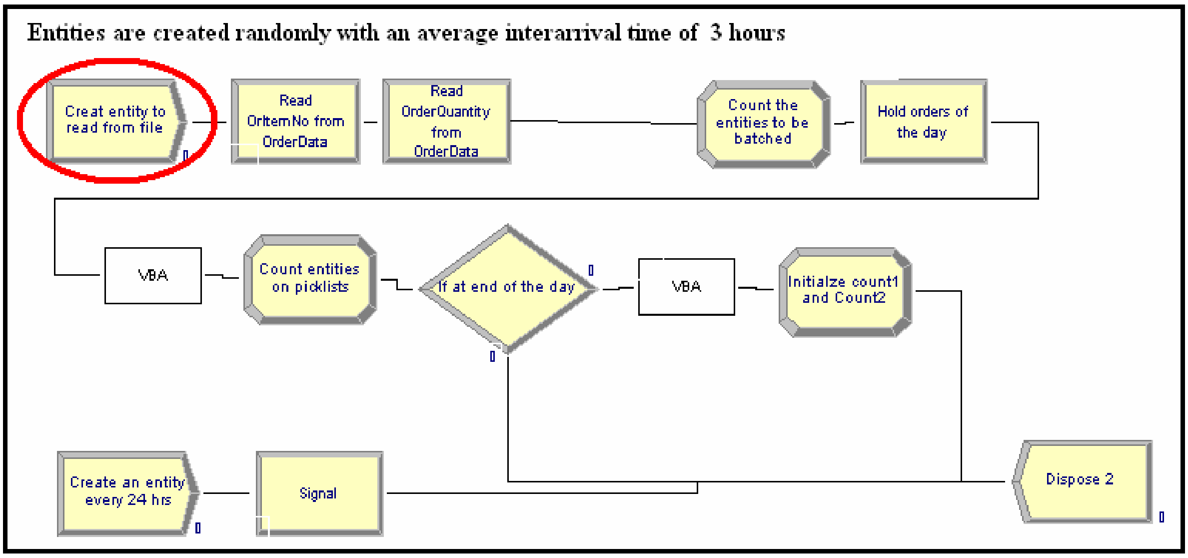
Step 6s:

Dispose entity from system



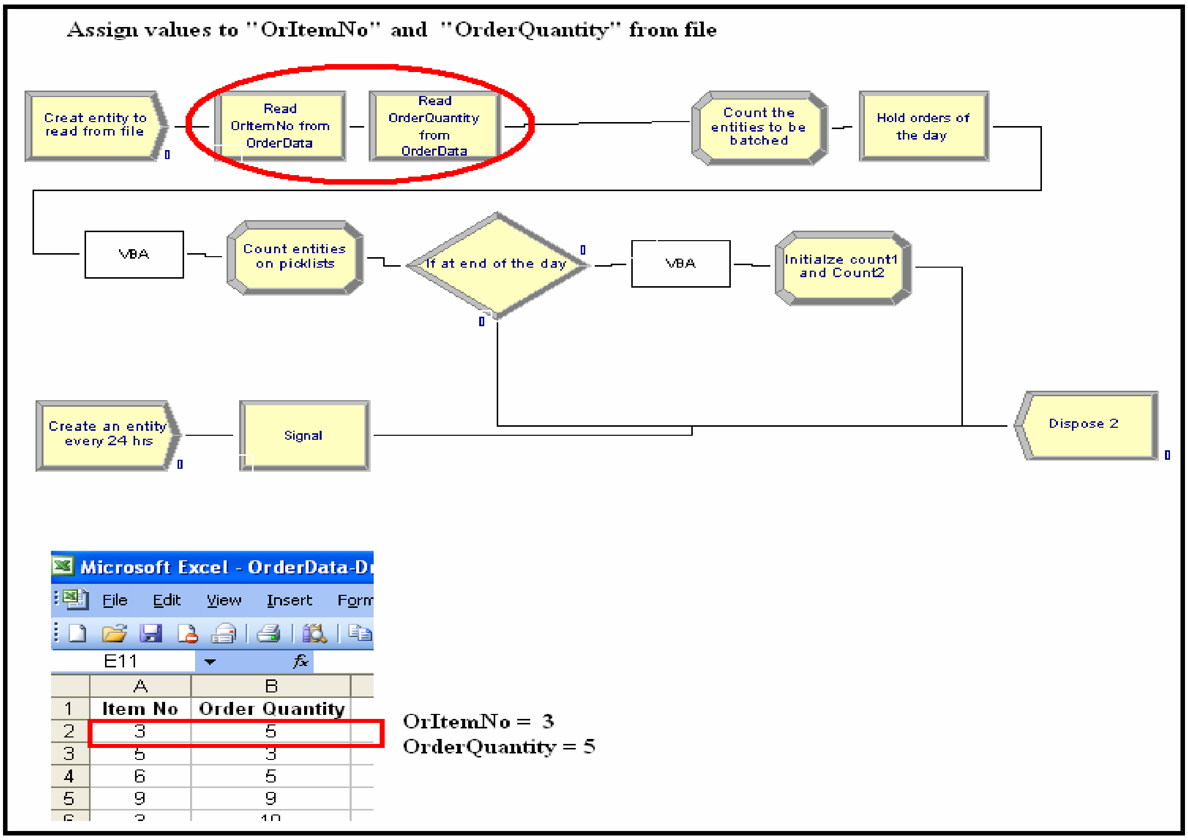
Simulation Manual for the Picking Sub-model

Step 1p:



Entities are created randomly with an average interarrival time of 3 hours

Step 2p:

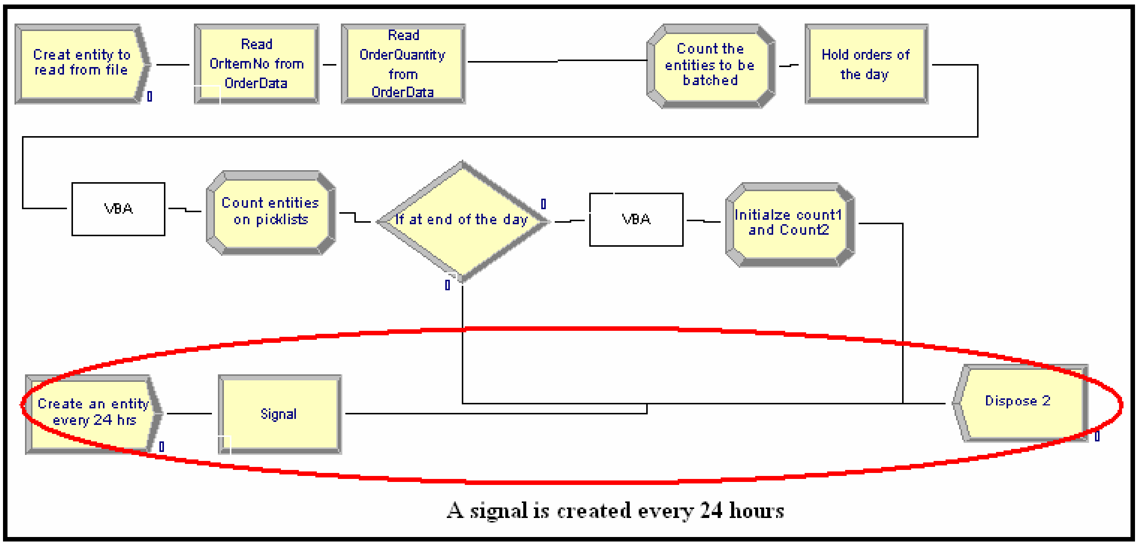


Assign values to "OrItemNo" and "OrderQuantity" from file

	A	B
1	Item No	Order Quantity
2	3	5
3	5	3
4	6	5
5	9	9

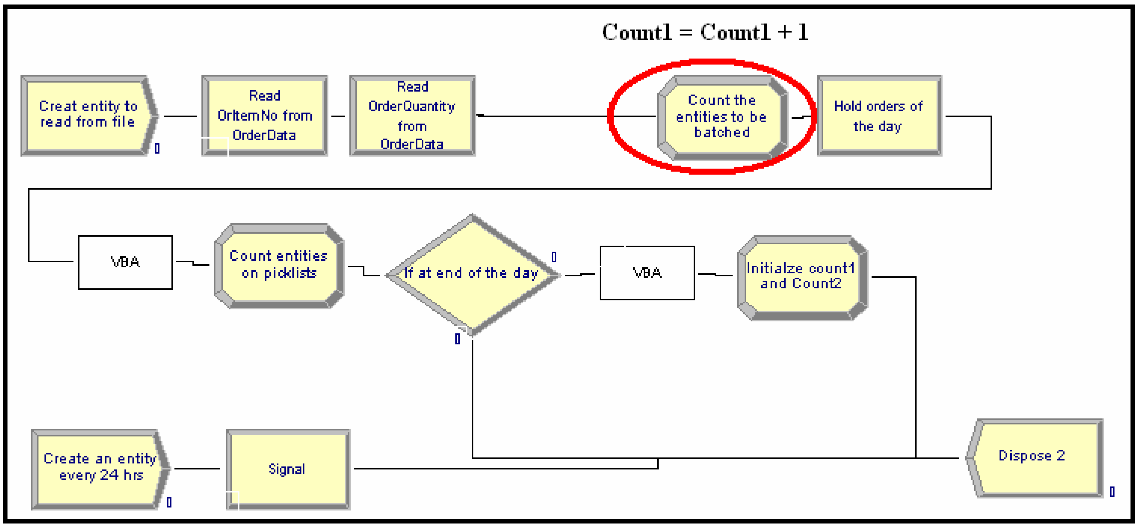
OrItemNo = 3
OrderQuantity = 5

Step 3p:



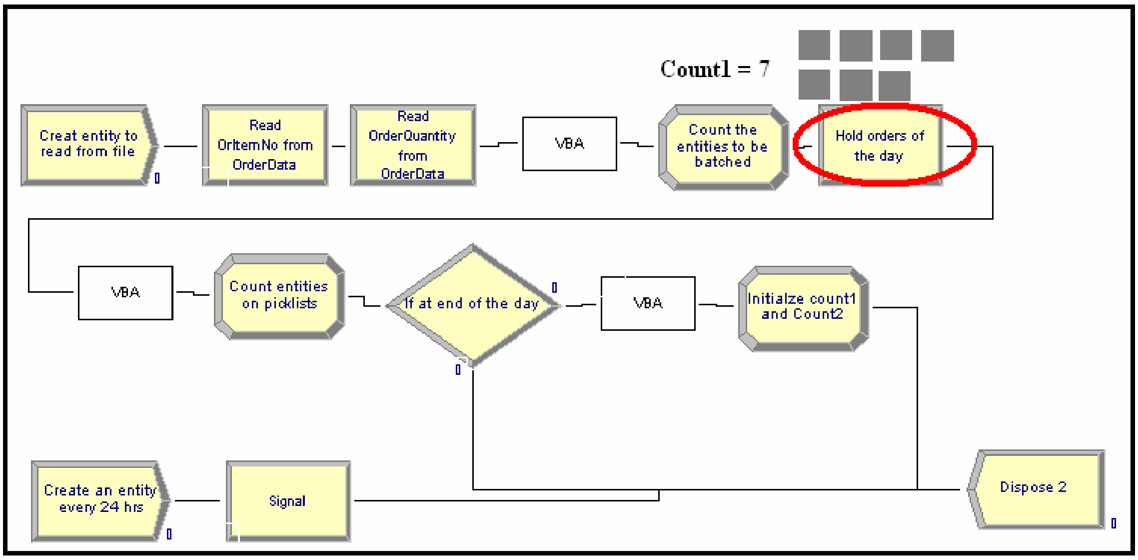
A signal is created every 24 hours

Step 4p:

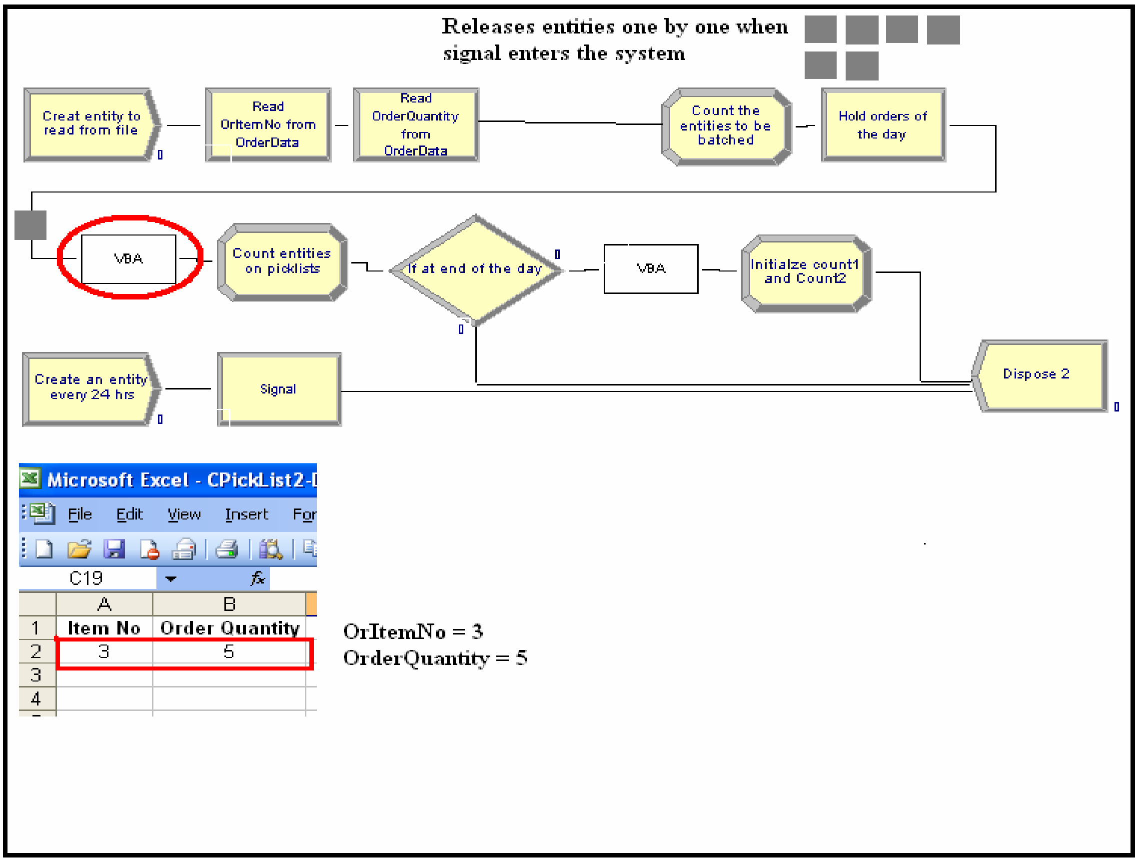


Count1 = Count1 + 1

Step 5p:



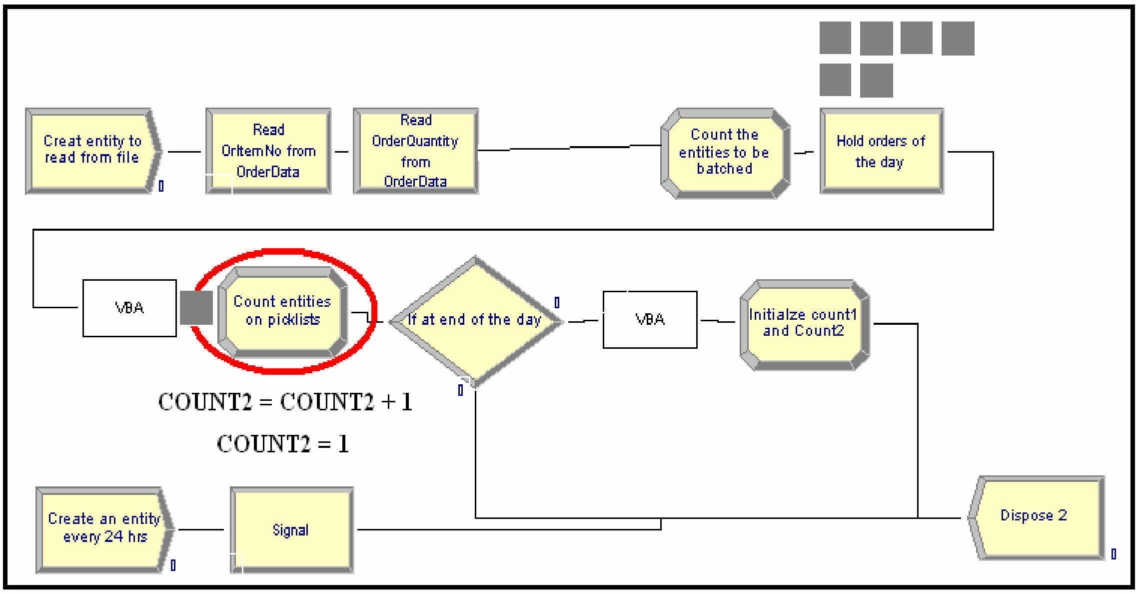
Step 6p:



	A	B
1	Item No	Order Quantity
2	3	5
3		
4		

OrItemNo = 3
OrderQuantity = 5

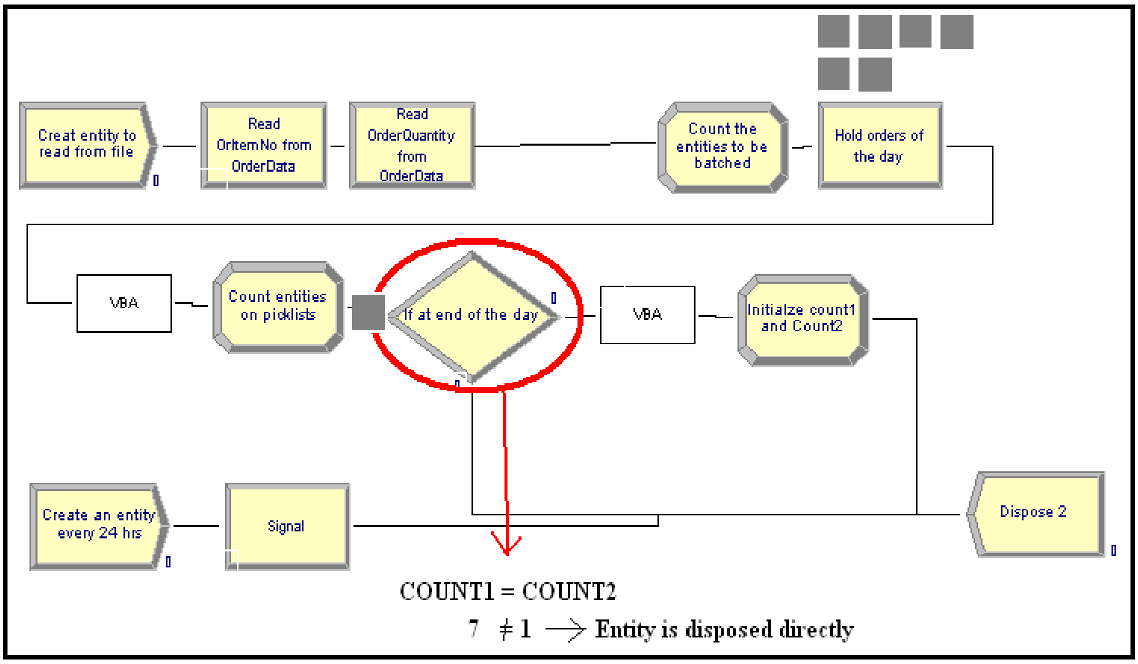
Step 7p:



COUNT2 = COUNT2 + 1

COUNT2 = 1

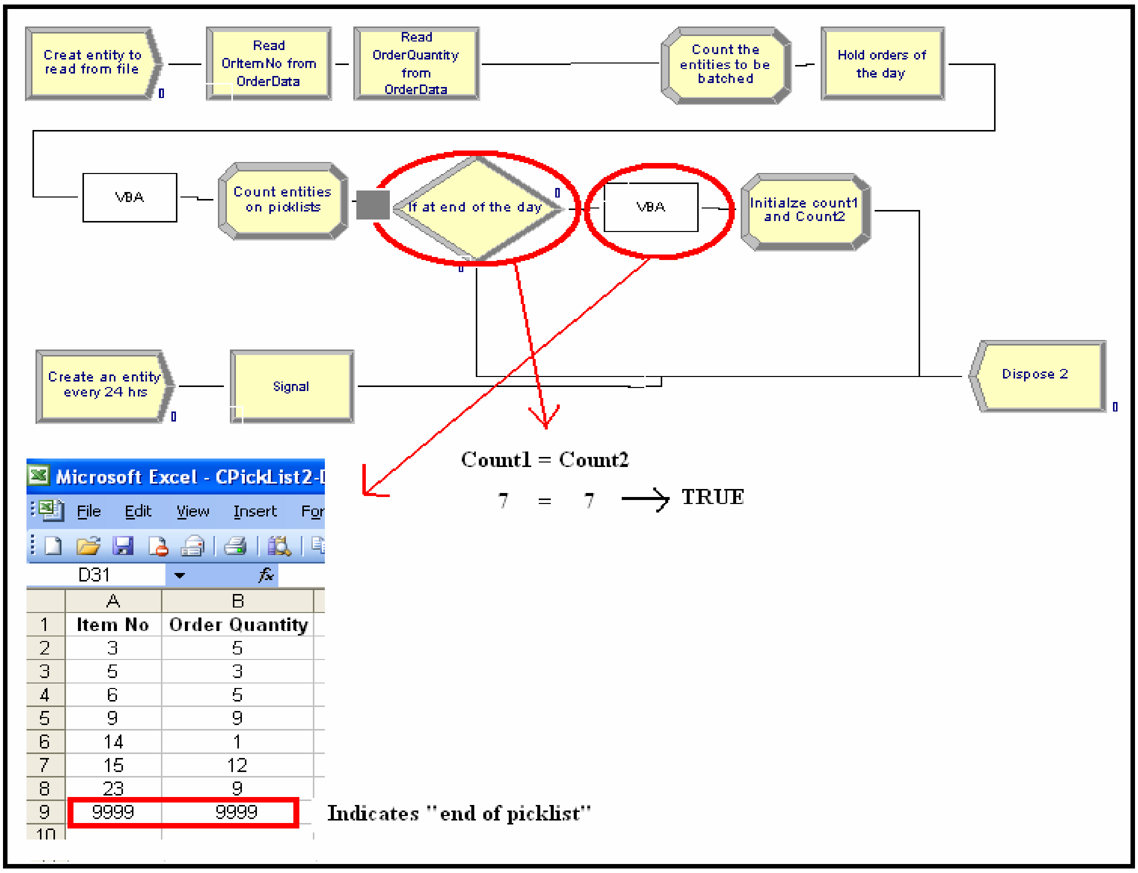
Step 8p:



COUNT1 = COUNT2

7 ≠ 1 → Entity is disposed directly

Step 9p:



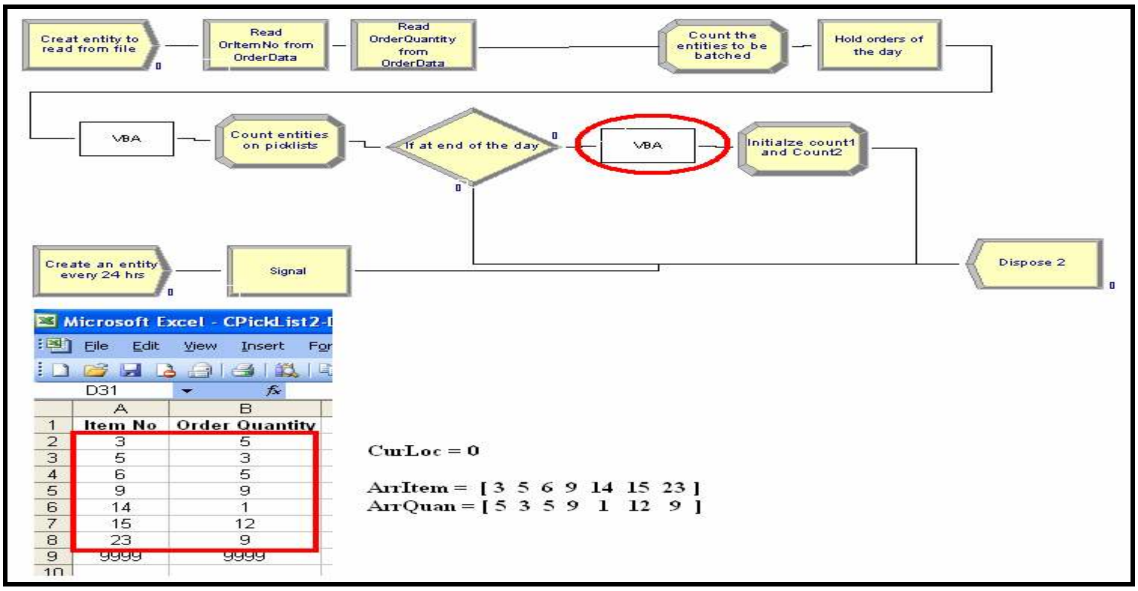
	A	B
1	Item No	Order Quantity
2	3	5
3	5	3
4	6	5
5	9	9
6	14	1
7	15	12
8	23	9
9	9999	9999
10		

Count1 = Count2

7 = 7 ⟶ TRUE

Indicates "end of picklist"

Step 10p:

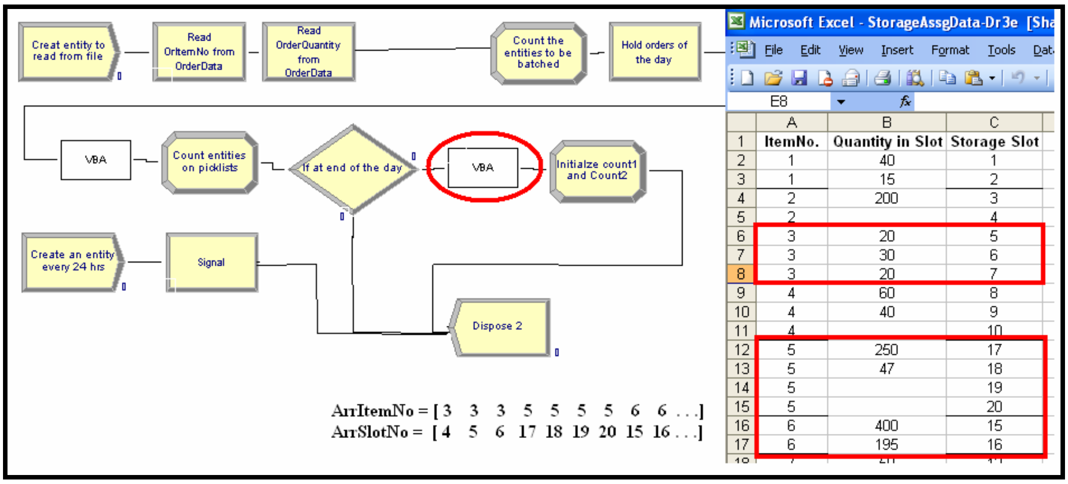


	A	B
1	Item No	Order Quantity
2	3	5
3	5	3
4	6	5
5	9	9
6	14	1
7	15	12
8	23	9
9	9999	9999
10		

CurLoc = 0

ArrItem = [3 5 6 9 14 15 23]
ArrQuan = [5 3 5 9 1 12 9]

Step 11p:

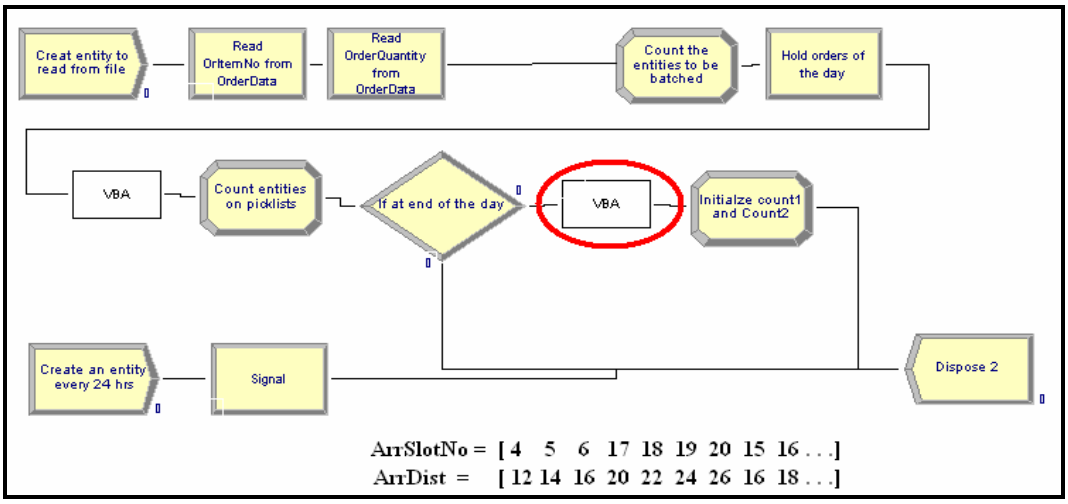


Microsoft Excel - StorageAssgData-Dr3e [Sha

	A	B	C
1	ItemNo.	Quantity in Slot	Storage Slot
2	1	40	1
3	1	15	2
4	2	200	3
5	2		4
6	3	20	5
7	3	30	6
8	3	20	7
9	4	60	8
10	4	40	9
11	4		10
12	5	250	17
13	5	47	18
14	5		19
15	5		20
16	6	400	15
17	6	195	16

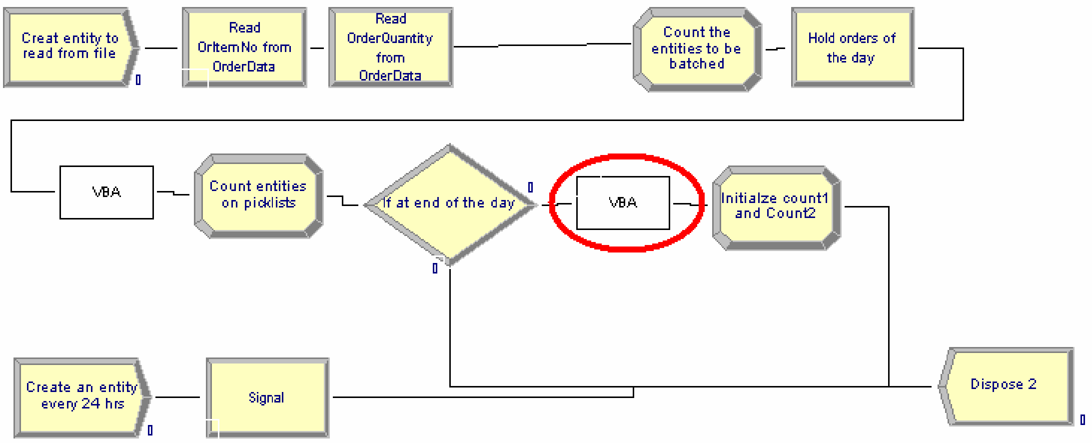
ArrItemNo = [3 3 3 5 5 5 5 6 6 . . .]
ArrSlotNo = [4 5 6 17 18 19 20 15 16 . . .]

Step 12p:



ArrSlotNo = [4 5 6 17 18 19 20 15 16 . . .]
ArrDist = [12 14 16 20 22 24 26 16 18 . . .]

Step 13p:

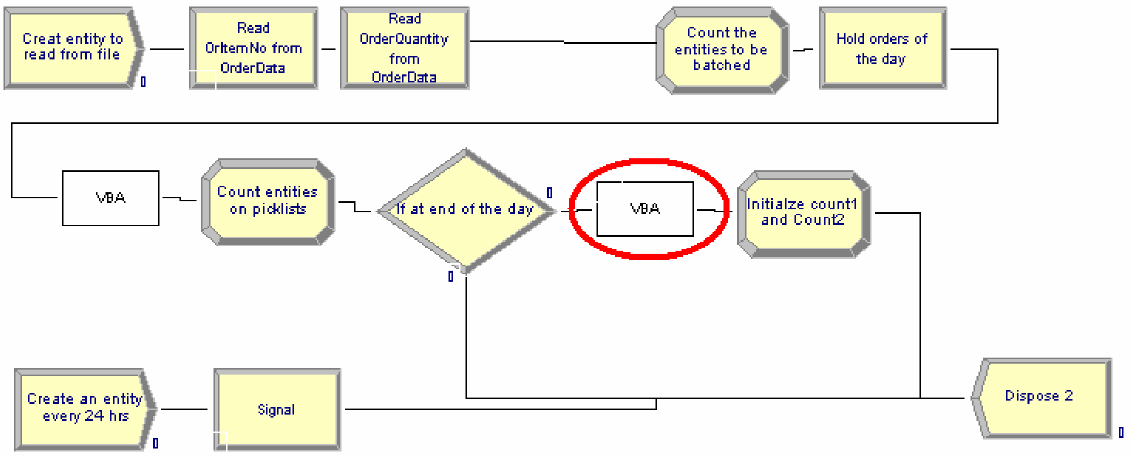


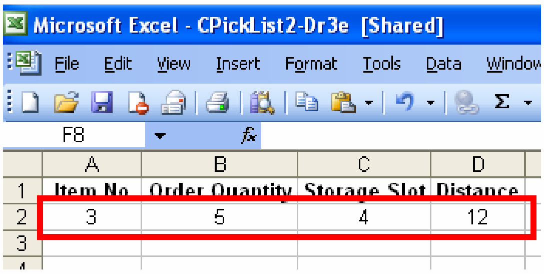
ArrItemNo = [3 3 3 5 5 5 5 6 6 . . .]
ArrSlotNo = [4 5 6 17 18 19 20 15 16 . . .]
ArrDist = [12 14 16 20 22 24 26 16 18 . . .]

→

Sorted arrays:
ArrDist = [12 14 16 16 18 20 22 24 26 . . .]
ArrSlotNo = [4 5 6 15 16 17 18 19 20 . . .]
ArrItemNo= [3 3 3 6 6 5 5 5 5 . . .]

Step 14p:



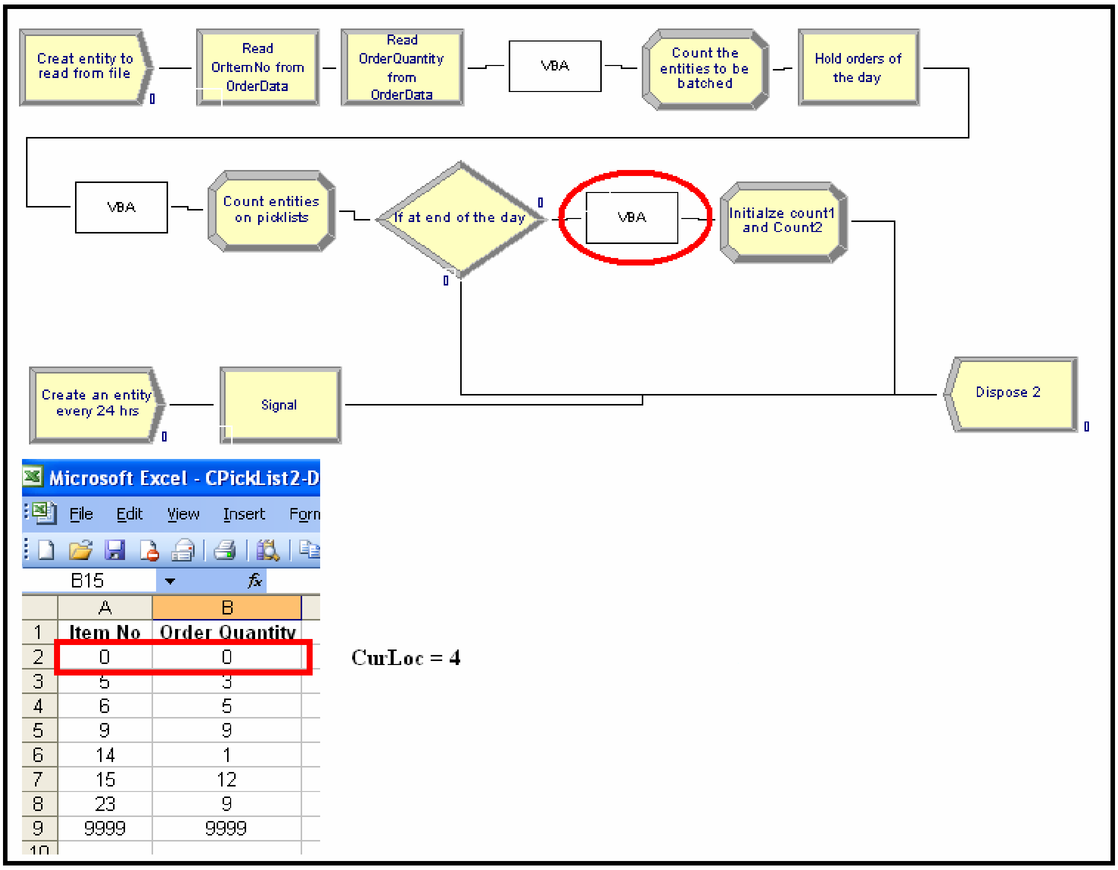


	A	B	C	D
1	Item No	Order Quantity	Storage Slot	Distance
2	3	5	4	12
3				

ArrDist = [12 14 16 16 18 20 22 24 26 . . .]
ArrSlotNo = [4 5 6 15 16 17 18 19 20 . . .]
ArrItemNo= [3 3 3 6 6 5 5 5 5 . . .]

ArrItem = [3 5 6 9 14 15 23]
ArrQuan = [5 3 5 9 1 12 9]

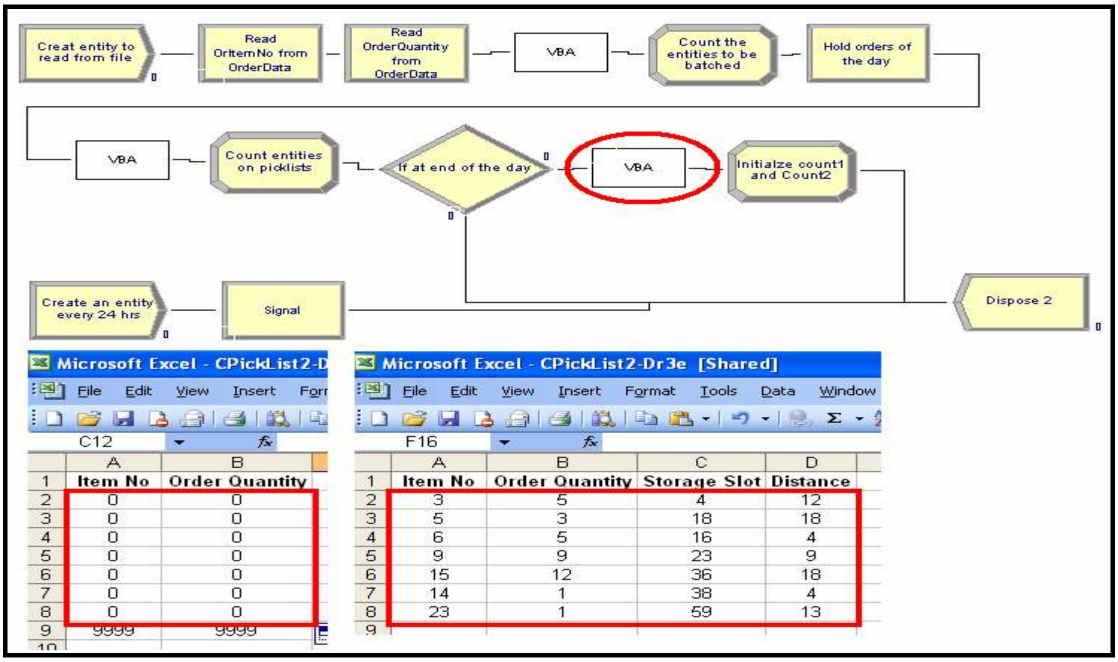
Step 15p:



	A	B
1	Item No	Order Quantity
2	0	0
3	5	3
4	6	5
5	9	9
6	14	1
7	15	12
8	23	9
9	9999	9999

CurLoc = 4

Step 16p:



	A	B
1	Item No	Order Quantity
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	9999	9999

	A	B	C	D
1	Item No	Order Quantity	Storage Slot	Distance
2	3	5	4	12
3	5	3	18	18
4	6	5	16	4
5	9	9	23	9
6	15	12	36	18
7	14	1	38	4
8	23	1	59	13