# Operational-Level Rule-Based Heuristics for Receiving Logistics

by

Yeiram Martínez Meléndez

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

MASTER OF SCIENCE

in

INDUSTRIAL ENGINEERING
(Management Systems)

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS

May 2017

Approved by:

________________________________ ___________
Sonia M. Bartolomei Suárez, Ph.D. Date
Member, Graduate Committee

________________________________ ___________
Héctor J. Carlo Colón, Ph.D. Date
President, Graduate Committee

________________________________ ___________
Nazario D. Ramírez Beltrán, Ph.D. Date
Member, Graduate Committee

________________________________ ___________
Viviana I. Cesaní Vázquez, Ph.D. Date
Department Director

________________________________ ___________
Edgar Soto Rodríguez, DBA Date
Graduate Studies Office Representative

We ourselves must walk the path.

Buddha

## Abstract

This study is concerned with the *receiving logistics problem* (RLP) which consists of optimizing inbound operations at distribution centers (DCs), warehouses, and cross-docks (XDs) with staging areas. The objective of the RLP is to minimize the makespan required to move all unit loads from trailers to flow racks, and from flow racks to their respective storage (or delivery) locations. It is assumed that a set of inbound trailers with known composition have been assigned and sequenced to inbound dock doors. The following three inbound logistics decisions are simultaneously considered: *i*) unloaders' assignment and scheduling, *ii*) loads-to-flow rack assignment, and *iii*) assignment and haulers' scheduling. In this study we describe the relationship between the problem of minimizing makespan and balancing the unloader-hauler workload. A linear mixed integer formulation of the RLP is presented, and the problem is shown to be *NP-hard*. Hence, four rule-based heuristics are proposed and evaluated by replicating a variety of unloading scenarios and comparing results. It is concluded that a simple heuristic rule that does not require additional technology outperforms the other rules in terms of solution quality.

This study also presents a proof of concept where a single Microsoft Kinect sensor is used for automated monitoring of a dock door in real-time. The proposed system will automatically and in real-time: *i)* detect when an object breaches the dock door perimeter, and its corresponding speed and direction, *ii)* count the number of pallets loaded/unloaded to/from a trailer, *iii)* record the loading/unloading time of each load, and *iv)* reconstruct an image of every loading/unloading trip at a dock door, irrespectively of the material handling travel speed and direction. Particular emphasis is given to discussing how to extend the proposed concept by using multiple Kinect sensors, the technological challenges for implementation, and the expected benefits of a real-time dock door monitoring system.

# Resumen

Este trabajo se enfoca en resolver el problema de logística de recepción (RLP) que se enfoca en optimizar las operaciones de recepción de mercancía en centros de distribución, almacenes o "cross-docks" con almacenaje temporero. El objetivo del RLP es minimizar el tiempo total de descargar las unidades de los vagones a un estante de gravedad y posteriormente a su lugar de almacenaje (o entrega). Se presume que un conjunto de vagones cuyo contenido es conocido ya ha sido asignado y secuenciado a las puertas de descarga. Los siguientes tres problemas de logística de recepción son considerados simultáneamente: *i)* asignar y coordinar el personal a descargar vagones ("unloaders"), *ii)* asignar un estante de gravedad ("flow rack") a cada paleta que se descarga, y *iii)* asignar y coordinar el personal que transporta las paletas del estante hasta el lugar de almacenaje. En este estudio se describe la relación entre el problema de minimizar el tiempo total de la operación y balancear las cargas de trabajo para ambos recursos que desmontan camiones ("unloaders") y transportan las paletas al área de almacenaje ("haulers"). Se presenta una formulación matemática lineal entera-mixta para el RLP y se demuestra que el problema tiene complejidad "NP-difícil". Por tanto, se proponen cuatro heurísticos basados en reglas para resolver el problema de manera eficiente y se evalúan y comparan estas reglas en una variedad de escenarios. Se concluye que una regla simple que no requiere tecnología adicional supera las otras reglas en términos de calidad de solución.

Además, se presenta un concepto en donde se utiliza un sensor de profundidad "Microsoft Kinect" como mecanismo de recolección de datos en el área de descargar camiones. El sistema propuesto es capaz de hacer lo siguiente automáticamente y tiempo real: *i)* detectar cuando una paleta entra o sale de un camión, documentando su velocidad y dirección, *ii)* contar el número de paletas que entran o salen en/de un camión, *iii)* documentar el tiempo que toma montar o descargar una paleta de/en un camión, y *iv)* reconstruir una imagen de lo que entra o sale en/de un camión, irrespectivamente de la velocidad y dirección del movimiento. El énfasis principal está en discutir cómo establecer un sistema de monitoreo con múltiples sensores para las operaciones de recepción de mercancía, los retos tecnológicos para su implementación y los beneficios de un sistema de monitoreo en tiempo real.

# Table of Contents

# List of Figures

# List of Tables

## 1. Introduction to Receiving Logistics

A trend in large supply chain networks is to use storage facilities, such as distribution centers (DCs) and warehouses (WHs), as redistribution points to fulfill orders from multiple demand points. The basic functions of DCs are to receive, store, and ship a large variety of items. These large scale DCs are commonly used in practice by large retail companies such as Amazon and Walmart in order to respond quickly to demand. The operations at DCs are extensive and require different equipment and coordination between operation areas. Gu *et al.* (2006) divides the physical space in storage facilities into *receiving*, *storage*, *picking*, and *shipping* areas as shown in Figure 1. The two main differences between DCs and WHs are: (1) DCs are designed to serve more customer orders so they are designed for faster storage and retrieval at different storage keeping unit (SKU) levels (e.g., pallets, cases, eaches); and (2) DCs tend to include some product transformation (or value-added) operations for postponement. Henceforth, for convenience, this thesis will refer to storage facilities as DCs.

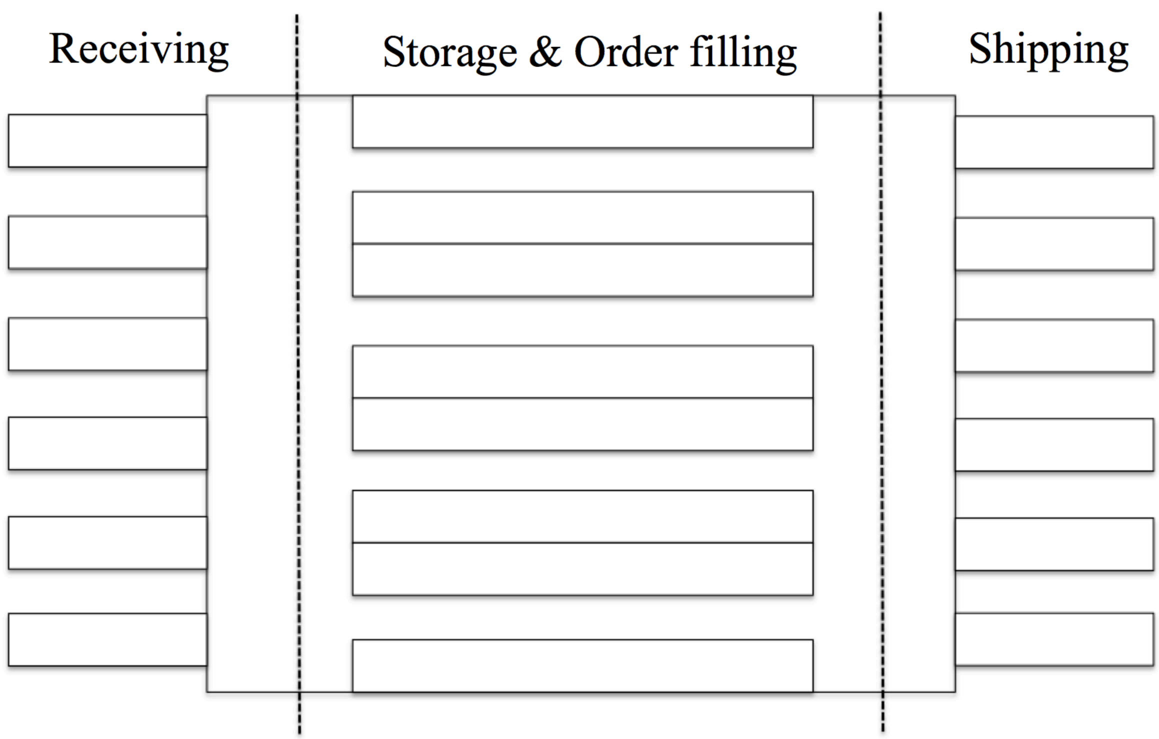



*Figure 1. Distribution Center General Layout*

Furthermore, DCs are typically organized into the following areas:

- **Systems** electronically receives customer orders from all demand points, consolidates them into internal retrieval jobs (or orders) that are passed on to the transportation department to generate trailer loads.
- **Transportation** decides which orders will be combined and which (outbound) trailer will deliver them. This information is transferred to the shipping department, where order-fillers build the pallets from multiple case-picks throughout the DC. Transportation also coordinates trailer yard logistics; inbound freight, trailers to be received and unloaded, and outbound freight (*i.e.*, trailers headed to the demand points).
- **Receiving**, usually at the operational level, handles trailer to door assignments and unloading scheduling. It is convenient that receiving department handles these operations due to their prior knowledge on unloading times and difficulty of loads. On occasions unloading times are extended because pallets require additional operations such as breakdowns, were one pallet is broken down into multiple pallets for storage purposes. Overall, this department handles unloading operations, pallet identification, sorting of inbound loads, and put-away (*i.e.*, storing inbound loads as inventory).
- **Shipping** handles order-picking operations, in which personnel travels through the storage racks picking internal orders. These internal orders are sorted (if necessary) and pallets are built to fulfill the customer orders. Afterwards, pallets are wrapped in plastic (*i.e.*, shrink-wrapped) and loaded into their assigned outbound trailer according to the schedule set by the transportation department.

## 1.1 Thesis Objectives

This study focuses on the receiving operations of DCs, also known as inbound intralogistics (Martínez and Carlo, 2016) by: (1) proposing a low-cost concept to gather real-time performance data for receiving operations, and (2) optimizing the operational-level receiving logistics. The main objective of this project is the development of a framework used to efficiently solve operational-level problems in receiving logistics. This includes to simultaneously solve the following problems: *inbound trailer-to-door assignments*, *inbound doors scheduling*, *unloaders scheduling*, *load-to-temporary storage assignment*, and *haulers scheduling*. We refer to the problem addressed in this thesis as the *receiving logistics problem* (RLP). The primary and secondary objectives of this thesis are:

**Primary objectives**

- Formulate the operational-level optimization problem for receiving logistics.
- Establish the mathematical complexity of the problem.
- Propose several ruled-based heuristics to efficiently solve the problem.

**Secondary objectives**

- Describe how a Microsoft Kinect-based dock door monitoring system can be integrated with the model as a data acquisition tool for receiving logistics.

Chapter 2 presents the framework for the dock door monitoring system using a Kinect depth sensor, the initial research that led to the development of the *receiving logistics problem* (RLP). After demonstrating how to obtain operational-level data automatically and in real-time, we focus on solving the RLP assuming this information is readily available for the decision makers. Chapter 3 and 4 presents the problem description and literature review, respectively, for the RLP. Chapter 5 presents the mathematical formulation and complexity, followed by the rule-based heuristics in Chapter 6. Chapter 7 and 8 present the experimental experiments and simulation built to replicate the unloading scenarios. Finally, Chapters 9 and 10 present the experimental results, conclusions, and recommendations for future works.

# 2. Implementation Framework: Real-Time Dock Door Monitoring System

This Chapter presents a proof of concept where a single Microsoft Kinect sensor is used for automated monitoring of a dock door in real-time (Carlo, Martínez and Pomales-García, 2014). The proposed system will automatically and in real-time: (1) detect when an object breaches the dock door perimeter and its corresponding speed and direction, (2) count the number of pallets loaded/unloaded to/from a trailer, (3) record the loading/unloading time of each load, and (4) reconstruct an image of every loading/unloading trip at a dock door, irrespectively of the material handling travel speed and direction. Particular emphasis is given to discussing how to extend the proposed concept by using multiple Kinect sensors, the technological challenges for implementation, and the expected benefits of a real-time dock door monitoring system. Such framework could work as a data gathering mechanism to obtain the operational-level parameter data required in the methods in Chapters 4 and 5.

## 2.1 Framework Overview

An automated dock door monitoring system could provide valuable real-time data on receiving and shipping dock operations. The data collected may be used to help dock managers to: remotely visualize the docks' status, monitor current and projected productivity standards and performance metrics, and manage dock operations (including supervising employees, job prioritization, trailer/truck security, and traffic safety enforcement). This work presents a *proof of concept* for a dock door monitoring system using a single Kinect sensor. The proposed system will automatically and in real-time: detect when the dock door perimeter has been breached and at what speed and direction, count the number of pallets loaded/unloaded to/from a trailer, record the loading/unloading time of each load, and reconstruct an image of every loading/unloading trip at a dock door, irrespectively of the material handling travel speed and direction.

## 2.2 Overview of Sensor Technologies in Warehouse

This section briefly describes and evaluates the most common sensor technologies used in warehouses in terms of their potential to be used as a dock door monitoring system. The following technologies are considered: photoelectric sensors, radio frequency identification technology, high definition cameras, and 3-D range sensors.

*Photoelectric sensors* send a beam of light from an emitter to a receiver. Objects are detected when the (light) communication between the emitter and receiver is interrupted. These sensors are commonly used to activate automated equipment such as pallet wrapping machines and AS/RSs, and to activate automated lighting and doors. Unfortunately, although photoelectric sensors may detect objects, they cannot recognize them. Hence, this sensor would have to be combined with other sensors if used as part of a dock door monitoring system.

*Radio frequency identification* (RFID) technology uses radio-frequency electromagnetic fields to transfer data. RFID technology includes an RFID tag and a RFID reader (Landt, 2005). In 2004, the Gillette Company filed a patent that presented the framework for an RFID system with capability of tracking products at different points in a distribution chain (Sullivan and Jamshed, 2004). The main challenge with using RFID technology as part of a dock door monitoring system is that RFID tags need to be placed on the loads in order to be recognized. Hence, the technology is not feasible for unloading operations (as loads are identified after being unloaded) and the system can be easily manipulated by removing the RFID tags.

*High definition cameras* are capable of taking high precision red-green-blue (RGB) images commonly used for processes that require fast color imaging tasks such as packaging and color printing. This technology can be used to obtain and manipulate RGB data for applications such as object tracking but it does not provide the required data to reconstruct the surface of an object, obtain time samples, and measure speed (Weng *et al*., 2006) (Comaniciu *et al*., 2003).

*Range imaging technology* has been around since 1970s (Blais, 2004) This technology, also known as *range* or *depth sensor technology*, include single point and laser scanners, slit scanner, patter projection, and time-of-flight systems. Since then, low cost commercial systems have been developed with the capability of providing full 3-D images that can be used to reconstruct the surface object based on depth data. The main limitation of these sensors includes poor readings due to background light, shadows and reflections, and reading interference. Ideally, a real-time dock door monitoring system would include multiple sensor technologies. The next sub-Section describes the Microsoft Kinect sensor, which incorporates a depth sensor, RGB camera, and microphone.

### 2.3 Microsoft Kinect Sensor

The Kinect sensor is a low-cost depth sensor that was designed as an add-on peripheral for the Xbox in 2010. Microsoft introduced a second generation Kinect exclusively for developers in 2012, named Kinect for Windows. The Kinect sensor has the capability of providing depth and color data for each pixel of an image. Shown on Figure 2, it consists of an infrared (IR) emitter, which projects an infrared pattern onto objects in the sensors field of view, and an infrared camera that detects the changes in the IR pattern. It also contains a RGB camera, a 4-microphone array, and a tilt motor to adjust viewing angle. The sensor has an angular field of view of 57° horizontally / 43° vertically, with the capability of tilting up to 27° up and down. This sensor is capable of providing depth and color images at a maximum of 30 frames per second, at its default resolution is 640×480 pixels.

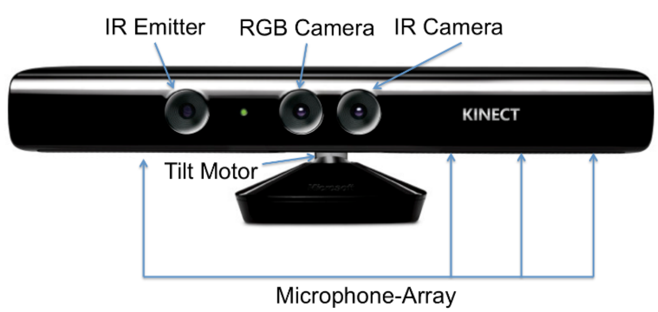



*Figure 2. Kinect for Windows*

The range for the depth sensor is 0.4–4.0 m (1.3–13.1 ft.). The microphone array features four microphone capsules, with each channel processing 16-bit audio at a sampling rate of 16 kHz. In summary, the Kinect is capable of capturing 3D motion, facial recognition, voice recognition, and acoustic source location, while suppressing ambient light and noise. For further information, operating capabilities and comparisons of the Kinect sensor the reader is referred to Khoshelham

(2010), Macknojia *et al.* (2012), Smisek *et al.* (2012), Dutta (2012), and Livingston *et al.* (2012). The Kinect sensor has been successfully used to develop a real-time ergonomic monitoring system using the skeletal tracking capabilities of the sensor (Martin *et al.,* 2012). It has also been used in studies for facial recognition (Li *et al.,* 2013) and real-time object tracking (Nakamura, 2011). The Kinect was also used for a warehouse related study focused on order picking (Li *et al*., 2012), where the authors introduced a monitoring approach based on RGB and depth data to recognize and monitor different box-shaped picked items.

We believe the Kinect for Windows is an appropriate technology for real-time dock door monitoring and for gathering operational-level data that can be used to optimize the receiving logistics operations. In the following sub-Section, we present a proof of concept of how the Kinect sensor may be used as part of a real-time dock door monitoring system.

### 2.4 Proof of Concept of Dock Door Monitoring System

A small-scale controlled-setting environment was designed as a proof of concept for the real-time dock door monitoring system. The setting, depicted in Figure 3, includes a Kinect sensor mounted on a wall overlooking a tabletop, which represents a dock door. Over the tabletop in Figure 3 one can also note that the Kinect sensor is connected to a laptop computer that controls the sensor. Figure 4 shows a top view of a region on the tabletop that represents the inside of the dock.


*Figure 3. Camera Setup*

The objective is to demonstrate that the Kinect sensor may be used as a low-cost real-time dock door monitoring system. The proposed dock door monitoring system will use the Kinect depth sensor to monitor changes in depth while objects pass below the sensor, just as if it was placed over a dock door for monitoring. This section will discuss the setup and workspace used, followed by an overview of the application developed to measure speed, time, and occurrences of objects passing below the sensor.

The sensor was placed vertically on a wall, facing down approximately 1.0 meter above a worktable, as seen on Figure 3. A portion of the table was covered in brown paper to avoid the reflection of light on a glossy surface as it may cause invalid depth readings. (In our prototype, the close proximity of the sensor, combined with a shiny surface of the table, and strong indoor illumination created reflectance problems. However, we understand that reflectance should not be a major issue in real-life docks.)

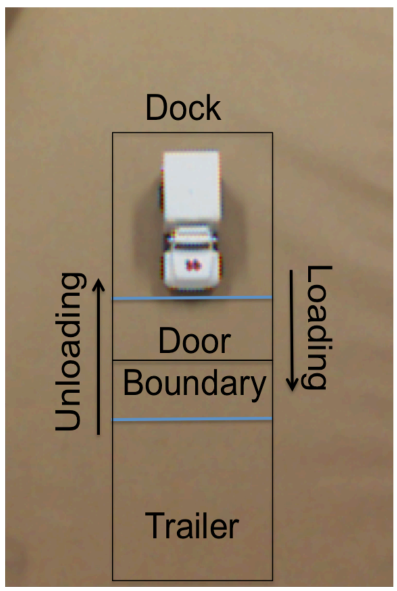



*Figure 4. Setup Top View*

An application to control the Kinect sensor was written in Visual Studio using the *C#*. Figure 4*,* shows an example where a toy ambulance is used to represent a material handling equipment about to enter a trailer. We will observe depth changes in three over-imposed lines in order to detect breaches (*i.e.*, an object approaching the dock door). *Line 1* (blue), located just below the truck in Figure 4, is used to recognize that a load is approaching toward the trailer. *Line 2* (black), labeled "door boundary", marks the division between the dock and trailer. Although *Line 2* is located inside the dock, any load that crosses this line is considered to have left the dock. *Line 3* (blue) is located just below *Line 2* and is the inbound counterpart of *Line 1*. The approaching speed of an object is computed based on the time and distance between *Lines 1* or *3*, and *Line 2*.

In a DC loading scenario, a breach in *Line 1* marks the beginning of a loading movement (*i.e.*, an object is being loaded to the trailer). The object moves toward the trailer until *Line 2* is breached; here, the time difference between both breaches is used to calculate entry speed, and a truck-entry counter is increased. The same concept applies for an unloading scenario; the only difference is that we evaluate an object exiting the trailer. In unloading trips *Line 3* will detect a breach instead of *Line 1* and the breaches will be added to a second counter used to log exit occurrences. The application developed also records the total time a material handler spends inside the trailer (i.e., the time it takes to either to unload or load a single pallet) and the total time incurred to load or unload a trailer.

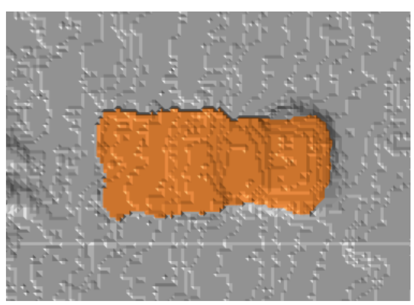
An application to control the Kinect sensor was written in Visual Studio using the *C#*. Figure 4, shows an example where a toy ambulance is used to represent a material handling equipment about to enter a trailer. We will observe depth changes in three over-imposed lines in order to detect breaches (*i.e.*, an object approaching the dock door). *Line 1* (blue), located just below the truck in Figure 4, is used to recognize that a load is approaching toward the trailer. *Line 2* (black), labeled "door boundary", marks the division between the dock and trailer. Although *Line 2* is located inside the dock, any load that crosses this line is considered to have left the dock. *Line 3* (blue) is located just below *Line 2* and is the inbound counterpart of *Line 1*. The approaching speed of an object is computed based on the time and distance between *Lines 1* or *3*, and *Line 2*.

In a DC loading scenario, a breach in *Line 1* marks the beginning of a loading movement (*i.e.*, an object is being loaded to the trailer). The object moves toward the trailer until *Line 2* is breached; here, the time difference between both breaches is used to calculate entry speed, and a truck-entry counter is increased. The same concept applies for an unloading scenario; the only difference is that we evaluate an object exiting the trailer. In unloading trips *Line 3* will detect a breach instead of *Line 1* and the breaches will be added to a second counter used to log exit occurrences.

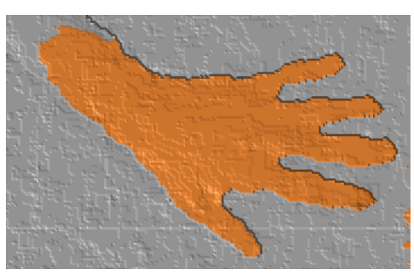
The application developed also records the total time a material handler spends inside the trailer (*i.e.*, the time it takes to either to unload or load a single pallet) and the total time incurred to load or unload a trailer.

Besides recording travel speeds and entry/exit count and timestamps, the application also records two additional sets of information for every breach: a color image screenshot and a depth data surface plot reconstruction. Figure 5a, shows a toy ambulance used to test the programs' ability to export depth data in real-time for a moving object. As soon as the toy ambulance breached *Line 2* (*i.e.*, there was a depth change in the line) the depth data from the sensor was used to automatically create the surface plot in Figure 5b. Note that Figures 5a and 5b are both taken for a moving object and generated automatically.


(a) – Color image



(b) – Depth surface plot



(c) – Hand depth surface plot

*Figure 5. Depth Plots Using Kinect Sensor*

In a real life scenario, a color screenshot with date and time could be used for accountability of excess speed events, it could also be used to find confirm pallets have been received, and to aid in tracking of missing or unlabeled pallets. Furthermore, it could be used to document who had access to a particular trailer, which is vital for security sensitive trailers. A depth data-based surface plot would be used to recognize, document, and differentiate objects that interact with the trailer: either a person or a material handling equipment (with and without a load). The object identification will depend on the dimensions of the object, which may be obtained by analyzing the depth data.

During this exercise, while plotting multiple objects passing below the sensor, we noticed that the borders of the objects are not as accurate or smooth as a color image. Figure 5c, presents a surface plot of a hand, which has a different geometry than the previously tested. It is observed that the separation and width of some fingers appears to be distortional. Although in this experiment we are evaluating smaller objects (with smaller details), than required for real-life docks, we must consider improving the accuracy of the reconstructed image from the depth sensor by using methods such as the ones described in (Vijayanagar et al., 2012).

Figure 6 presents a screenshot of the graphic user interface (GUI) developed as part of the *C#* application to control the Kinect sensor. The two images on the top of Figure 6 are real-time depth (left) and color (right) frames from the Kinect sensor. The numbers of the lower-left portion of Figure 6 provide counters, time and speed data from the last breach, as well as well as depth measurements for the three (over-imposed) lines. On the middle-bottom of Figure 6 there are controls for the sensor tilt and to manually capture and record images such as the one presented on the lower-right of the figure. In the developed application the images are taken automatically when a breach occurs. These images are stored in the computer.

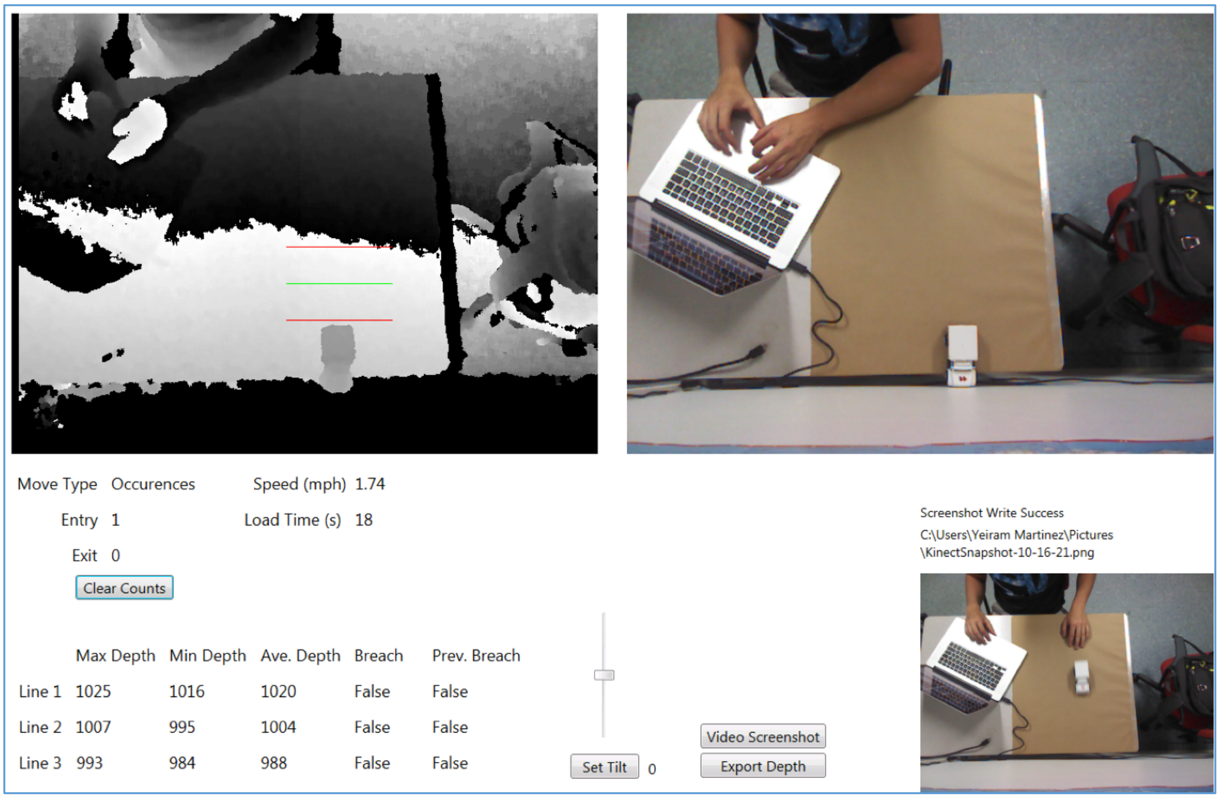



*Figure 6. GUI Screenshot*

The small-scale concept shows that the Kinect sensor may be used as part of a real-time dock door monitoring system that automatically identifies perimeter breaches, logs timestamps, monitors entry and exit speed, and exports a color image screenshot together with a depth data based surface plot. Since the Kinect sensor is controlled by a computer, the obtained data may be stored and analyzed in the computer or sent to a database so the information is shared.

### 2.5 Multi-Kinect Dock Door Monitoring System

In this section we discuss the challenges involved in extending the single-Kinect concept to a multi-Kinect environment. Given the sensor range of the Kinect sensor, we envision using one Kinect sensor per dock door. These sensors could either be connected to individual computers or connected in parallel to the same computer. If Kinects are connected individually, then it would be equivalent to having multiple single-door monitoring systems. Clearly, this option would require a larger investment (in computers) and would slightly increase electricity cost and resources required to store data from multiple sources. On the other hand, if multiple Kinects are to be connected to the same computer, then USB bandwidth and power requirements may become an issue. Irrespectively of how the sensors are connected to the computers, a central computer or database would be required to unify the information from all the sensors in order to conveniently present the information to management.

Another aspect to consider when using multiple sensors is the interference problem caused by the overlapping of the Kinect field of view. This occurs when a second sensor confuses the IR pattern projected by a first sensor with its own. Several academic studies have developed methods to reduce overlap interference of multiple sensors operating simultaneously or the use of multiple sensors for indoor human tracking (Maimone *et al.,* 2012; Utama *et al.,* 2012). A sensor-scheduling algorithm for object tracking using four Kinect sensors operating simultaneously was proposed in Faion *et al*. (2012). Developing such system would be a great advantage because we would be able to monitor, measure, and document all incoming and outgoing material in real-time.

**2.6 Discussion of Concept Implementation**

A real-time dock door monitoring system would facilitate and improve dock supervision in manufacturing, storage, and distribution buildings. It would aid managers to remotely monitor a live video feed video of unloading/loading operations and verify historical data of entry/exit instances on a given dock door. It would also increase safety as approaching speeds would be constantly monitored and speed-safety violations would be documented with a picture and time of the occurrence for accountability purposes. Furthermore, the ability of this system to automatically log timestamps allows daily dock door metrics to be monitored in real-time; allowing management to predict performance and prioritize jobs or dock doors. Also, personnel performance could be compared in order to determine who performs better under certain conditions, *e.g.*, loading/unloading heavy or tall pallets difficult to handle. This information is key for assigning jobs or balancing workloads in order to reduce unloading/loading times. The proposed real-time dock door monitoring system could also be used to monitor dock operations trainees to understand when they are far enough into the learning curve as to be able to perform well in a real dock environment.

The cost per door of implementing the Kinect sensor-based real-time dock door monitoring system is relatively low. The current generation of the Kinect for Windows V2 costs approximately $250, whereas each desktop computer costs a few hundred dollars. Alternatively, any Window-based computer could be used to connect to a virtual computer. The assembly of the system would require a sensor to be strategically placed above each door. It might also require extended USB and power cables, and internet access, depending on the location of the computer and connection requirements.

### 2.7 Framework Conclusion and Future Work

This Section presents a proof of concept for of a Kinect-based automated real-time dock door monitoring system. An automated dock door monitoring system could provide valuable real-time data on receiving and shipping dock operations. The data collected may be used to help dock managers to: remotely visualize the docks' status, monitor current and projected productivity standards and performance metrics, and manage dock operations, including supervising employees, job prioritization, trailer/truck security, and traffic safety enforcement.

The proposed system has the capability to automatically identify when objects approach a dock door, records timestamps that can be used for time studies and performance monitoring, measure the time incurred in loading/unloading individual loads and trailers. The system can also monitor approaching speeds, which may be used to monitor safety compliance. Time and speed data could be used to compare the difficulty in handling tall and heavy pallets with short and light pallets to assign the corresponding time allowances while developing work standards. The developed prototype of the system also automatically exports a color image screenshot to document all incoming and outgoing material and personnel, and address any accountability issues related to excess speed events or trailer safety. It also exports a depth data frame that can be used to recognize objects that pass below the sensor. The advantage of using the Kinect sensor is its cost, combination and quality of sensors, and the capability to control it with a computer using customized software.

The second generation of Kinect for Windows V2 is already available and has a range up to 4.5 m (14.8 ft.), a wider field of view with a sharper depth image and a 1080p color stream resolution, and is capable of facial and expressions recognition. Carlo, et al (2014) provides additional details on how this proof of concept can become part of a decision making tool for dock managers.

A Kinect-based sensor system as the one described in this Chapter can also be used to gather operational-level data at other areas of a DC. Examples of some data that can be collected with such a monitoring system includes: material handlers' location and status (via tracking them and the number of loads in a flow-rack (via the depth sensor). Hence, in the next Chapter we solve the receiving logistics problem assuming that this operational-level information is available to the material handlers when deciding where to place the pallets.

# 3. Receiving Logistics Problem Introduction

Receiving logistics in DCs involves the following steps: *receiving inbound trailers*, *assigning inbound trailers to doors*, *scheduling of inbound trailers at dock doors*, *scheduling (trailer) unloaders*, *pallet staging (including identification and quality control)*, and *haulers scheduling for put-away*. Figure 7 depicts the traditional layout of DCs. Loads arrive through the receiving side of the DC (the left side in Figure 7), they are moved to the storage locations via a staging area, and they are eventually picked and shipped from the another side of the DC.

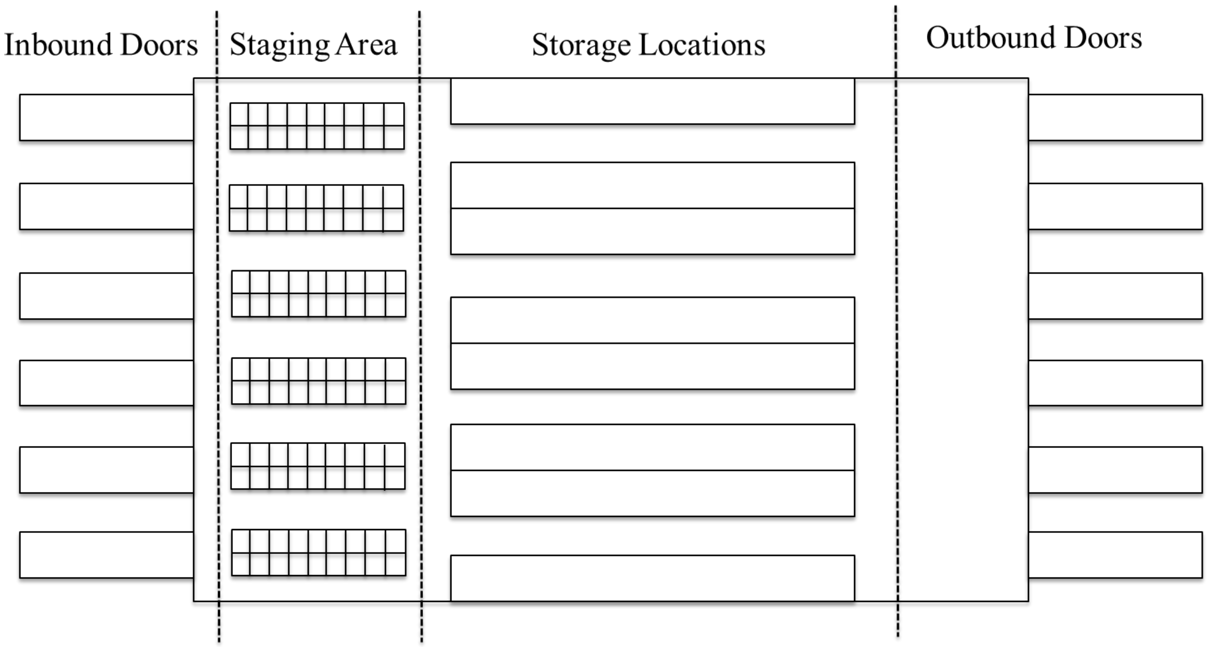



*Figure 7. Distribution Center General Layout with Staging Areas*

### 3.1 Receiving Inbound Trailers

Receiving logistics start with receiving the inbound trailers and assigning them to a designated parking area in the storage yard until they are retrieved to be unloaded on an inbound door (Carlo and Aponte-Rodríguez, 2012). In this study it is assumed that all trailers are available in the parking area so that any trailer may be assigned to any door anytime. Clearly, in reality trailers arrive sporadically depending on when they are released from their previous location. However, in general, if the DC unloading schedule for a trailer is known (e.g., an appointment system), prioritization can be given to that trailer at the origin previous location in order to comply with the unloading schedule at the DC. Trailer unloading operations honor a level-of-service (LOS); this means that all trailers must be unloaded within a given time frame. In practice, receiving dock managers usually follow the *first-come-first-served* (FCFS) priority rule, but on occasions they are forced to prioritize truckloads that contain high-demand items. On other occasions trailers have to be held off because they contain seasonal items that cannot be shipped yet. Prior to each unloading shift, a trailer selection must be made. This decision is usually left to receiving dock managers due to their prior knowledge of truckload contents.

### 3.2 Inbound Trailers to Doors Assignment and Scheduling

In this receiving logistics stage each inbound trailer is assigned an inbound dock door. Given the physical nature of dock doors, only one trailer may be unloaded from each door at a time. The door assignment decision needs to consider the storage locations of the truckload contents; the best door for a certain trailer would be the one that minimizes the *total parts travel* (*i.e.*, total travel distance between dock door and storage locations). Usually, warehouses with dedicated storage policies are arranged by type or categories of products; similar to the way they are received in the truckloads. In such scenario the door assignment decision would be simple, just select the door closest to the aisle in which those type or class of items are stored. The door selection becomes a much harder when the truck load composition is highly mixed (Bozer and Carlo, 2008) (*i.e.*, storage locations are scattered through different aisles) or if it has a high volatility (Carlo and Bozer, 2011) (*i.e.*, the composition of the loads change daily).

Although minimizing the total parts travel will yield the best door assignment for a given trailer, it is not the appropriate objective function when considering all the inbound logistics. Notice that a potential optimal solution that minimizes total parts travel may assign all inbound trailers to the same inbound door. The appropriate objective function to optimize inbound logistics is to minimize the makespan, for which the total parts travel is one of its components. We define makespan to be the total time of the unloading operation for a given set of trailers to be unloaded on any given day; the operation begins since the first pallet from the first trailer is unloaded and ends when all pallets from all trailers are delivered to their storage location. Hence, the objective function considered in this thesis to optimize the inbound logistics is to minimize the makespan.

### 3.3 Unloader Scheduling

Once the trailer-to-door assignment and scheduling are determined, the unloading operations at the receiving dock begin. *Unloaders*, who are material handlers tasked with unloading the inbound trailers with dedicated equipment (*e.g.*, counter-balanced lift trucks), are assigned to each trailer, so that multiple trailers may be unloaded simultaneously throughout the receiving dock floor. Material handling equipment is selected based on the types of loads to be handled. Material handling equipment used in DCs include counter-balanced fork lift trucks, pallet jacks, clamp trucks, automated guided vehicles (AGVs) (Thompkins *et al.,* 2010). The unloader scheduling problem seeks to determine which unloader will unload each trailer on each door, and in which sequence, to place arriving pallets on a staging area.

### 3.4 Pallet Staging Location

In practice, unloaded pallets are placed on floor area by the sides or even in front of the active dock door, where they await to be inspected, documented, and labeled. Afterwards, material handlers called *haulers* pick-up the pallets and move them to their assigned storage location. When pallets are placed on the sides of the door, they block adjacent doors; therefore, trailers on these doors cannot be unloaded simultaneously. If pallets are placed in front of the door, they take up space between the receiving dock and the storage racks, mainly where haulers navigate the receiving dock. The selection of pallet staging areas should consider door blockage conditions due to the fact that the trailer located at a blocked door cannot be unloaded until the pallets occupying space of receiving dock are cleared by the haulers. This could be used to determine unloading sequences at the dock because pallets could be staged in front of a door that contained a trailer that was just unloaded. This allows haulers to clear pallets while the trailer yard driver is switching empty trailers for full trailers, and trailers idle times caused by block doors are avoided. The pallet staging problem is to determine the staging area assigned to each incoming pallet so the unloader can perform the transport move. The selection of the pallet staging location for all loads is considered in the development of the mathematical model and ruled-based heuristics.

### 3.5 Haulers Scheduling for Put-away

After unloaders move the pallets to their staging area, haulers will retrieve the pallets and move them in their final storage locations. The hauler scheduling problem seeks to determine the sequence in which haulers will transport loads from the staging area to their storage location.

### 3.6 Scenario Description and Cross-Dock Comparison

Figure 8 depicts inbound intralogistics personnel assuming a flow rack as the staging area. As described above *unloaders* move loads from inbound doors to the staging area. On the other hand, *haulers* put-away loads that were placed in the flow racks by the unloaders. Unloaders and haulers should collaborate to store the inbound loads as quickly as possible.

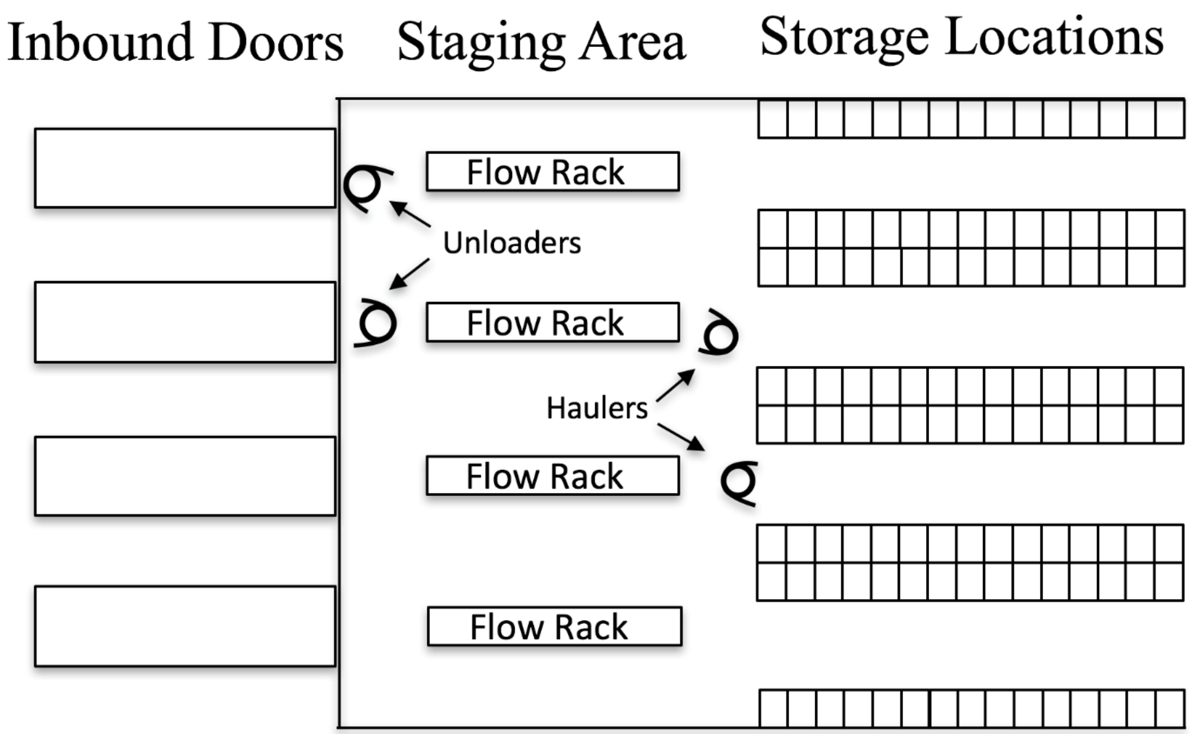



*Figure 8. Distribution Center Intralogistics*

The inbound intralogistics operations in cross-docks (XDs) closely resemble those of DCs. XDs can be described as deconsolidation/consolidation facilities where loads are unloaded and immediately recombined with loads sharing the same destination (Bozer and Carlo, 2008). In traditional XDs the inbound logistics are much simpler than the one in DCs as loads are moved directly from inbound to outbound trailers by unloaders. Vis and Roodbergen (2008) and Yu and Egbelu (2008) describe a special type of XDs where pallets are assigned to an intermediary shortterm storage location (*i.e.*, staging area) after being unloaded in order to decouple inbound and outbound operations. The inbound logistics of these XDs with staging areas is very similar to the one in DCs, with the only difference that in XDs loads are placed in outbound trailers located at outbound doors (*a.k.a.*, stack doors) instead of storage locations. Clearly, if the outbound trailer (or destination)-to-door assignment is known and outbound trailer schedules are disregarded, the XD with stages inbound intralogistics becomes identical to the one described for DCs. Figure 9 illustrates the receiving intralogistics for XDs with staging areas.

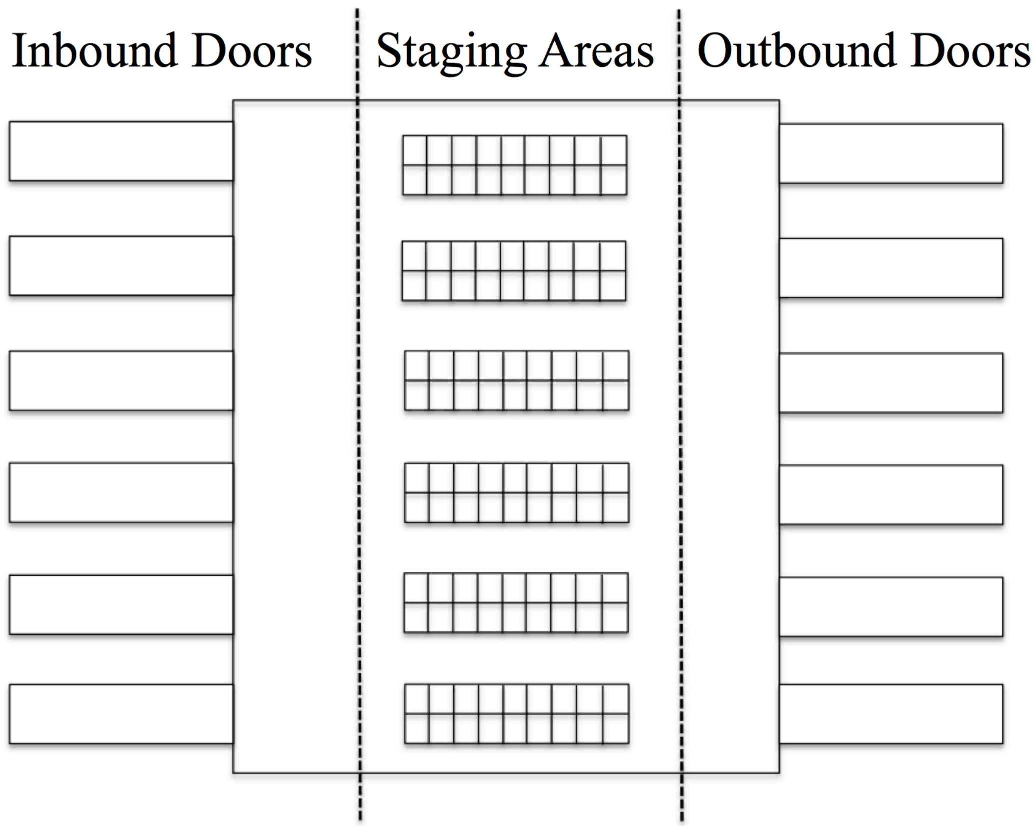



*Figure 9. XDs with Staging Areas*

The main assumptions considered for modeling this problem are the following:

- Inbound trailers with known palletized load composition are available in the yard when requested.
- Trailer-to-door assignment and sequence is given prior to the beginning of the unloading operations.
- Pallet identification and quality control operations are disregarded and do not affect the availability and quantity of loads to be stored.

- The number and type of material handling equipment used is known a priori.
- Pallets in the staging area use a pallet flow rack (*i.e.*, they do not block each other and they have fixed input and output points) so their availability is according to the FIFO policy.
- Pallet inspection, documentation, and labeling are not considered (*i.e.*, assumed to be integrated into the flow racks).
- The storage location for each pallet is known (e.g., it is determined by a WMS).
- There is no pre-emption for unloaders; *i.e.*, once an unloader starts a trailer, he must unload all loads before starting the next trailer.
- Unloaders and haulers travel rectilinear distances.
- Travel time is a linear function of distance (*e.g.*, acceleration/deceleration are infinite)
- All trailer contents, travel distances, number of unloaders and haulers, and storage locations are assumed known.

## 4. Literature Review

Gu *et al.* (2006) published a review on warehouse operations, which explains main research topics areas and objectives. The literature is divided into the four basic warehouse functions: *receiving*, *storage*, *order picking*, and *shipping*. According to Gu *et al.* (2006) the available research on receiving and shipping operations is very limited. In fact, all references made in Gu *et al.* (2006) to receiving and shipping operations in warehouses are related to the dock door assignment problem in XDs. Buijs *et al.* (2014) present an exhaustive literature review and classification of XDs. The following are the only published work purely on the dock door assignment problems in literature (i.e., truck scheduling is not considered).

Tsui and Chang (1990, 1992) developed a bilinear programming model to solve the dock door assignment problem (i.e., inbound and outbound door-to-trailer assignment) in a less-than-truckload (LTL) XD. The model minimizes travel distances between inbound and outbound doors assuming that inbound and outbound doors are on opposite sides of the facility. A decomposition heuristic and a branch & bound algorithm, respectively, are proposed to solve the problem. The problem in Tsui and Chang (1990,1992) is similar to the required dock door assignment in this thesis, except that the travel distances in DCs include unloading from inbound doors to staging areas and from staging areas to storage locations, which is performed by a different material handler.

Gue (1999) developed a model to determine the arrangement of strip (inbound) and stack (outbound) doors, and the assignments of destinations to stack doors of an LTL terminal. The main conceptual difference in this paper is that it considers how supervisors tend to assign inbound doors closest to their destinations (outbound doors) in order to reduce travel distances. Hence, a look-ahead scheduling is proposed, where the material flow depends on the layout if the supervisor assigns incoming trailers to doors based on the contents and the location of the outbound door. In order to produce a layout, the author solves two problems: (1) estimate labor cost of material flows caused by look-ahead scheduling for a given layout, and (2) searching the solution space of all layouts to determine the one with lowest cost. To solve the first problem, the author develops a parametric model of material flows that considers the supervisor's look-ahead scheduling policy. He implements a parameter in the model that represents the level of influence the supervisor has over flows in the terminal. Such level of influence determines how close the inbound door will be to the outbound doors, considering the mix of freight inside incoming trailers and on the length of the queue for trailers to be assigned. For the second problem, the author uses a local search algorithm to find the best layout for a given level of influence. Experimental results show that using the proposed look-ahead algorithm reduced travel related labor cost by more than 15% than if a first-come-first-serve (FCFS) policy is used. Such study provides valuable insight to our work because the author was able to incorporate a parameter in the model that accounts for manager's influence (look-ahead policy) on the trailer selection and door assignment process, similar to the topic discussed earlier on Section 2.1, of prioritizing truckload selections for unloading based on their contents.

Bartholdi and Gue (2000) developed a cost model for determining the layout of LTL terminals, which considers travel times and waiting times caused by internal congestion, assuming average trailers are received at the docks. The authors explain how shortest door-to-door distance is not a good measure of travel time because actual travel depends on the type of freight, equipment and local freight movement rules. They developed a cost model, which not only considers travel with different types of material handling systems but also considers equipment interference and floor space congestions.

One common aspect of these door assignment XDs related works is that the authors consider static door assignments. This means that the outbound doors are fixed over a planning horizon while the inbound doors are reassigned every day. Bozer and Carlo (2008) study both static and dynamic door assignments. In the dynamic door assignment outbound door-to-destinations are also changed every night. Bozer and Carlo (2008) formulate the dock door assignment problem as a quadratic assignment problem. A simulated annealing-based heuristic is proposed to minimize the total parts travel. In Carlo and Bozer (2011) the authors analyze the properties of optimal inbound and outbound trailer-to-door assignments.

Previously discussed works do not consider pallet-staging positions in the door assignment process. Vis and Roodbergen (2008) developed a network formulation for XDs with staging areas in order to assign temporary storage locations for inbound pallets such that the total parts travel is minimized. A block assignment approach is used, where a block represents a predetermined time period in which loads that arrive are assigned an available storage location until they are picked up for loading. When positions become empty, they can be used for the assignment of the next block. Their "*row based storage assignment*" algorithm assigns cost to flows depending on the storage locations available, if the location is on the row on shortest path between the inbound and outbound door, no cost is assigned. If another row that is not on the shortest path is selected, a cost that accounts for the extra travel distance is assigned. The authors use a minimum cost flow algorithm to determine the flows that go to each intermediary storage location that minimizes total travel distance. The main difference between Vis and Roodbergen (2008) and this thesis is that they focus on the strategic level decision of which staging area to use for each pallet, whereas the focus of this thesis is at the operational level – which also includes when will the movement take place and who will execute it. Also, they use predetermined time periods or "time blocks" in which assignments are made, in this thesis multiple decisions are made on an operational level continuous-time basis. Another difference would be our objective function to minimize makespan instead of minimizing total travel. The problem addressed in this thesis also considers the workforce capacity (*i.e.*, the assignment and scheduling of unloaders and haulers) and the staging area capacity.

Examples of strategic-level decisions include the design of the DC which dictates the number of dock doors, number of flow racks, the MHE used, and the travel distances. Examples of tactical-level decisions include the number of inbound trailers received, the number of unloaders and haulers, the storage location assignment, etc. Given the strategic and tactical-level decisions, we focus on assisting dock managers to make day-to-day decisions regarding assignment and scheduling of inbound logistics (which is the focus of this thesis).

Buijs and Vis (2014) state that the main performance indicator for the XD manager is to maximize the throughput rate, consisting of three interrelated components: *size of the workforce*, *freight volume handled*, and *makespan*. They state that freight volumes are the result of transport planning, so they cannot be influenced by the XD manager, which also has little influence in the makespan as it is largely determined by the planned arrival and departure times of trailers. Hence, to maximize throughput managers should focus on maximizing the productivity of the workforce by planning the workforce capacity over time and by assigning material handlers so that operations are performed efficiently. The only inbound intralogistics publication that considers workforce restrictions as part of inbound logistics is Shakeri *et al.* (2012), which focuses on the truck scheduling problem at XDs considering the availability of resources such as doors and material handlers (unloaders) to minimize makespan. The dock door assignment is solved as part of the truck scheduling problem. A two-phase heuristic is proposed where a heuristic is used to determine a feasible sequence of trucks for the door assignment, and a rule-based heuristic is used to assign each sequenced truck to dock door. Our work is then the second publication that considers workforce productivity in receiving logistics.

## 5. Mathematical Formulation

In this Chapter we present a mathematical model for the *receiving logistics problem* (RLP). The model assigns unloaders-to-trailers, pallets-to-flow racks, and haulers-to-pickup pallets in flow racks. The objective of the model is to minimize the makespan to serve all trailers (until the last pallet is transported by a hauler and delivered to its storage location). The model presented considers pallet flow racks blocking and provide FIFO access to loads for the hauler. Clearly, the formulation of the problem in this thesis may be obtained by simplifying the staging requirements in this model. Note that this model was developed for non-automated storage operations.

**Sets**

$J$ – set of trailers, indexed on $j = 1, 2, \ldots, |J|$

$I_j$ – set of loads (pallets) per trailer, indexed on $i = 1, 2, \ldots |I|$ ; $I_j = I_1, \ldots |I_J|$

$K$ – set of doors, indexed on $k = 1,2, \ldots, |K|$

$F$ – set of flow racks, indexed on $f = 1, 2, \ldots, |F|$

$U$ – set of unloaders, indexed on $u = 1, 2, \ldots, |U|$

$H$ – set of haulers, indexed on $h = 1, 2, \ldots, |H|$

$T$ – set of times, indexed on $t = 1, 2, \ldots, |T|$

**Parameters**

$\delta$ – trailer change-over time

$q_f$ – Flow rack capacity (in pallets)

$\tau$ – Time it takes one pallet to move through an empty flow rack.

$d_{jf}$ – distance from the door to which trailer $j$ is assigned to flow rack $f$ (expressed in time)

$d_{fij}$ – distance from flow rack $f$ to storage location of pallet $i$ from trailer $j$ (expressed in time)

$a_{jk}$ – 1 if trailer $j$ is assigned to door $k$, 0 otherwise

$a_{jj'k}$ – 1 if trailer $j$ is assigned to door $k$ sometime before trailer $j'$, 0 otherwise

$b_{ii'j}$ –1 if pallet $i$ has to be unloaded before pallet $i'$ from trailer $j$ due to their position in the trailer.

$M$ – a very large constant

**Variables**

$w$ – makespan

$x_{jk}^{t}$ – 1 if trailer *j* is available for stripping on door *k* at time *t,* 0 otherwise

$y_{ju}^{t}$ – 1 if unloader *u* begins unloading trailer *j* on time *t,* 0 otherwise

$z_{ijf}^{t}$ – 1 if pallet *i* from trailer *j* is placed on flow rack *f* on time *t,* 0 otherwise

$w_{ijfh}^{t}$ –1 if pallet *i* from trailer *j* is retrieved from flow rack *f* by hauler *h,* 0 otherwise

$E_{iji'j'f}$ – 1 if pallet *i* from trailer *j* is placed at flow rack *f* before pallet *i'* from trailer *j',* 0 otherwise

$P_{iji'j'h}$ – 1 if hauler *h* retrieves pallet *i* from trailer *j* before pallet *i'* from trailer *j',* 0 otherwise

**Objective Function**

The objective function in Eq. (1) is to minimize the makespan, which is computed in Eq. (2) as the time the pallet is retrieved from the flow rack and transported to its predefined storage location:

$$\text{Min } z_{RLP} = w \quad (1)$$

The objective function is subjected to the following constraints: Constraint sets (3) to (11) are associated with the receiving intralogistics operations up to the point where pallets are placed on the flow rack. Constraint sets (12) to (18) pertain the remaining receiving intralogistics process. Equations (19) to (24) are variable declaration constraints sets.

Makespan calculation:

$$w \geq w_{ijfh}^{t}\left(t + d_{fij}\right) \;\; \forall\, i \in I_j, j \in J, f \in F, h \in H, t \in T \quad (2)$$

Each trailer should be available to unload once (unique ready time) and in exactly the door to which it was assigned to:

$$\sum_{t \in T} \sum_{k \in K} a_{jk} x_{jk}^{t} = 1, \qquad \forall\, j \in J \quad (3)$$

Each trailer should arrive at a door before the unloader starts serving it:

$$\sum_{t\in T} t x_{jk}^{t} \leq \sum_{t'\in T}\sum_{u\in U} t' y_{ju}^{t'}, \qquad \forall\, j \in J, k \in K \tag{4}$$

For each trailer, unloading should begin once and by one unloader:

$$\sum_{t\in T}\sum_{u\in U} y_{ju}^{t} = 1 \quad \forall\, j \in J \tag{5}$$

Each pallet from all trailers should be dropped at one time and one flow rack:

$$\sum_{t\in T}\sum_{f\in F} z_{ijf}^{t} = 1 \quad \forall\, i \in I_j, j \in J \tag{6}$$

No more than one pallet can be dropped at the same time at any flow rack:

$$\sum_{i\in I_j}\sum_{j\in J} z_{ijf}^{t} \leq 1 \quad \forall\, f \in F, t \in T \tag{7}$$

The start time for the following trailer should be greater than the unloading time of the last pallet of the previous trailer plus trailer changeover time, (if they precede on the same door):

$$\sum_{t'\in T}\sum_{u\in U} t' y_{j'u}^{t'} \geq \delta + \sum_{t\in T}\sum_{f\in F} t z_{\|I_j\| jf}^{t} - M\left(1 - a_{jj'k}\right) \quad \forall\, j \in J, j' \in J, k \in K \tag{8}$$

All pallets from the same trailer should be dropped at a flow rack on the same order they are unloaded:

$$\sum_{t'\in T}\sum_{f'\in F} \left(t' - d_{jf'}\right) z_{i'jf'}^{t'} \geq \sum_{t\in T}\sum_{f\in F} \left(t - d_{jf}\right) z_{ijf}^{t} - M\left(1 - b_{ii'j}\right) \quad \forall\, i \in I_j, i' \in I_j, j \in J \tag{9}$$

The next constraint defines an indicator variable $\left(E_{iji'j'f}\right)$ that establishes the pallet sequence of all pallets *i* and *j* on any flow rack *f*, notice that the last two terms are required to ensure that both pallets were routed through flow rack *f* :

$$\sum_{t'\in T} t' z_{i'j'f}^{t'} - \sum_{t\in T} t z_{ijf}^{t} \leq M\left(E_{iji'j'f}\right) + M\sum_{t\in T}\left(1 - z_{ijf}^{t}\right) + M\sum_{t'\in T}\left(1 - z_{i'j'f}^{t'}\right)$$

$$\forall\, i \in I_j, j \in J, i' \in I_{j'}, j' \in J, f \in F \tag{10}$$

All pallets dropped at a flow rack should be picked up by a hauler from the same flow rack:

$$\sum_{t\in T}\sum_{i\in I_j}\sum_{j\in J}\sum_{h\in H} w_{ijfh}^{t} = \sum_{t\in T}\sum_{i\in I_j}\sum_{j\in j} z_{ijf}^{t} \quad \forall\, f \in F \tag{11}$$

The following constraint accounts for a pallets travel time in any flow rack:

$$\sum_{t\in T}(t+\tau) z_{ijf}^{t} \leq \sum_{t'\in T}\sum_{h\in H} t' w_{ijfh}^{t'} \quad \forall\, i \in I_j, j \in J, f \in F \tag{12}$$

For each flow rack, there must be a pallet in order to retrieve it, no pallet should be deposited if the flow rack is at maximum capacity:

$$0 \leq \sum_{t'\leq t}\sum_{i\in I_j}\sum_{j\in j} z_{ijf}^{t'} - \sum_{t'\leq t}\sum_{i\in I_j}\sum_{j\in J}\sum_{h\in H} w_{ijfh}^{t'} \leq\ q_f \quad \forall f \in F, t \in T \tag{13}$$

Each pallet should be picked by one hauler from one flow rack at one time:

$$\sum_{t\in T}\sum_{f\in F}\sum_{h\in H} w_{ijfh}^{t} = 1 \quad \forall\, i \in I_j, j \in J \tag{14}$$

At any time no more than one pallet can be picked up from any flow rack:

$$\sum_{i\in I_j}\sum_{f\in F,}\sum_{h\in H} w_{ijfh}^{t} \leq 1 \quad \forall\ j \in J, t \in T \tag{15}$$

If a pallet precedes another in any flow rack $\left(E_{iji'j'f}\right)$, the first to be deposited should be picked up for hauling before the next, paired with Eq. (11):

$$\sum_{t'\in T}\sum_{h'\in H} t' \cdot w_{i'j'fh'}^{t'} - \sum_{t\in T}\sum_{h\in H} t \cdot w_{ijfh}^{t} \leq M\left(E_{iji'j'f}\right) + M\sum_{t\in T}\left(1 - z_{ijf}^{t}\right) + M\sum_{t'\in T}\left(1 - z_{i'j'f}^{t'}\right)$$

$$\forall\, i \in I_j, j \in J, i' \in I_{j'}, j' \in J, f \in F \tag{16}$$

The following constraint set activates the indicator variable $\left(P_{iji'j'h}\right)$ in the case that pallet *i* from trailer *j* is served by hauler *h* before pallet *i'* from trailer *j'* in a similar fashion than Eqs. (10) and (16)

$$\sum_{t' \in T} \sum_{f' \in F} t' \cdot w_{i'j'f'h}^{t'} - \sum_{t \in T} \sum_{f \in F} t \cdot w_{ijfh}^{t}$$
$$\leq M\left(P_{iji'j'h}\right) + M \sum_{t \in T} \sum_{f \in F} \left(1 - w_{ijfh}^{t}\right) + M \sum_{t' \in T} \sum_{f' \in F} \left(1 - w_{i'j'f'h}^{t'}\right)$$
$$\forall\, i \in I_j, j \in J, i' \in I_{j'}, j' \in J, h \in H \qquad (17)$$

The following constraint set (18) ensures that the pallet retrieval times from the flow racks include the haulers' travel time to and from the respective storage locations:

$$\sum_{t \in T} \sum_{f \in F} w_{ijfh}^{t}\left(t + d_{fij}\right) \leq \sum_{t' \in T} \sum_{f' \in F} w_{i'j'f'h}^{t'}\left(t' - d_{f'ij}\right) + M\left(1 - P_{iji'j'h}\right)$$
$$\forall\, i \in I_j, j \in J, i' \in I_{j'}, j' \in J, h \in H \qquad (18)$$

**Variable Declarations**

$$x_j^t \in \{0,1\}\ \forall j \in J, t \in T \qquad (19)$$

$$y_{uj}^t \in \{0,1\}\ \forall u \in U, j \in J, t \in T \qquad (20)$$

$$z_{ijf}^t \in \{0,1\}\ \forall i \in I_j, j \in J, f \in F, t \in T \qquad (21)$$

$$w_{ijfh}^t \in \{0,1\}\ \forall i \in I_j, j \in J, f \in F, h \in H, t \in T \qquad (22)$$

$$E_{iji'j'} \in \{0,1\}\ \forall i \in I_j, j \in J, i' \in I_{j'}, j' \in J \qquad (23)$$

$$P_{iji'j'h} \in \{0,1\}\ \forall i \in I_j, j \in J, i' \in I_{j'}, j' \in J, h \in H \qquad (24)$$

### 5.1 Mathematical Complexity of the problem

The total makespan in the RLP includes three main components: unloaders, flow racks, and haulers. If one considers a very simple case of the problem with a single unloader and a single hauler as resources, the problem becomes a two-stage setup dependent machine scheduling problem where the pallets are the jobs and the resources are the machines. This problem can be polynomially reduced to setup dependent single machine scheduling problem (SDMSP) with the objective of minimizing makespan. Hence the RLP is a polynomial reduction of the SDMSP (*i.e.*, SDMSP $\leq_p$ RLP). Lenstra *et al.* (1977) show that SDMSP is *NP-hard*, meaning that since the RLP contains multiple parallel machines and stages it must be at least *NP-hard*. Given mathematical the complexity of the problem, heuristic methods should be considered to solve the problem.

### 5.2 Model Reformulation

As suggested in Buijs and Vis (2014) the focus of a XD manager should be on managing the internal resources as to maximize workforce productivity. We now argue that given a trailer-to-door assignment and sequence, minimizing makespan may be viewed as a resource balancing problem. Consider the logic below:

$$Min\ makespan = Min(\underset{loads}{\text{Max}}\{ready\ time + time_{trailer-flow\ rack} + time_{flow\ rack-storage}\})$$

Makespan may also be considered from the perspective of the hauler and the objective function becomes:

$$\begin{aligned} Min\ makespan \\ &= Min(\underset{haulers}{\text{Max}}\{idle\ time + travel\ time_{flow\ rack-storage} \\ &+ travel\ time_{storage-flow\ rack}\}) \end{aligned}$$

Since travel time is assumed rectilinear we consider both vertical and horizontal components, although the horizontal component of travel time is fixed, the objective function becomes:

$$Min\ makespan$$
$$= Min(\underset{haulers}{\text{Max}}\{idle\ time$$
$$+ vertical\ travel\ time_{flow\ rack-storage} + horizontal\ travel\ time_{flow\ rack-storage}$$
$$+ vertical\ travel\ time_{storage-flow\ rack} + horizontal\ travel\ time_{storage-flow\ rack}\})$$

Since travel time is assumed to be a linear function of time we have:

$$Min\ makespan$$
$$= Min(\underset{haulers}{\text{Max}}\{idle\ time$$
$$+ vertical\ travel\ distance_{flow\ rack-storage} + horizontal\ travel\ distance_{flow\ rack-storage}$$
$$+ vertical\ travel\ distance_{storage-flow\ rack} + horizontal\ travel\ distance_{storage-flow\ rack}\})$$

However, since the horizontal travel distance component is fixed, the makespan is a function of: haulers' idle time, haulers' vertical travel distance from flow rack to storage, and haulers' vertical travel distance from storage to flow rack. Therefore, to minimize the makespan one needs to ensure that haulers are kept busy (minimize idle time) and that loads are placed in the flow rack closest to their storage location. Unfortunately, the two components of this objective function may be contradictory. Notice that to minimize hauler idle time unloaders need to feed the flow racks as fast as possible. This is done by placing loads in the closest flow rack. Clearly, if the loads are placed by unloaders in the closest flow rack, then they may not end in the flow rack closest to the storage location (to minimize haulers' vertical travel).

## 6. Rule-based Heuristics

Given on the mathematical complexity of the RLP problem, alternate methods are required to solve the problem. This Section proposes four ruled-based heuristics to carry out the receiving logistics process. We define rule-based heuristics as simple, yet efficient, *rule-of-thumb* strategies that a manager can implement without requiring software. Generally, each rule-based heuristic has a different set of unloading rules, which are based on the intuition gained from the reformulation of the problem that suggests that minimizing makespan requires a work-balance between haulers and unloaders. Each proposed heuristic varies in the decision unloaders make at the time of selecting a flow rack for all loads. This occurs once the unloader exits the trailer with the loaded pallet. Flow rack selection has an effect on vertical distance travelled for the unloaders, whether the selected flow rack is further or closer to the unloading door.

For all heuristics, unloaders and haulers are available for assignment at an initial common location labeled home in Figure 10, if either resources do not have loads to attend, they will move to the closest idle location to ensure they do not physically block other resources. Unloaders will always select the next trailer available located on the closest unloader door to their initial or idle location. As for the haulers, the load selection decisions remain constant through all heuristics, they will choose the closest load to their position. For example, in the beginning of the process they will select the load on the closest flow rack to their initial position. Once they have delivered the load to its respective aisle and storage position, the hauler returns through the same aisle to select the following load located on the closest flow rack.

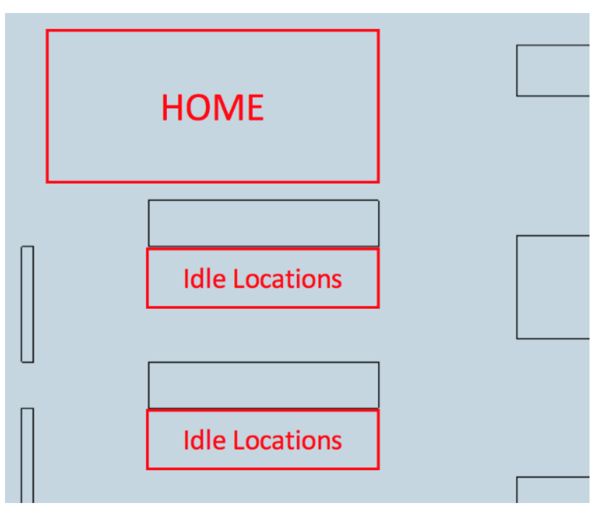



*Figure 10. Initial and Idle Locations*

**6.1 H1 - Minimize Unloader Vertical Travel**

In H1 each load will be moved by the unloader to the flow rack closest to the load's inbound door on the way to the storage location. For example, Figure 11 shows an unloading scenario where the load is unloaded from the 4th door (from the top), and it will be stored on the 1st aisle (from the top), therefore the unloader will select the 4th flow rack, which is closest to the load's door. If the selected flow rack is full, the load will be placed in the first available flow rack on the way to the storage location (Figure 12). We refer to this flow rack as the *logical flow rack* as it is in the minimal required vertical travel distance for the load. The logic behind H1 is to minimize the unloaders' loaded vertical travel distances, which in turn minimizes the haulers' idle time and their expected vertical travel distance as loads become available faster.

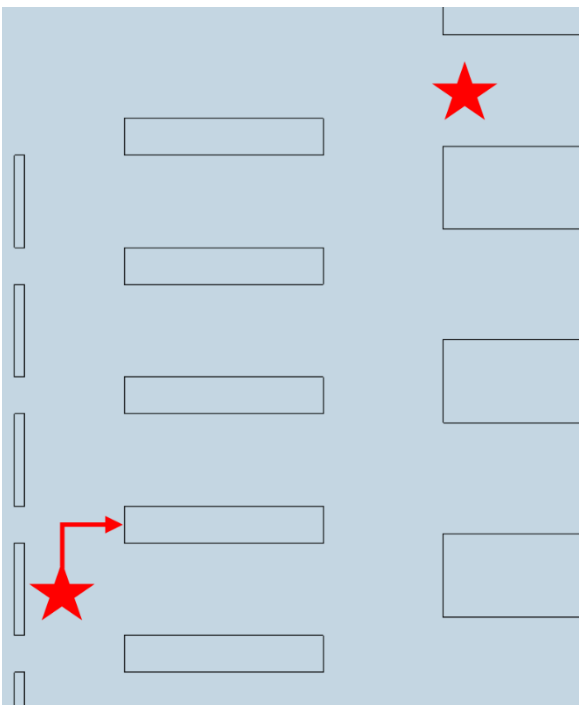


*Figure 11. H1 Flow Rack Selection*

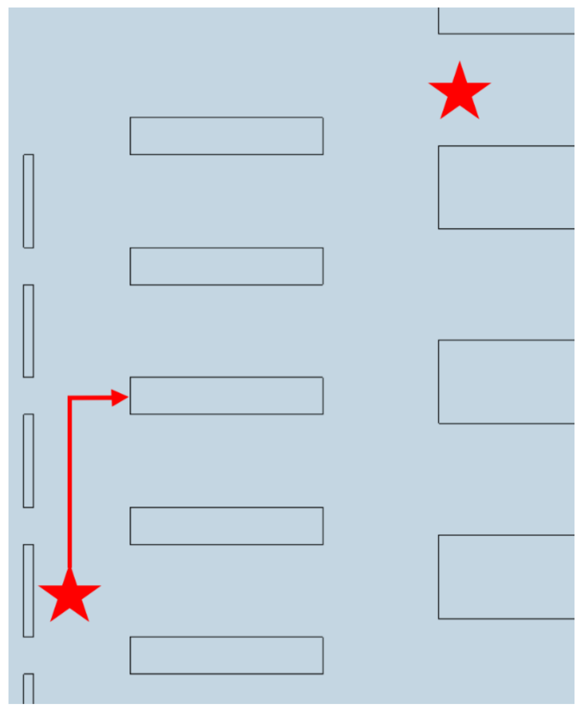


*Figure 12. H1 Next Available Flow Rack*

### 6.2 H2 - Unloader Adapts to Haulers' Workload

In H2 unloaders will verify if any hauler is idle in order to determine in which flow rack to place a load. If there is an idle hauler, the unloader will place the load in the flow rack closest to the inbound door (Figure 13); otherwise the load will be placed in the flow rack closest to the storage location (Figure 14). If the desired rack is full, the next logical rack is selected. The data input required by H2 may be obtained by monitoring system such as the one described in Chapter 2. H2 is designed to favor the hauler by minimizing their idle times and loaded vertical travel distances.

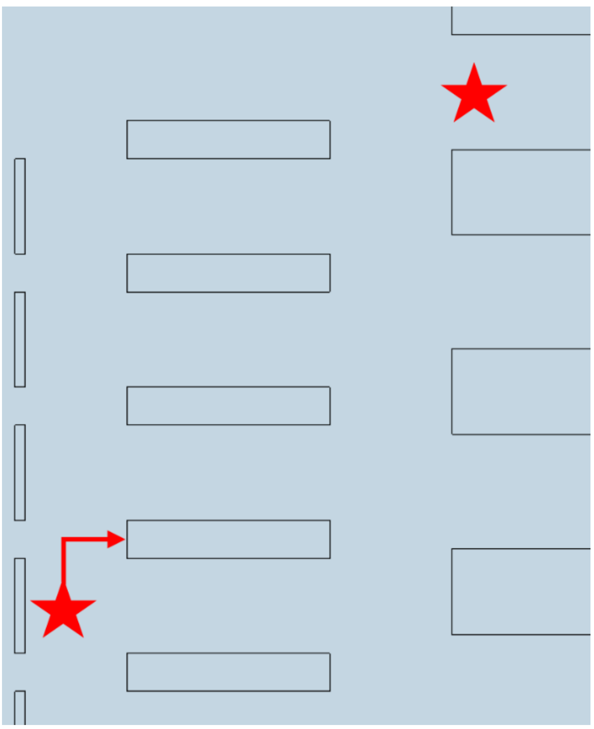


*Figure 13. H2 Flow Rack Selection when some Hauler is Idle*

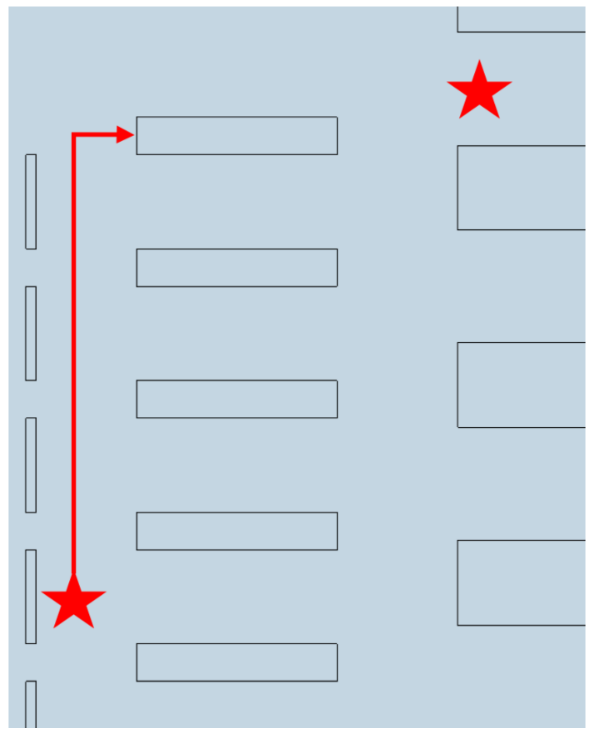


*Figure 14. H2 Flow Rack Selection When All Haulers Busy*

**6.3 H3 - Least Utilization Rack Closest to Storage Location**

In this heuristic-rule the unloaders will select the flow rack with the least utilization that is located between the load and its storage location. Ties will favor the flow rack closest to the pallets' storage location. For example, Figure 15 shows the number of pallets in each flow rack, the 2nd flow rack will be selected because it has the lowest number of pallets, and it is the closest to the loads storage aisle. The data input required by H3 may be obtained by monitoring system such as the one described in Chapter 2. H3 is based on the observation that after storing a load, haulers pass by a flow rack that if full provides the next job, hence minimizing haulers' empty vertical travel.

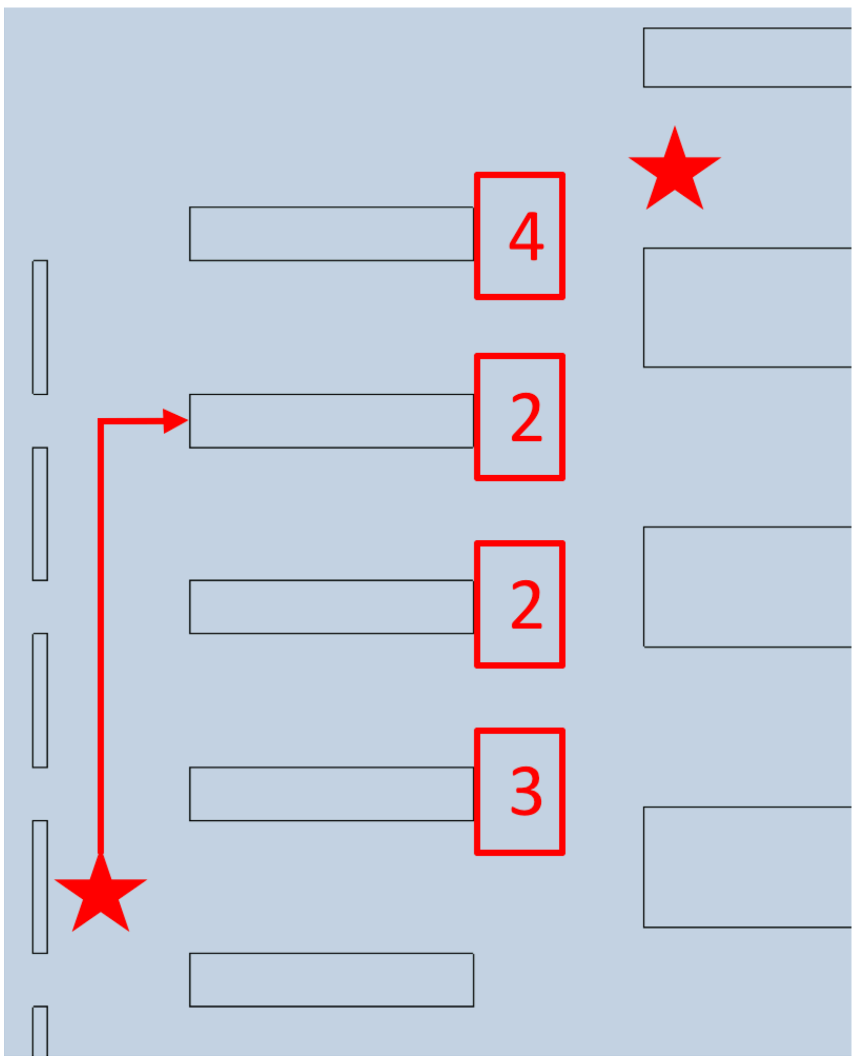



*Figure 15. H3 Flow Rack Selection*

### 6.4 H4 - Least Utilization Rack Closest to Unloading Door

This heuristic is rather the opposite to H3, the unloader selects the flow rack with the least utilization located between the load's trailer vertical position and its storage location closest to the unloading door. If the desired flow rack is full, the next logical rack is selected. Ties will always favor the flow rack closest to the loads inbound door. The same example shown for H3 is shown below in Figure 16, the 3rd flow rack would be selected because it is the one with the lowest number of pallets closest the unloading door. The data input required by H4 may be obtained by monitoring system such as the one described in Chapter 2.

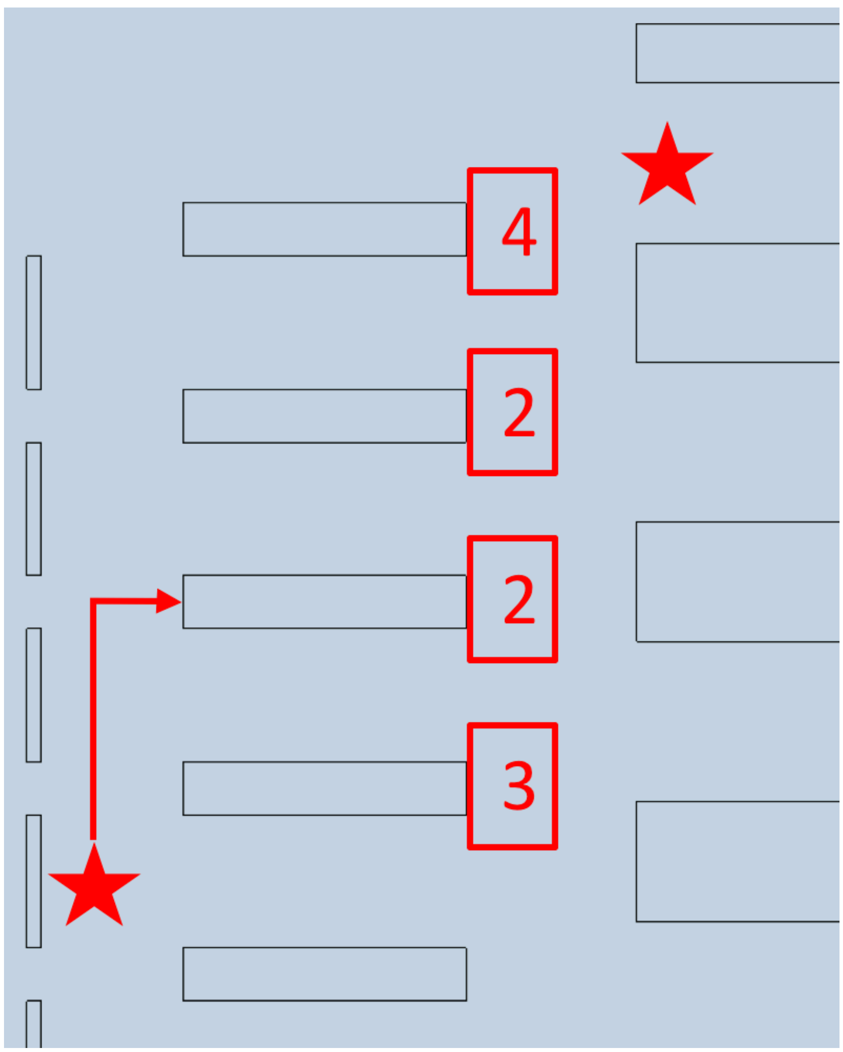



*Figure 16. H4 Flow Rack Selection*

## 7. Experimental Design

A design of experiments was implemented in order to properly compare the four rule-based heuristics. Five experimental (control) factors are considered: (1) *number of doors*, (2) *number of trailers*, (3) *trailer composition*, (4) *trailer to door assignments*, and (5) *heuristics*. The first four experimental factors and their corresponding levels are described below. The fifth factor, *heuristics*, was explained on the previous Chapter.

### 7.1 Number of Doors

The layout of the DC can be characterized by the number of (inbound dock) doors, flow racks, and storage aisles. The number of flow racks and storage aisles changes according to the number of doors. The number of doors was varied between 5, 10, and 20 doors. Table 1 shows the number of flow racks and aisles for each door number scenario, as well as the number of unloader and hauler resources available. Table 2 presents all measurements considered to generate the layouts, measurements are referenced by number in and shown in Figure 17, a layout example for the 5 door scenario.

*Table 1. Fixed Parameters per Door Scenario*

| Doors | Flow Racks | Aisles | Unloaders | Haulers |
|---|---|---|---|---|
| 5 | 6 | 5 | 2 | 3 |
| 10 | 11 | 8 | 5 | 6 |
| 20 | 21 | 15 | 10 | 12 |

*Table 2. Warehouse Dimensions*

| Dimension | Description | Length (ft) |
|---|---|---|
| 1 | Door Width | 10 |
| 2 | Between Doors | 4 |
| 3 | Door to Flow Rack | 10 |
| 4 | Aisle Width | 12 |
| 5 | Flow Rack Width | 4 |
| 6 | Storage Rack Width | 9 |
| 7 | Between Flow Racks | 10 |
| 8 | Flow Rack to Aisle | 12 |
| 9 | Flow Rack Length | 24 |
| 10 | Storage Slot Width | 4 |

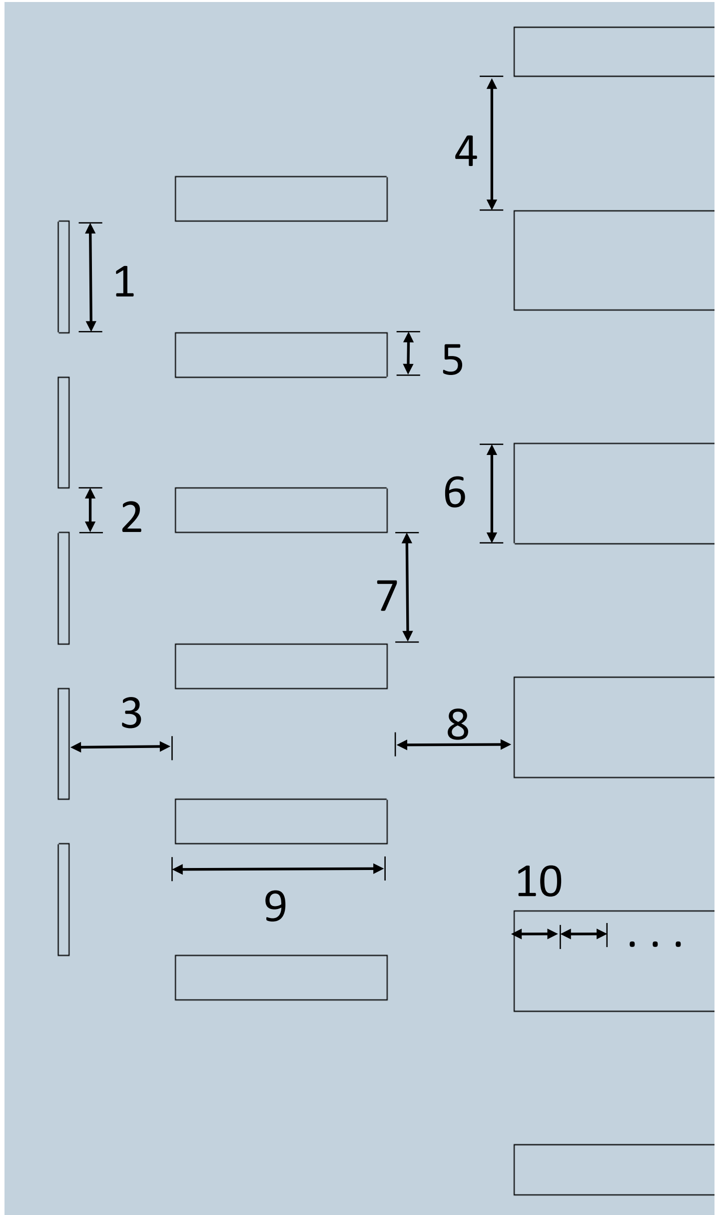



*Figure 17. Layout Dimension References*

All scenario layouts are very similar, only with variations on the number of flow racks and aisles. Rectilinear distance matrices were generated for each door number scenario adding horizontal and vertical travel distances. Table 3 presents the vertical position for the first of each type of component; the rest of the components' vertical positions was calculated adding their respective vertical spacing from Table 2. It is important to know that all vertical positions are measured at the center of each component, the center of the aisle, door, etc. The *Home* component is the initial location for all resources *Unloaders* and *Haulers* located just above the first upper flow rack. As for the *Idle Location* component, there is an idle location between all flow rack pairs, in front of each door. *Idle Locations* are used by *unloaders* on the left side, facing the doors, and *haulers* on the right side, facing the aisles.

*Table 3. Vertical Positions for Warehouse Components*

| Door Number | 5 | 10 | 20 |
|---|---|---|---|
| Vert Pos (ft) - Aisle 0 | 0 | 0 | 0 |
| Vert Pos (ft) - Home | 0 | 2.5 | 0 |
| Vert Pos (ft) - Flow Rack 0 | 5 | 3.5 | 5 |
| Vert Pos (ft) - Door 0 | 12 | 10.5 | 12 |
| Vert Pos (ft) - Idle Location 0 | 12 | 10.5 | 12 |

The remaining parameters are shown in Table 4. There will be a total of 60 possible hauler drop points per aisle, considered as *"storage locations";* 30 locations per side. Since put-away and replenishment operations are not considered, it is assumed that there are multiple storage levels. Once the haulers drop the loads in front of the storage locations, another resource such as forklift deposits the pallet in the storage racks. Another factor considered is the time to switch empty trailers, *Trailer Change-Over Time*, at the beginning of the simulation trailers are generated and the first trailer in queue per door will be considered placed at the door and available to unload. The factor *Pallet Flow Rack Drop Time* considers a 10 (simulation-) second delay for a pallet to move from one end of a flow rack to another if the flow rack is empty. A similar factor was also embedded within the programming, for the second pallet in any flow rack queue; once the first pallet in queue is picked up for hauling, if there is a pallet behind it, it will take 2 simulated seconds for it to reach the first flow rack position and be available to haul.

*Table 4. Additional DC Layout Parameters*

| | |
|---|---|
| Pallet Dimensions (inch) | 36 x 48 |
| Pallets Per Trailer | 20 |
| Storage Positions Per Aisle | 60 |
| Trailer Change-Over Time (sec) | 600 |
| Pallet Pick/Drop Time (sec) | 5 |
| Flow Rack Capacity (pallets) | 6 |
| Pallet Drop Time FR@0 (sec) | 10 |
| Unloader Speed (ft/sec) | 3 |
| Hauler Speed (ft/sec) | 3.5 |
| Turn Around Time (sec) | 3 |

### 7.2 Number of Trailers

The second factor is the number of trailers to be unloaded, which depends on the number of doors. For each door number value there will be three different number of trailers. The trailer to door ratios considered are two, three, and four times the number of doors. This variation of trailer numbers will allow us to study the heuristic performance running the scenarios at different sizes (doors) and capacity (trailers).

### 7.3 Trailer Composition

The third factor, trailer composition, has two levels with different pallet storage assignment distribution per trailer. The first level will generate a combination of three different types of trailers. For example, shown further below in Table 5, in the first scenario of 5 doors 10 trailers are generated, from which 3 will be labeled as *Pure*, 4 as *Mix,* and 3 as *High Mix* adapted from the trailer composition characterization from Bozer and Carlo (2008). Our definition of *pure, mixed*, and *highly mixed* trailers is described below.

- *Pure Trailers* – all pallets in trailer are headed to the same randomly selected storage aisle;
- *Mixed Trailers* – pallets in trailer are distributed randomly between multiple but not all aisles (see Table 5);
- *Highly Mixed Trailers* – pallets in trailer are distributed randomly between all aisles.

*Table 5. Aisle Distribution for Mixed Trailers*

| Doors | Mix Aisles |
|---|---|
| 5 | 3 |
| 10 | 5 |
| 20 | 8 |

The trailer composition factor in our design of experiments will include two levels: *combined* and *pures*. The first level (c*ombined trailers)* includes the three trailer compositions. The distribution of pures, mixed, and highly mixed trailers for each combination of the first two experimental factors is presented in Table 6. The second level of the trailer composition factor will have all pure trailers; where a single storage aisle will be randomly assigned to all pallets per trailer.

*Table 6. Trailer Composition Distribution for Combined Trailers: Pure, Mixed, High-Mix*

| Doors | Trailers | Pure | Mix | High-Mix |
|---|---|---|---|---|
| 5 | 10 | 3 | 4 | 3 |
| | 15 | 5 | 5 | 5 |
| | 20 | 6 | 7 | 7 |
| 10 | 20 | 6 | 7 | 7 |
| | 30 | 10 | 10 | 10 |
| | 40 | 13 | 14 | 13 |
| 20 | 40 | 13 | 14 | 13 |
| | 60 | 20 | 20 | 20 |
| | 80 | 26 | 27 | 27 |

Summarizing the factors and levels explained so far, we have three different warehouses with 5, 10, and 20 doors for which we will generate three different amounts of trailers (2*x*, 3*x*, and 4*x* the number of doors), and each of these with two levels for the third factor (*combined* and *pure trailers).*

**7.4 Trailer to Door Assignments**

For both types of trailer generations, each set of trailers will be assigned to unloading doors in the following manner:

- *Random* – For this method each trailer will be assigned a random door;
- *Greedy* – This second assignment will select the door that minimizes the total travel distance for all pallets in each trailer

In both methods the trailer sequence per door occurs on the same order the trailers are assigned, the assignment begins in trailer 0, 1, 2, 3, and so forth until the last trailer. It is fair to mention in both of this methods it is possible for one door to have more trailers assigned than another. For future works balancing trailers to reduce long queues in may be an option, but for the moment we will evaluate these two assignments. Table 7 presents a summary of the factors and levels considered.

*Table 7. Factors and Levels Summary*

| Factors | | Levels | | |
|---|---|---|---|---|
| 1 | Number of Doors | 5 | 10 | 20 |
| 2 | Number of Trailers | 10, 15, 20 | 20, 30, 40 | 40, 60, 80 |
| 3 | Trailer Types | *Combined, All Pure* | | |
| 4 | Trailer Door Assignments | *Random, Greedy* | | |
| 5 | Heuristics | *H1, H2, H3, H4* | | |

**7.5 Size of the Experiment**

A full factorial experiment with *200 replicas was considered.* A total of *28,800* individual runs in this experiment as detailed below.

*Doors (3) × Trailer Count (3) × Trailer Types (2) × Door Assignments (2) × Heuristics (4) × Replicas (200) = 28,800 Runs*

The variation per replica will occur mainly on the trailer generation phase, on the moment in which random aisles and storage locations are generated. Also, variations occur during the *random door assignments.* Therefore, each replica will generate a new set of trailers with a different door assignment.

For each factor-level combination (*i.e.*, problem instance) and all replicates, the following *response variables* are stored:

- Total time to unload all trailers (*i.e.*, makespan)
- Unloaders empty travel % (*i.e.*, $\frac{\sum_{u \in U} unloaded\ travel_u}{\sum_{u \in U} total\ travel_u}$)
- Average Unloaders idle time % (*i.e.*, $\sum_{u \in U} \frac{total\ idle\ time_u}{makespan} \Big/ |U|$)
- Haulers empty travel % (*i.e.*, $\frac{\sum_{h \in H} unloaded\ travel_h}{\sum_{h \in H} total\ travel_h}$)
- Average Haulers idle time % (*i.e.*, $\sum_{h \in H} \frac{total\ idle\ time_h}{makespan} \Big/ |H|$)

The first response variable is the most important one and the key indicator of the solution quality of the heuristics. The other four response variables were added as *post facto* data to gain additional insights on the performance of the heuristics. Empty travel percent could be used to study "unproductive moves". Higher empty travel percent's show that resources moved more often to idle locations and/or that they were "called" or assigned jobs form further locations. It also helps to understand how the heuristics coordinate resource actions, and how resources behavior varies through heuristics. Idle times will help determine which group of resources is the bottleneck for the operation, and how unloader-hauler workloads compare.

## 8. Simulation Programming Concept

The four rule-based heuristics were coded in Microsoft Visual Studio C# as part of a discrete-event simulation (DES). This Chapter describes the DES, henceforth referred to as "the simulation". The simulation is organized into *parameter definition*, *trailer generation*, *door assignments* and performing the *heuristics*. The main loop is the replicates, which repeats the experiment 200 times. The following loop builds a scenario for each door number. For each door number a layout is generated and within each door number another loop contains all trailer numbers. Afterwards all 4 heuristics are performed for both type of trailer sets generated, *Combined* and *All Pure,* for both door assignment levels *Random* and *Greedy.* The following list describes the order of the logic within the program.

- **Parameter Definition** – parameters are defined and the distance matrices are generated depending on the door number;
- **Trailer Generation 1** – depending on the door number the program establishes the trailer number levels and generates *Combined Trailers,* this same set of trailers is used for door assignments A and B;
- **Door Assignments A** – random door assignments are performed;
- **Heuristics** – all four heuristics solve the same set of trailers with door assignments A;
- **Door Assignment B** – greedy door assignments are performed for the same set of trailer generation 1;
- **Heuristics** – all four heuristics solve the same set of trailers with door assignments B;
- **Trailer Generation 2** – now a new set of *All Pure Trailer* is generated;
- **Door Assignments AP** – random door assignments are performed;
- **Heuristics** – all four heuristics solve the same set of trailers with door assignments AP;
- **Door Assignment BP** – greedy door assignments are performed for the same set of trailer generation 2;
- **Heuristics** – all four heuristics solve the same set of trailers with door assignments BP.

Heuristic sections coordinate *trailer to door*, *unloader* and *hauler actions* on a second to second basis. Actions take place within an incrementing loop which simulates real time in seconds. Heuristics finish once all pallets are hauled and dropped in their respective storage location. In order to manage time based actions or events, a variety of lists were generated and used as event queues. *Generic Lists* are of great convenience because they are dynamic in nature, data lines can be added, removed, and sorted. *Tuple Lists* work the same way, but are able to join multiple values in single data lines. Table 8 shows indexes used to refer to elements such as *trailers, pallet, doors*, etc.

*Table 8. Index Descriptions*

| Index | Description |
|---|---|
| *j* | Trailer |
| *i* | Pallet |
| *k* | Door |
| *f* | Flow Rack |
| *a* | Aisle |
| *s* | Storage Position |
| *u* | Unloader |
| *h* | Hauler |
| *t* | Time |

Table 9 summarizes all lists that were generated and the data values each list stores. All lists that include the variable *t*, are actions that occur at a specific time, once events are added, the list is sorted to ensure no events are skipped. On a general aspect, heuristic sections work with specific lists to determine if events occur on each time loop value. If there is no event for the current time value, the sections skips to the next, on the contrary the event is executed and deleted from queue. Let's take for example the fourth list labeled *Trailer Unloading Start Event,* this list stores information for indexes *(t, k, j, u)*. Within the programming, entry lines on this list means that on *time t*, *unloader u* will be ready in front of *door k*, to begin unloading *trailer j*. All items added to this list will be sorted through all 4 indexes from lower to higher values. To read events on all sections in the program that use lists, the program will match index by index to verify if an event occurs. This means that the section will first verify if the main time loop value matches the time value *t* on the first item or line on the list, if so there is an event pending for the specific time value, then a *k-index* loop will match the door value, once found a *j-index* loop is accessed and in a similar manner it will match the trailer value, afterwards a u-*index* loop is accessed matching the unloader value. Once all values are matched, all index values *(t, k, j, u)* are stored on the multiple loop counter variables and used to execute instructions or event actions. After the event is executed, the entry line representing the event just accessed will be deleted, removing the entry from queue. Afterwards, the section re-loop on the same time to evaluate the next item on cue, if the current time value matches the time value on the list, the section will continue to match the reaming indexes and carry out the event. If the time value does not match, it means that all events on the specific time value were attended, the program continues to the following section until all sections are attended and the list is re-accessed in the following time value. This is general method used to carry out different events through the program. Further below all program sections will be explained.

*Table 9. Rule-Based Heuristics Programming Event Lists*

| List | Index |
|---|---|
| Storage Locations | *j,i,a,s* |
| Trailer Door Assignments | *j,k* |
| Trailer Available To Unload Event | *t,j,k* |
| Trailer Unloading Start Event | *t,k,j,u* |
| Unloader Ready For Next Pallet Event | *t,k,j,u* |
| Unloader Pallet Pick Event | *t,k,j,u,i* |
| Unloader Pallet Drop Event | *t,k,j,u,i,f* |
| Unloader Ready At Idle Location Event | *t,u,f* |
| Pallet Ready To Haul Event | *t,j,i,f* |
| Hauler Ready For Next Pallet Event | *t,h,a* |
| Hauler Pallet Pick Event | *t,j,i,f,h* |
| Hauler Ready At Idle Location Event | *t,h,f* |
| Hauler Pallet Drop Event | *t,j,i,h,a,s* |
| Flow Rack Updates Event | *t,f* |

The following list present main heuristic sections, a brief description, and the data list each section interacts with.

- Trailer Door Actions – this first section loops through all door values, for each door it verifies if there is a trailer to be assigned at this door in the *Trailer Door Assignment* list; if so and the door is free, it will add a new entry with *time, trailer,* and *door* values in the *Trailer Available To Unload Event* list.
- Verify Unloader Idle Location – This sections reads values from the *Unloader Ready At Idle Location Event* list to determine if an unloader has arrived at any idle location, if so it declares the unloader idle, and marks his current idle location and the list line is deleted.

- Unloader Trailer Assignments – This sections reads from the list *Trailer Available To Unload* to determine if trailers are available at doors for unloading. For each trailer at any door, it loops through all unloaders and assigns the closest unloader available. It calculates the travel time it will take for the assigned unloader to reach the door, and add a new entry to the *Trailer Unloading Start Event* list. If there are no unloaders available to assign, it will re-add the same entry to the *Trailer Available To Unload* on the following time value (t+1), this will continue to happen until an unloader is available to assign.
- First Pallet Unloadings – This section reads from the *Trailer Unloading Start Event* list to execute the first unloading of the first pallet of each trailer. At this moment the unloader arrived at the assigned door and is ready to perform the first unloading. If there is an event on the current time a new entry is added to the *Unloader Pallet Pick Event* list considering the time to entry, load the pallet, and exit the trailer.
- Next Pallet Unloadings – This section is similar to the previous, only that it handles the following unloadings, remember that each trailer is assigned to only one unloader, this section schedules all unloadings after the first. It reads from the *Unloader Ready For Next Pallet Event* list, meaning that the unloader has returned to the door after dropping the pallet at the flow rack. It will also add a new entry to the *Unloader Pallet Pick Event* list considering the time to remove the pallet.
- Pallet Picks – This event precedes both previous sections, *first* and *next pallet unloadings.* At this point the unloader has exited the trailer and has to make the decision to select a flow rack. **This is the decision point that varies between all four heuristics.** An example of the first heuristic, the unloader will select the closest flow rack on the way to the storage aisle. The program calculates the travel time and add a new entry the *Unloader Pallet Drop Event* list, meaning that the pallet was dropped on the selected flow rack.
- Flow Rack Drop Updates – Following the previous section, this section verifies if there is a pending item in the *Unloader Pallet Drop Event* list. It updates the pallets' position in the flow rack queue, updates the flow rack load, and adds a new entry to the *Pallet Ready To Haul* list. Meaning that the pallet has reached the opposite end of the flow rack. As stated earlier, if there are no pallets on the flow rack, it takes 10 seconds for the pallet to reach the opposite end, and be available for hauling. Afterwards, it will add a new line to the *Unloader Ready For Next Event*; if the pallet dropped was the last in the trailer, it will give a temporary rest to the unloader

by sending the unloader to the closest idle location. Meaning a new entry will be added to the *Unloader Ready At Idle Location List* considering 5 seconds for the unloader to arrive.

- Verify Hauler Idle Location – This section verifies if a hauler has arrived any idle location. It reads from the list *Hauler Ready At Idle Location Event*. If an action occurs at the current time, the hauler will be declared idle, the idle location will be marked and the accessed entry will be remove from the list.
- Verify Hauler Ready For Next Pallet Location – This section verifies if a hauler has returned to the entry point of the aisle after performing a drop, making the hauler available to be assigned another pallet. It reads from the list *Hauler Ready For Next Pallet Event* list. If an action occurs at the current time the hauler will be declared ready for next pallet and the location will be marked to consider the hauler for pallet assignments.
- Hauler Assignments – This section reads from the list *Pallet Ready To Haul,* if an action occurs at the moment, the program evaluates all available hauler positions, *home, idle,* or *ready for next* to select the closest hauler to the flow rack location of the pallet considered to be assigned. If no haulers are available to assign, a re-entry for the pallet will be added on the *Pallet Ready To Haul* list for next time (t+1). Once a hauler is selected, a new entry will be added to *Hauler Pallet Pick Event* list considering hauler travel time to the reach and pick up the pallet.
- Move Haulers From Next To Idle Location – This is a sub-section of the previous which verifies after the hauler assignments if a hauler remains at the entry point of the aisle, or the ready for next location. If so, it will send the hauler to the closest idle location, calculate the travel time and add an entry to the list *Hauler Ready At Idle Location Event*.
- Hauler Pick Updates – This section reads from the list *Hauler Pallet Pick Event*. At this point the hauler has a pallet loaded and ready to move to the storage position. The program reads from the list *Storage Locations* to obtain the pallets' aisle and storage position (*a, s*) to calculate travel time and add a new entry to the list *Hauler Drop Event.* Finally, the flow rack load value is updated, if the remaining flow rack load is not 0, meaning that there are more pallets in the flow rack after this pick, the position queues for the remaining pallets has to be updated, for this a new entry is added to the list *Flow Rack Pick Updates Event.*
- Flow Rack Pick Updates – This section reads if an action was triggered in the list *Flow Rack Pick Updates Event.* If so, pallet position queues are subtracted 1. Bringing forward in the flow

rack one position all reaming pallets. For the pallet position queue 2 to reach position 1, a 2 second delay is considered, and an entry is made for this pallet in the *Pallet Ready To Haul* list.

- Hauler Drop Event – This final section reads from the *Hauler Pallet Drop Event* list. If an action occurs at the current time the pallet is declared stored and the *Pallets Remaining Count* is subtracted 1 value, the total Pallets Remaining value is calculated at the beginning of each run. Once the *Pallets Remaining Count* reaches 0, meaning all pallets were hauled, the heuristic ends. If the heuristic continues, the final step for this section is to add a new entry on the list *Hauler Ready For Next Pallet*. For this it will calculate the travel time for the hauler to reach the entry point of the current aisle, making the hauler available temporarily for a new pallet assignment.

## 9. Experimental Results

This Chapter presents the statistical analysis and results summary. The discrete-event simulation coded in C# explained in the previous Chapter exports to a text file all factor levels and response variables results for each run. It took approximately 2 hours to run all 28,800 instances. The results presented were obtained using *Minitab Statistical Software* using the General Linear Model (GLM), an ANOVA procedure used for analyzing data from multiple different experiments such as this case. The purpose is to determine if there is a statistical difference in population means for our response variables. Also, to determine the effect of factor interactions, depending whether the factors are crossed (identical on all levels) or nested (differ on a specific level); for this case the only nested factor would be the *number of trailers*, because the number of trailers are not identical for all door numbers. Therefore, the second factor *number of trailers* is nested in the first factor *number of doors*. For the GLM nested factor interactions are excluded. Therefore, any combination that contains *Doors*Trailers* factors will not be considered in the ANOVA. Table 10 summarizes makespan results by averaging the 200 replicas for each instance. Notice that the main factor levels (*number of door and heuristics*) are presented as columns. The remaining factors are presented in the rows of the tables as a three-digit number where each digit represents a factor level. The first digit represents the number of trailers' factor, the second the type of trailer, and the third the door assignment. For example, 111 is the instance where factors 2, 3, and 4 are at the first level.

*Table 10. Experimental Results: Average Makespan (hrs) Over 200 Replicates*

| | 5 Doors | | | | 10 Doors | | | | 20 Doors | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | H1 | H2 | H3 | H4 | H1 | H2 | H3 | H4 | H1 | H2 | H3 | H4 |
| 111 | **1.54** | 1.84 | 1.82 | 1.57 | **1.84** | 2.50 | 2.46 | 1.89 | **2.38** | 3.99 | 3.98 | 2.44 |
| 112 | **1.49** | 1.61 | 1.60 | 1.51 | **2.10** | 2.49 | 2.46 | **2.10** | **2.56** | 3.58 | 3.54 | 2.59 |
| 121 | **1.52** | 1.83 | 1.81 | 1.55 | **1.83** | 2.47 | 2.43 | 1.87 | **2.36** | 3.96 | 3.96 | 2.41 |
| 122 | **1.73** | **1.73** | **1.73** | **1.73** | **1.84** | **1.84** | **1.84** | **1.84** | **2.49** | **2.49** | **2.49** | **2.49** |
| 211 | **2.27** | 2.68 | 2.64 | 2.31 | **2.69** | 3.46 | 3.39 | 2.70 | **3.43** | 5.45 | 5.45 | 3.46 |
| 212 | **2.22** | 2.37 | 2.34 | 2.24 | 3.14 | 3.62 | 3.62 | **3.11** | **3.85** | 5.25 | 5.23 | 3.86 |
| 221 | **2.25** | 2.64 | 2.59 | 2.28 | **2.69** | 3.44 | 3.37 | 2.72 | **3.42** | 5.50 | 5.51 | 3.46 |
| 222 | **2.52** | **2.52** | 2.53 | **2.52** | 2.50 | **2.49** | 2.50 | 2.50 | **3.41** | **3.41** | **3.41** | **3.41** |
| 311 | **3.01** | 3.50 | 3.45 | 3.03 | **3.46** | 4.26 | 4.14 | 3.48 | **4.52** | 6.96 | 6.95 | **4.52** |
| 312 | **2.96** | 3.14 | 3.11 | 2.98 | 4.15 | 4.77 | 4.80 | **4.14** | 4.98 | 6.66 | 6.64 | **4.95** |
| 321 | **3.04** | 3.55 | 3.49 | 3.07 | **3.44** | 4.30 | 4.20 | 3.47 | **4.50** | 7.07 | 7.06 | 4.57 |
| 322 | 3.36 | **3.35** | 3.37 | 3.37 | 3.20 | **3.19** | 3.20 | 3.20 | 4.41 | 4.41 | 4.41 | **4.40** |

### 9.1 Sample Size Analysis

The initial number of replicates for each experimental run was set for 200. Therefore, there will be a total of 200 samples for each instance or combination of factors. To verify that the sample size is sufficient, and no additional runs are required, the sample size for each instance was calculated with a significance level, $\alpha = 0.01$ and relative error $e_r = 0.05$. Table 11 shows the average and standard deviation for the makespan in hours, for the results of the initial 200 runs. The largest sample size (*N*) was obtained for instance 2-1-1-2-3 for a total of 149 samples, therefore we can conclude that the initial 200 replicates will suffice for a dependable statistical analysis.

*Table 11. Sample Size per Instance*

| Instance | Count of Makespan (hrs) | Average of Makespan (hrs) | StdDev of Makespan (hrs) | N (α = 0.01) |
|---|---|---|---|---|
| 11111 | 200 | 1.537 | 0.234 | 61 |
| 11112 | 200 | 1.837 | 0.311 | 76 |
| 11113 | 200 | 1.822 | 0.313 | 78 |
| 11114 | 200 | 1.573 | 0.246 | 65 |
| 11121 | 200 | 1.487 | 0.229 | 63 |
| 11122 | 200 | 1.607 | 0.264 | 72 |
| 11123 | 200 | 1.596 | 0.265 | 73 |
| 11124 | 200 | 1.514 | 0.229 | 61 |
| 11211 | 200 | 1.519 | 0.224 | 58 |
| 11212 | 200 | 1.831 | 0.319 | 80 |
| 11213 | 200 | 1.808 | 0.321 | 84 |
| 11214 | 200 | 1.547 | 0.239 | 63 |
| 11221 | 200 | 1.731 | 0.398 | 140 |
| 11222 | 200 | 1.731 | 0.400 | 142 |
| 11223 | 200 | 1.732 | 0.399 | 141 |
| 11224 | 200 | 1.732 | 0.398 | 140 |
| 12111 | 200 | 2.268 | 0.267 | 37 |
| 12112 | 200 | 2.679 | 0.395 | 58 |
| 12113 | 200 | 2.643 | 0.399 | 60 |
| 12114 | 200 | 2.305 | 0.285 | 41 |
| 12121 | 200 | 2.219 | 0.329 | 59 |
| 12122 | 200 | 2.368 | 0.390 | 72 |
| 12123 | 200 | 2.344 | 0.393 | 75 |
| 12124 | 200 | 2.241 | 0.335 | 59 |

| | | | | |
|---|---|---|---|---|
| 12211 | 200 | 2.251 | 0.263 | 36 |
| 12212 | 200 | 2.644 | 0.411 | 64 |
| 12213 | 200 | 2.592 | 0.417 | 69 |
| 12214 | 200 | 2.278 | 0.285 | 42 |
| 12221 | 200 | 2.523 | 0.519 | 112 |
| 12222 | 200 | 2.521 | 0.523 | 114 |
| 12223 | 200 | 2.525 | 0.519 | 112 |
| 12224 | 200 | 2.524 | 0.517 | 112 |
| 13111 | 200 | 3.012 | 0.339 | 34 |
| 13112 | 200 | 3.502 | 0.466 | 47 |
| 13113 | 200 | 3.449 | 0.471 | 49 |
| 13114 | 200 | 3.026 | 0.360 | 38 |
| 13121 | 200 | 2.961 | 0.368 | 41 |
| 13122 | 200 | 3.143 | 0.431 | 50 |
| 13123 | 200 | 3.107 | 0.436 | 52 |
| 13124 | 200 | 2.976 | 0.370 | 41 |
| 13211 | 200 | 3.042 | 0.357 | 36 |
| 13212 | 200 | 3.553 | 0.459 | 44 |
| 13213 | 200 | 3.489 | 0.476 | 49 |
| 13214 | 200 | 3.070 | 0.361 | 37 |
| 13221 | 200 | 3.361 | 0.604 | 86 |
| 13222 | 200 | 3.355 | 0.608 | 87 |
| 13223 | 200 | 3.365 | 0.602 | 85 |
| 13224 | 200 | 3.365 | 0.602 | 85 |
| 21111 | 200 | 1.845 | 0.290 | 66 |
| 21112 | 200 | 2.495 | 0.518 | 114 |
| 21113 | 200 | 2.463 | 0.524 | 120 |
| 21114 | 200 | 1.891 | 0.307 | 70 |
| 21121 | 200 | 2.097 | 0.448 | 121 |
| 21122 | 200 | 2.489 | 0.577 | 143 |
| **21123** | **200** | **2.462** | **0.584** | **149** |
| 21124 | 200 | 2.097 | 0.436 | 115 |
| 21211 | 200 | 1.832 | 0.290 | 67 |
| 21212 | 200 | 2.475 | 0.524 | 119 |
| 21213 | 200 | 2.429 | 0.536 | 129 |
| 21214 | 200 | 1.872 | 0.299 | 68 |
| 21221 | 200 | 1.841 | 0.371 | 108 |
| 21222 | 200 | 1.843 | 0.373 | 109 |
| 21223 | 200 | 1.844 | 0.372 | 108 |
| 21224 | 200 | 1.843 | 0.372 | 108 |

| | | | | |
|---|---|---|---|---|
| 22111 | 200 | 2.693 | 0.300 | 33 |
| 22112 | 200 | 3.465 | 0.537 | 64 |
| 22113 | 200 | 3.390 | 0.542 | 68 |
| 22114 | 200 | 2.699 | 0.328 | 39 |
| 22121 | 200 | 3.138 | 0.585 | 92 |
| 22122 | 200 | 3.617 | 0.783 | 125 |
| 22123 | 200 | 3.624 | 0.793 | 127 |
| 22124 | 200 | 3.112 | 0.581 | 92 |
| 22211 | 200 | 2.695 | 0.328 | 39 |
| 22212 | 200 | 3.438 | 0.642 | 93 |
| 22213 | 200 | 3.367 | 0.667 | 104 |
| 22214 | 200 | 2.724 | 0.353 | 44 |
| 22221 | 200 | 2.499 | 0.360 | 55 |
| 22222 | 200 | 2.492 | 0.364 | 57 |
| 22223 | 200 | 2.497 | 0.360 | 55 |
| 22224 | 200 | 2.498 | 0.360 | 55 |
| 23111 | 200 | 3.460 | 0.385 | 33 |
| 23112 | 200 | 4.263 | 0.680 | 68 |
| 23113 | 200 | 4.145 | 0.692 | 74 |
| 23114 | 200 | 3.482 | 0.394 | 34 |
| 23121 | 200 | 4.155 | 0.695 | 74 |
| 23122 | 200 | 4.769 | 0.932 | 101 |
| 23123 | 200 | 4.801 | 0.936 | 101 |
| 23124 | 200 | 4.137 | 0.700 | 76 |
| 23211 | 200 | 3.443 | 0.361 | 29 |
| 23212 | 200 | 4.303 | 0.654 | 61 |
| 23213 | 200 | 4.198 | 0.681 | 70 |
| 23214 | 200 | 3.473 | 0.372 | 30 |
| 23221 | 200 | 3.204 | 0.444 | 51 |
| 23222 | 200 | 3.188 | 0.448 | 52 |
| 23223 | 200 | 3.199 | 0.442 | 51 |
| 23224 | 200 | 3.203 | 0.443 | 51 |
| 31111 | 200 | 2.377 | 0.320 | 48 |
| 31112 | 200 | 3.990 | 0.818 | 111 |
| 31113 | 200 | 3.982 | 0.824 | 114 |
| 31114 | 200 | 2.443 | 0.328 | 48 |
| 31121 | 200 | 2.558 | 0.421 | 72 |
| 31122 | 200 | 3.584 | 0.674 | 94 |
| 31123 | 200 | 3.541 | 0.686 | 100 |
| 31124 | 200 | 2.590 | 0.393 | 61 |

| | | | | |
|---|---|---|---|---|
| 31211 | 200 | 2.361 | 0.299 | 43 |
| 31212 | 200 | 3.963 | 0.828 | 116 |
| 31213 | 200 | 3.957 | 0.817 | 113 |
| 31214 | 200 | 2.410 | 0.323 | 48 |
| 31221 | 200 | 2.486 | 0.520 | 116 |
| 31222 | 200 | 2.487 | 0.521 | 116 |
| 31223 | 200 | 2.490 | 0.520 | 116 |
| 31224 | 200 | 2.489 | 0.518 | 115 |
| 32111 | 200 | 3.427 | 0.344 | 27 |
| 32112 | 200 | 5.445 | 0.913 | 75 |
| 32113 | 200 | 5.448 | 0.913 | 75 |
| 32114 | 200 | 3.464 | 0.368 | 30 |
| 32121 | 200 | 3.854 | 0.665 | 79 |
| 32122 | 200 | 5.249 | 1.074 | 111 |
| 32123 | 200 | 5.234 | 1.101 | 117 |
| 32124 | 200 | 3.859 | 0.661 | 78 |
| 32211 | 200 | 3.424 | 0.331 | 25 |
| 32212 | 200 | 5.497 | 0.905 | 72 |
| 32213 | 200 | 5.513 | 0.916 | 73 |
| 32214 | 200 | 3.462 | 0.333 | 25 |
| 32221 | 200 | 3.412 | 0.653 | 97 |
| 32222 | 200 | 3.406 | 0.656 | 98 |
| 32223 | 200 | 3.410 | 0.654 | 97 |
| 32224 | 200 | 3.410 | 0.651 | 97 |
| 33111 | 200 | 4.515 | 0.395 | 20 |
| 33112 | 200 | 6.961 | 1.119 | 69 |
| 33113 | 200 | 6.947 | 1.128 | 70 |
| 33114 | 200 | 4.517 | 0.427 | 24 |
| 33121 | 200 | 4.979 | 0.724 | 56 |
| 33122 | 200 | 6.661 | 1.187 | 84 |
| 33123 | 200 | 6.645 | 1.215 | 89 |
| 33124 | 200 | 4.951 | 0.698 | 53 |
| 33211 | 200 | 4.502 | 0.458 | 27 |
| 33212 | 200 | 7.070 | 1.247 | 83 |
| 33213 | 200 | 7.062 | 1.256 | 84 |
| 33214 | 200 | 4.566 | 0.488 | 30 |
| 33221 | 200 | 4.410 | 0.826 | 93 |
| 33222 | 200 | 4.407 | 0.832 | 95 |
| 33223 | 200 | 4.410 | 0.826 | 93 |
| 33224 | 200 | 4.402 | 0.829 | 94 |

| | |
|---|---|
| Total | 28800 |

## 9.2 Results for Response Variable 1: Makespan

The following table presents the ANOVA results for the first response variable, *makespan*, the total unloading time expressed in hours. Table 12 contains individual factors and order 2 interactions; higher interactions were not considered; our main interest is to see how factors interact independently. As explained earlier, the interaction for the nested factors *Doors*Trailers* was excluded. For this test the null hypothesis states that all population means are equal and the alternate hypothesis states that at least one pair of means is different. To determine this, we compare the *p-value* to our significance level, $\alpha = 0.01$. If the *p-value* is less than our significance level, we can state there is enough evidence to reject the null hypothesis, and conclude that not all population means are equal. If the *p-value* is greater than our significance level, there is not enough evidence to reject the null hypothesis, therefore the difference in means is not statistically significant.

*Table 12. ANOVA: Response Variable Makespan*

| Source | DF | Adj SS | Adj MS | F-Value | P-Value |
|---|---|---|---|---|---|
| Door Number | 2 | 7.004 | 3.502 | 23661.08 | 0.000 |
| Trailer Number | 2 | 9.863 | 4.931 | 33316.32 | 0.000 |
| Trailer Composition | 1 | 0.189 | 0.189 | 1277.88 | 0.000 |
| Door Assignment | 1 | 0.135 | 0.134 | 908.64 | 0.000 |
| Heuristic | 3 | 1.193 | 0.398 | 2685.79 | 0.000 |
| Door Number*Trailer Composition | 2 | 0.252 | 0.126 | 852.25 | 0.000 |
| Door Number*Door Assignment | 2 | 0.054 | 0.027 | 182.18 | 0.000 |
| Door Number*Heuristic | 6 | 0.280 | 0.047 | 314.80 | 0.000 |
| Trailer Number*Trailer Composition | 2 | 0.004 | 0.002 | 13.20 | 0.000 |
| Trailer Number*Door Assignment | 2 | 0.002 | 0.001 | 5.44 | 0.004 |
| Trailer Number*Heuristic | 6 | 0.012 | 0.002 | 13.54 | 0.000 |
| Trailer Composition*Door Assignment | 1 | 0.178 | 0.178 | 1203.35 | 0.000 |
| Trailer Composition*Heuristic | 3 | 0.062 | 0.021 | 139.68 | 0.000 |
| Door Assignment*Heuristic | 3 | 0.350 | 0.117 | 788.14 | 0.000 |
| Error | 28763 | 4.257 | 0.000 | | |
| Lack-of-Fit | 107 | 0.489 | 0.005 | 34.73 | 0.000 |
| Pure Error | 28656 | 3.769 | 0.000 | | |
| Total | 28799 | 23.834 | | | |

Reviewing Table 12, ANOVA results for *makespan,* all factors and 2-way interactions have a *p-value* less than our significance level 0.01, meaning that there is enough evidence to reject the null hypothesis that all population means are equal. Therefore, at least some population means differ, to determine which factors differ mean pairwise comparisons will be presented further on this Section.

The following step is to verify the validity of the model by observing the residual plots to determine any signs of non-normality or non-constant variance. The Normal Probability Plot in Figure 18 presents a graphical output of the normal probabilities vs residual data. Normality can be confirmed in a probability plot by observing a 45-degree line. If signs of non-normality are present, typically the response data is transformed using a Box-Cox transformation. The original results for the response variable *Makespan* did show signs of non-normality, for which the data was transformed. Figure 18 shows the results after the transformation. Box-Cox transformation require the selection of a λ parameter, Minitab GLM realizes such transformation with an optimal λ parameter, in this case λ = -0.07797.

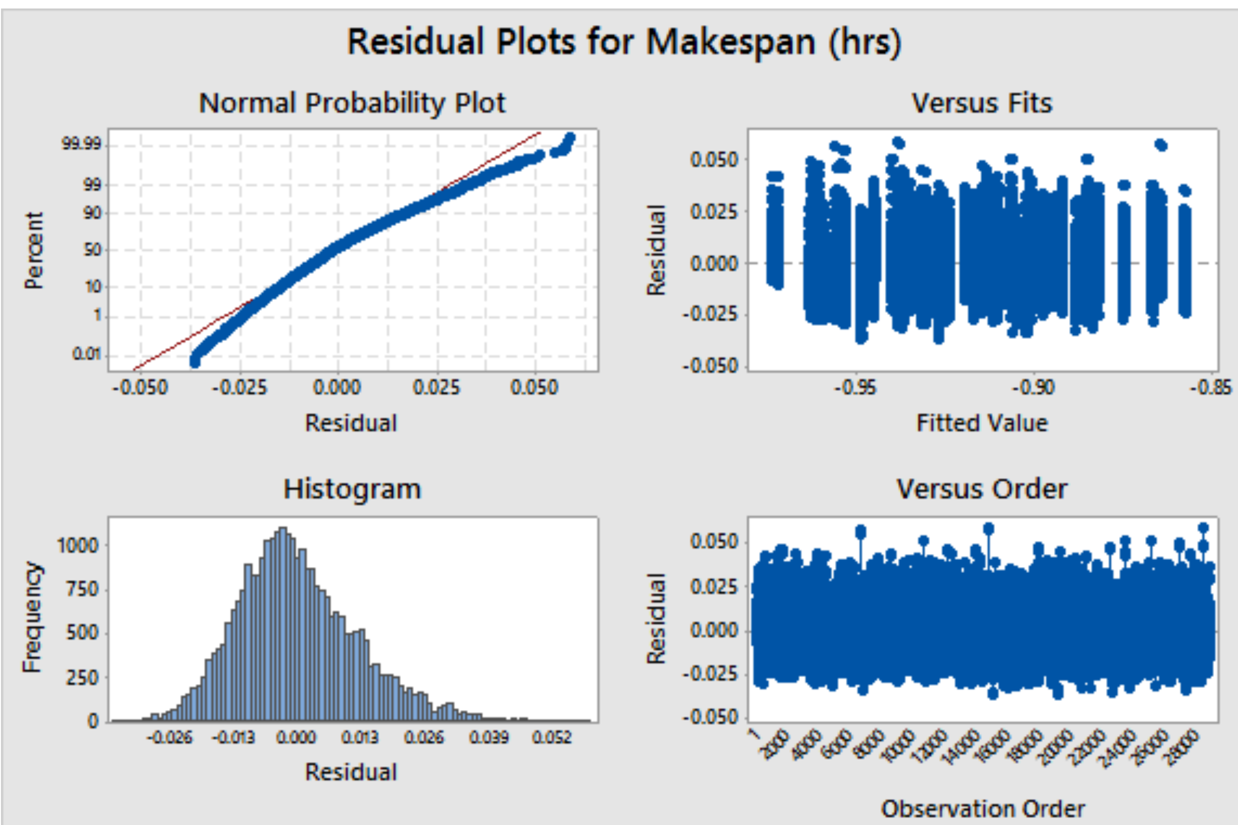


*Figure 18. Residual Plots: Response Variable Makespan*

Versus Fits and Versus Order are used to identify signs of non-constant variance, which would be a specific patter or tendency. In this case we can see there is no pattern or tendency, which would indicate that there is no violation to the constant variance assumption. Besides the residual plot analysis, the *R-sq* value is also a factor to be considered for the validity of the model. The resulting *R-sq* for the GLM is 82.14% as shown in Table 13. This means that 82.14% of the variability in the model is explained by the factors and interactions considered. This value would increase if we had considered order 3 and higher interactions in the model, but for the current purpose the results are acceptable.

*Table 13. Model Summary Response Variable Makespan*

| Model Summary for Transformed Response | | | |
|---|---|---|---|
| *s* | R-sq | R-sq(adj) | R-sq(pred) |
| 0.01217 | 82.14% | 82.12% | 82.09% |

The ANOVA results indicated that all factors, and interactions means are not equal for the response variable *Makespan*, but it does not indicate which groups are equal or differ. The following tables present result for the Tukey Method, a statistical test used to determine if group means differ. For all main factors except the fifth factor *Heuristic*, we obtained separate groupings, this means that for all *Door Number*, *Trailer Number*, *Trailer Type,* and *Door Assignment* levels, we obtained statistically different means with 99% confidence, a significance level, $\alpha = 0.01$. The test presents groups with different letters, Table 14 below presents the results for the fifth factor *Heuristic* which indicates that there is no significant difference in means for makespan between the heuristics 1 and 4, 2 and 3. Confidence interval plots for heuristics are shown below in Figure 19.

*Table 14. Tukey Pairwise Comparisons Response Variable Makespan Factor Heuristic*

| Heuristic | Count of Makespan (hrs) | Average of Makespan (hrs) | StdDev of Makespan (hrs) | Grouping |
|---|---|---|---|---|
| 1 | 7200 | 2.864 | 1.01 | A |
| 4 | 7200 | 2.882 | 1.005 | A |
| 3 | 7200 | 3.543 | 1.589 | B |
| 2 | 7200 | 3.565 | 1.584 | B |

For all order 2 interactions with the factor *Heuristics* we obtained the same grouping as in Table 14, this means that the heuristics behave in the same manner regardless of the other factors *Doors, Trailers, Trailer Composition and Door Assignment.*

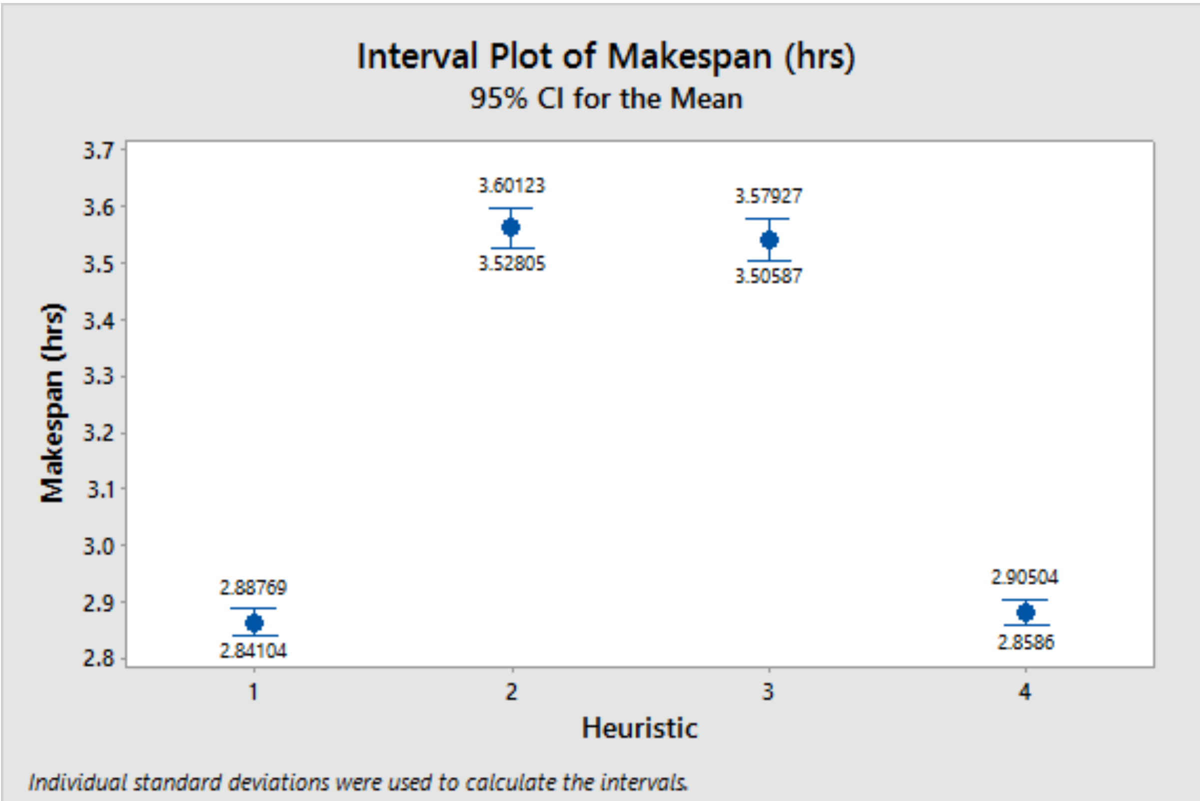



*Figure 19. Makespan Confidence Interval Plots for Heuristics*

Makespan results in terms of the first factor *Door Number* are summarized below in Table 15, followed by the confidence interval plots shown in Figure 20. For the factor *Door Number* there is a significant difference in makespan results due to the size of the experiments.

*Table 15. Makespan Results Summary for Factor Door Number*

| Door Number | Count of Makespan (hrs) | Average of Makespan (hrs) | StdDev of Makespan (hrs) |
|---|---|---|---|
| 5 | 9600 | 2.444 | 0.772 |
| 10 | 9600 | 2.983 | 0.979 |
| 20 | 9600 | 4.213 | 1.563 |

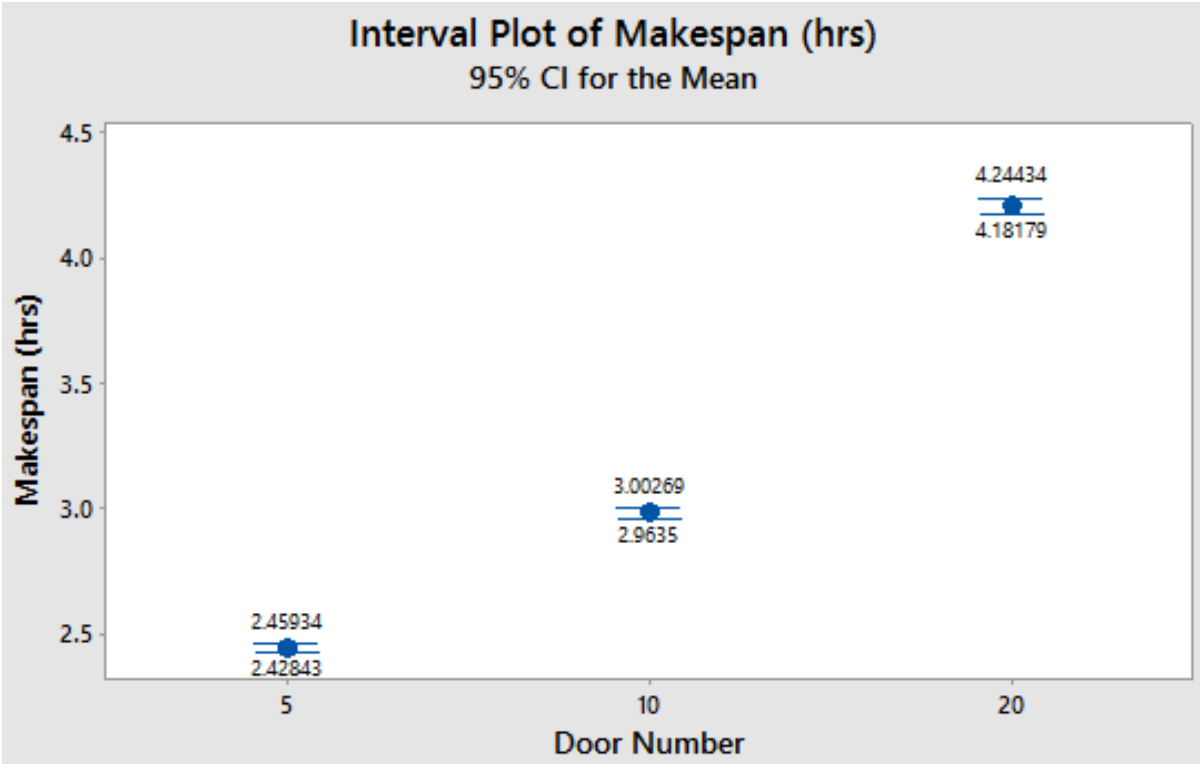



*Figure 20. Makespan Confidence Interval Plots for Door Number*

A combined boxplot for main factors Door Number and Heuristics is shown in Figure 21, from which we can observer the general behavior for all heuristics; H1 and H4 outperform H2 and H3 in all door number scenarios. The gap between heuristic pairs increases while door number increases as well, meaning that in larger unloading scenarios H1 and H4 perform much better than H2 and H4.

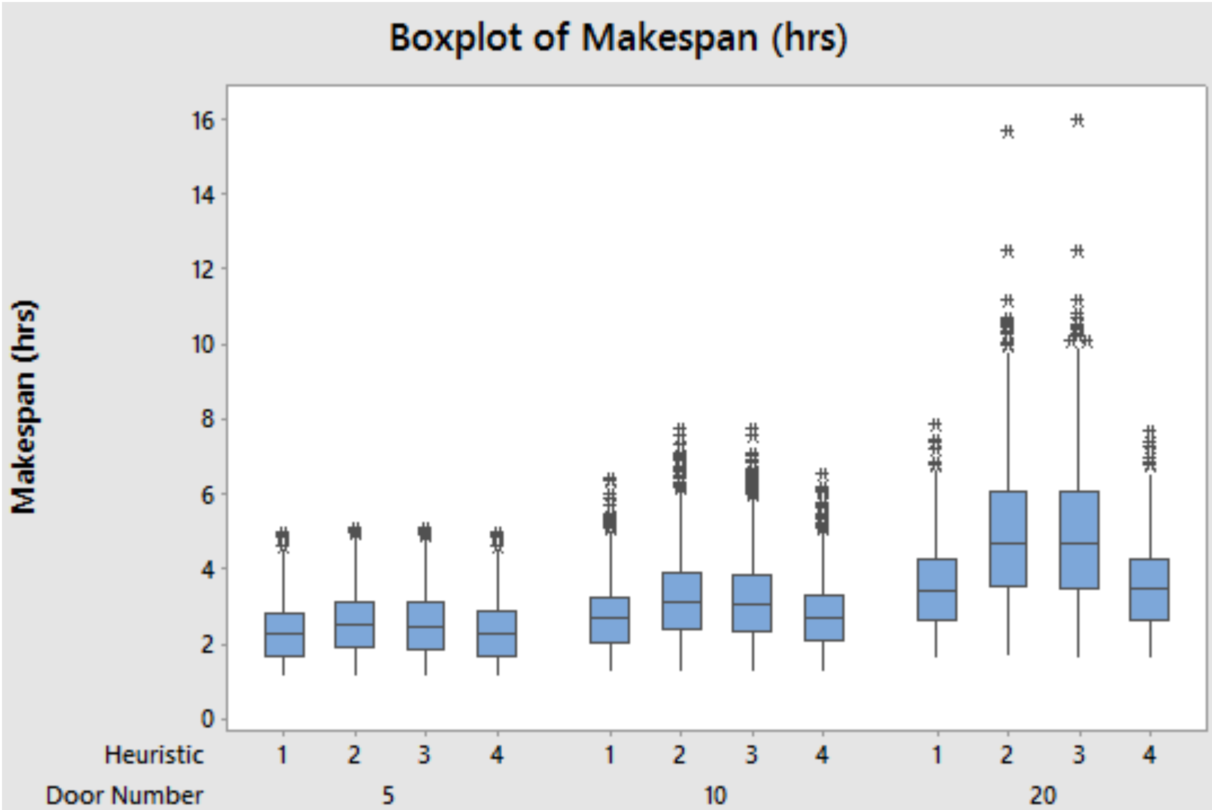



*Figure 21. Makespan Combined Boxplot for Factors Door Number-Heuristics*

Summarizing our results, in Figure 22 we can see an overview of the main factors mean comparison. As expected a higher number of doors and trailers increases makespan, and pure trailers and greedy door assignments reduce makespan. As for the heuristics, we can state that the best performing heuristics with lower makespan are (H1) in which unloaders select the flow rack closest to the unloading door on the way to storage aisle, and (H4) in which the unloaders select the flow rack with least utilization closest to the unloading door the way to the storage aisle.

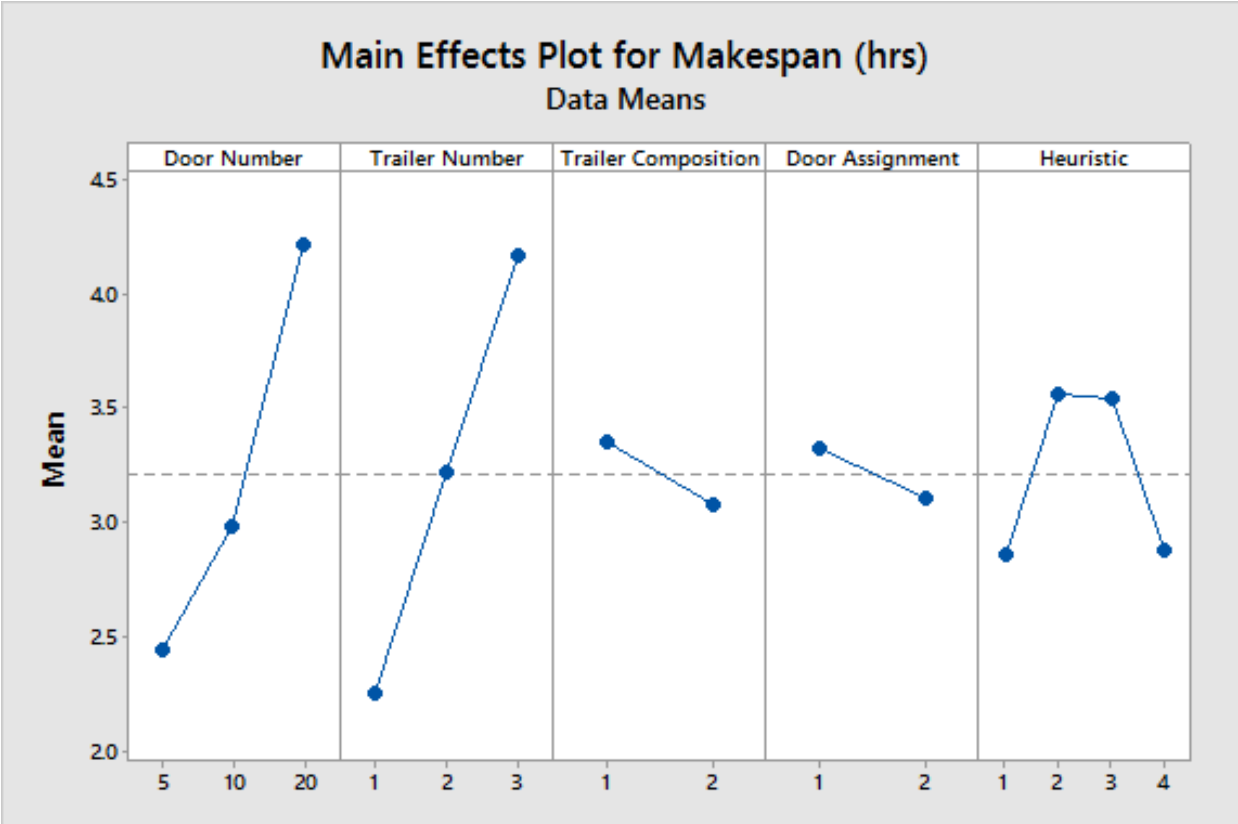



Figure 22. Main Effects Plot Response Variable Makespan

### 9.3 Heuristics Performance Metrics Analysis

In terms of makespan, Section 9.2 concluded that the best performing heuristics are H1 and H4. Tables 16– 18 show all five response variable results per heuristic for each door number scenario.

*Table 16. Response Variable Summary - 5 Doors*

| 5 Doors | H1 | H2 | H3 | H4 |
|---|---|---|---|---|
| Average of Makespan (hrs) | **2.33** | 2.56 | 2.54 | 2.35 |
| Average of Unloader Empty Travel % | 50.73 | 50.41 | 50.39 | 50.58 |
| Average of Hauler Empty Travel% | 51.52 | 55.14 | 54.14 | 52.15 |
| Average of Unloader Idle Time % | 27.07 | 25.38 | 25.50 | 26.32 |
| Average of Hauler Idle Time % | 16.19 | 29.74 | 27.64 | 17.99 |

*Table 17. Response Variable Summary - 10 Doors*

| 10 Doors | H1 | H2 | H3 | H4 |
|---|---|---|---|---|
| Average of Makespan (hrs) | **2.74** | 3.24 | 3.20 | 2.75 |
| Average of Unloader Empty Travel % | 51.07 | 50.52 | 50.55 | 50.89 |
| Average of Hauler Empty Travel% | 53.58 | 58.67 | 57.38 | 54.48 |

| | | | | |
|---|---|---|---|---|
| Average of Unloader Idle Time % | 38.85 | 39.77 | 40.00 | 38.80 |
| Average of Hauler Idle Time % | 21.13 | 40.27 | 38.21 | 23.01 |

*Table 18. Response Variable Summary - 20 Doors*

| 20 Doors | H1 | H2 | H3 | H4 |
|---|---|---|---|---|
| Average of Makespan (hrs) | **3.53** | 4.89 | 4.89 | 3.55 |
| Average of Unloader Empty Travel % | 51.56 | 50.68 | 50.69 | 51.22 |
| Average of Hauler Empty Travel% | 56.42 | 60.81 | 60.08 | 57.59 |
| Average of Unloader Idle Time % | 43.41 | 46.70 | 46.75 | 43.26 |
| Average of Hauler Idle Time % | 24.34 | 56.60 | 55.95 | 26.64 |

Comparing the makespan for each door number scenario, the same general behavior can be observed, H1 and H4 are the best performers and have similar results for all responses. Unloader empty travel %'s for H1 and H4 is slightly higher compared to H2 and H3, although the difference is no more than 1%. This could be due to the fact that in the simulation H1 and H4 have unloaders select the closest flow racks to the unloading door, meaning their "scatter" through the dock floor is less compared to H2 and H3 where unloaders move further away from the door when they choose the flow rack closer to the loads storage aisle. Once the trailer is finished the idle unloader goes to the idle location besides the flow rack of the last drop until called for the next trailer. It is possible that this "scatter" factor has an effect on empty travel if the next trailer assigned to the idle unloader is further away in more occasions in H1 and H4 than H2 and H3. As expected given the model reformulation of Section 5.2, there is a direct correlation between the first and fifth response variables (makespan and average haulers'idle time, respectively).

*Table 19. Response Variable Global Summary*

| Response | H1 | H2 | H3 | H4 |
|---|---|---|---|---|
| Average of Makespan (hrs) | **2.86** | 3.56 | 3.54 | 2.88 |
| Average of Unloader Empty Travel % | 51.12 | 50.53 | 50.54 | 50.90 |
| Average of Hauler Empty Travel% | 53.84 | 58.21 | 57.20 | 54.74 |
| Average of Unloader Idle Time % | 36.45 | 37.29 | 37.41 | 36.13 |
| Average of Hauler Idle Time % | 20.55 | 42.20 | 40.60 | 22.55 |

From Table 19 it can be observed that the Hauler Empty Travel % is lower for H1; this occurs because unloaders choose the closest flow rack to the loads door. Therefore, flow racks closer to the unloading door are highly repeated. Since the haulers select their loads based on SPT from their location, when they drop a load in an aisle close or in front of the unloaders "repeated flow rack" they always have pallets to pick, reducing vertical travel to adjacent flow racks. H2 has the highest hauler empty travel % because this rule makes the hauler move towards the unloaders if they are idle, increasing empty travel. Similar to Unloader Empty Travel %, there is not much of a difference in Unloader Idle Time %'s between heuristics, but the lowest values are obtained from H1 and H4. This slight difference is mainly due to the waiting times for the assignment of the following trailers. As for our final response, Hauler Idle Time %'s the lowest value is obtained in H1, followed by H4, significantly higher than H2 and H3 which show nearly double the amount of hauler idle time. This means that haulers wait longer for assignments in idle locations in H2 and H3. This seems to have a direct effect on makespan since the highest values of Hauler Idle Time % correspond to H2 and H3 which have the largest makespan values as well.

### 9.4 Conclusion Based on Experimental Results

Summarizing the results for this Chapter, it was determined that for our main response variable, *Makespan*, there was no significant difference between *Heuristics 1 and 4*, which compared to *Heuristics 2 and 3*, both presented better results. Reviewing these heuristic rules:

- Heuristic 1 (H1) – Unloader chooses flow rack closest to the unloading door on the way to the storage aisle
- Heuristic 4 (H4) – Unloader chooses flow rack closest to the unloading door with lowest utilization on the way to the storage aisle

The difference between both heuristics is that H1 will always select the closest available flow rack to the unloading door and H4 selects the emptiest flow rack closest to the door. In terms of *Makespan*, *Heuristic 1* would be the best choice in practice because it is trivial to implement as a rule-of-thumb without requiring any technology. For the remaining response variables *Heuristic 1* clearly favors *Hauler Empty Travel %* with the lowest value of 53.84% and *Hauler Idle Time %* with the lowest value of 20.55%. The result for the *Unloader Empty Travel %* is only 0.58% away from the best performing heuristic in this response variable, *Heuristic 2*; and the result for *Average Unloader Idle %* is only 0.44% away from the best performing heuristic in this response variable, *Heuristic 4*. In conclusion, *Heuristic 1* is the best-found heuristic as it balances the haulers' idle and total vertical travel time. Table 20 summarizes the results of this Chapter.

*Table 20. Response Variable Comparisons: Means vs Heuristics*

| Heuristic | Average of Makespan (hrs) | Average of Unloader Empty % | Average of Hauler Empty % | Average of Unloader Idle % | Average of Hauler Idle % |
|---|---|---|---|---|---|
| 1 | 2.86 | 51.12 | 53.84 | 36.45 | 20.55 |
| 2 | 3.56 | 50.53 | 58.21 | 37.29 | 42.20 |
| 3 | 3.54 | 50.54 | 57.20 | 37.41 | 40.60 |
| 4 | 2.88 | 50.90 | 54.74 | 36.13 | 22.55 |

Unfortunately, it was not possible for us to obtain the optimal solution for any of the instances or at least a tight lower bound. Hence, it is not possible for us to establish with any certainty how good H1. On the other hand, H1 was designed considering the inherent characteristics of the problem and it consistently outperformed the other heuristics. Furthermore, this is the first research to explore the RLP and it establishes an easy to code rule-of-thumb that can be used as a comparison basis for future research.

## 10. Conclusions & Future Work

This thesis presents new managerial insights on operational-level decisions related to the receiving logistics problem (RLP) in DCs and XDs with staging areas. The RLP simultaneously considers: *i*) unloaders' assignment and scheduling, *ii*) loads-to-flow rack assignment, and *iii*) haulers' assignment and scheduling. The objective of the problem is to minimize the makespan (*i.e.*, the time until all trailers are unloaded and all pallets are stored). A linear mixed integer linear formulation of the RLP is presented and the problem is shown to be NP-hard. An interesting transformation of the objective function is presented from the perspective of the haulers and it is concluded that minimizing the time to unload and store all pallets is equal to minimizing the haulers' idle time and vertical travel distance. Based on this insight, four heuristic-rule methods were proposed and evaluated with a full factorial design of experiment with different factor conditions such as *receiving door numbers*, the *number of trailers*, the *pallet storage assignment distribution per trailer*, and *trailer to door assignments*. The results of the simulation show that the preferred heuristic was H1 in which the unloaders simply select the flow rack closest to the inbound door. In terms of makespan, H1 outperforms H2 by 10.89%, H3 by 10.59%, and H4 by 0.30%. This heuristic not only minimized total makespan, but also minimized hauler empty travel %. H1 Unloader Empty Travel % was $4^{th}$ in ranking compared to other heuristics, but only 0.58% away from the best result in this response variable H2. By examining the percent of empty travel for unloaders and haulers under rule H1 it was observed that for these experiments the haulers were the bottleneck. Hence, a second set of experiments were designed where the distances between dock doors and flow racks were tripled. It was found that H1 was also the best-performing rule under these circumstances. Interestingly, a simple rule-of-thumb that does not require any technology (such as H1) outperforms more complex rules that require information not necessarily readily available to unloaders.

From a management point of view, having an efficient (good quality and fast) rule to solve the RLP, which is at the operational level, enables us to address tactical-level problems such as the trailer-to-door assignments and unloading sequences that would minimize total makespan given a set of trailers, their contents, and available resources. Notice that a simple meta-heuristic can be used to search the solutions' neighborhood space, and given a candidate solution one can evaluate its fitness using the methods defined in this thesis. Furthermore, the simulation could be easily adapted to a specific layout by modifying parameter data. Also, more factor levels could be implemented to establish the number of unloaders and haulers that would improve performance. The Kinect framework section provides real-time receiving dock monitoring system, and a methodology to gather operational data on real-time that could be used to feed the simulation such as pallet trailer unloading times (considering time gaps between trailer entry and exits).

The concept of implementing fixed staging areas or mechanized flow racks, also provides the framework for the implementation of automated material handling systems, such as automated guided vehicles (AGVs), on the receiving dock due to the fact that pallets will now become available on fixed locations easily available for pickup, instead of scattered through the receiving floor.

Another elaboration of this work would be to develop a mathematical model that incorporates operational costs. This new model could also incorporate the costs to operate different material handling equipment (e.g., Bartolomei-Suarez and Egbelu, 2000) and evaluate the results of varying the number of available resources. With this new response variable managerial decisions could be made to establish a balance between costs and makespan.

Future implementation of this work could be elaborated in a similar manner to consider all DC operations that follow hauling, these would be pallet put-away and replenishment, order filling, and shipping. Although, the complexity would definitely surpass this work, but the capacity to simulate an entire DC would provide great insights to a wide-range of operation-level decisions.

## 11. References

[1] Bartholdi, J.J. and Gue, K.R. (2000) “Reducing labor costs in an LTL crossdocking terminal” *Operations Research* 48(6), 823–832.

[2] Bartolomei-Suarez. S.M. and Egbelu, P.J. (2000) “Quadratic assignment problem QAP with adaptable material handling devices” International Journal of Production Research 38(4), 855-873.

[3] Blais, F. (2004) "Review of 20 Years of Range Sensor Development" *Journal of Electronic Imaging*, 231-240.

[4] Bozer, Y.A. and Carlo, H.J. (2008) “Optimizing Inbound and Outbound Door Assignments in Less-than-truckload Crossdocks” *IIE Transactions*, 40(11), 1007-1018.

[5] Buijs, P. and Vis, I.F.A. (2014) “Comparing Industry and Academic Perspectives on Cross-docking Operations” in *Progress in Material Handling Research: Volume XIII*, *MHI*, Charlotte, North Carolina.

[6] Buijs, P., Vis, I.F.A., and Carlo, H.J. (2014) “Synchronization in Cross-docking Networks: A Research Classification and Framework” *European Journal of Operational Research*, 239(3), 593–608.

[7] Carlo, H.J. and Aponte-Rodríguez, E.J. (2012) “A Decision Support Tool for Yard Logistics in Pharmaceutical Industry” *OR Insight*, 26(1), 32–46.

[8] Carlo, H.J. and Bozer, Y.A. (2011) “Analysis of Optimum Shape and Door Assignment Problems in Rectangular Unit-Load Crossdocks” *International Journal of Logistics: Research and Applications*, 14(3): 149–163.

[9] Carlo, H.J., Martínez, Y., and Pomales-García, C. (2014) “Real-Time Dock Door Monitoring System Using a Kinect Sensor” *in Progress in Material Handling Research: Volume XIII*, MHI, Charlotte, North Carolina.

[10] Comaniciu, D., Ramesh, V., and Meer, P. (2003) "Kernel-Based Object Tracking" *IEEE Transactions on Pattern Analysis and Machine Intelligence* (IEEE) 25(6), 564-577.

[11] Dutta, T. (2012) "Evaluation of the Kinect sensor for 3-D kinematic measurement in the workplace" *Applied Ergonomics 43*, 645-649.

[12] Faion, F., Friedberger, S., Zea, A., and Hanebck, U.D. (2012) "Intelligent Sensor-Scheduling for Multi-Kinect-Tracking" *International Conference on Intelligent Robots and Systems.* Vilamoura: IEEE, 3993-3999.

[13] Gu, J., Goetschalckx, M., and McGinnis, L. (2006) “Research on Warehouse Operation: A Comprehensive Review” *European Journal of Operational Research*, 177, 1-21.

[14] Gue, K.R. (1999) “The effects of trailer scheduling on the layout of freight terminals,” *Transportation Science*, 33(4), 419–428.

[15] Hillier, F.S. and Lieberman, G.J. (2001) *Introduction to Operations Research*, 7th Edition, McGraw-Hill, Boston.

[16] Khoshelham, K. (2010) "Accuracy Analysis of Kinect Depth Data" *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*: 133-138.

[17] Landt, J., (2005) "The history of RFID." *IEEE Potentials* 24(4), 8-11.

[18] Lenstra, J. K., Rinnooy Kan, A. H. G., and Brucker, P. (1977) “Complexity of machine scheduling problems” *Annals of Discrete Mathematics* 1, 343–3.

[19] Li, B.Y.L., Mian, A.S., Liu, W., and Krishna, A. (2013) "Using Kinect for Face Recognition Under Varying Poses, Expressions, Illumination and Disguise" *Applications of Computer Vision.* Tampa: IEEE, 186-192.

[20] Li, X., Chen, I.Y-H., Thomas, S., and MacDonald, B.A. (2012) "Using Kinect for monitoring warehouse order picking operations" *Australasian Conference on Robotics and Automation.*

[21] Livingston, M.A., Sebastian, J., Ai, Z., and Decker, J.W. (2012) "Performance Measurements for the Microsoft Kinect Skeleton" *Virtual Reality Short Papers and Posters (VRW).* IEEE, 119-120.

[22] Macknojia, R., Chávez-Aragón, A., Payeur, P., and Laganiére, R. (2012) "Experimental Characterization of Two Generations of Kinect's Depth Sensors" *Robotic and Sensors Environments.* Ottawa: IEEE, 150-155.

[23] Maimone, A. and Fuchs, H. (2012) "Reducing interference between multiple structured light depth sensors using motion" *Virtual Reality Short Papers and Posters.* Costa Mesa: IEEE, 51-54.

[24] Martin, C.C., Burkert, D.C., Choi, K.R., Wieczorek, N.B., McGregor, P.M., Herrmann, R.A., and Beling, P.A. (2012) "A Real-time Ergonomic Monitoring System using the Microsoft Kinect" *Systems and Information Egineering Design Symposium.* IEEE. 50-55.

[25] Martínez, Y. and Carlo, H.J. (2016) "Operational-Level Optimization of Inbound Intralogistics*" in Progress in Material Handling Research: Volume XIV*, MHI, Charlotte, North Carolina.

[26] Nakamura, Takayuki (2011) "Real-time 3D Object Tracking Using Kinect Sensor." *Robotics and Biomimetics (ROBIO).* IEEE, 784-788.

[27] Shakeri, M., Low, M.Y.H., Turner, S.J., and Lee, E.W. (2012) "A Robust Two-phase Heuristic Algorithm for the Truck Scheduling Problem in a Resource-constrained Crossdock," *Computers & Operatons Research*, 39(11), 2564-2577.

[28] Smisek, J., Jancosek, M., and Pajdla, T. (2011) "3D with Kinect." *Computer Vision Workshops.* Barcelona: IEEE, 1154-1160.

[29] Sullivan, Michael S., and Jamshed H. Dubash (2004) "RFID SYSTEM PERFORMANCE MONITORING" US Patent 7,161,489 B2.

[30] Thompkins, J., White, J., Bozer, Y., and Tanchoco, J. (2010) "*Facilities Planning* 4$^{th}$ Edition" pp. 204-208.

[31] Tsui, L.Y. and Chang, C.H. (1990) "A microcomputer based decision support tool for assigning dock doors in freight yards" Computers and Industrial Engineering 19 (1–4), 309–312.

[32] Tsui, L.Y. and Chang, C.H. (1992) "An optimal solution to a dock door assignment problem" *Computers and Industrial Engineering*, 23 (1–4), 283–286.

[33] Utama, M.R.U., Widyawan, W., Putra, G.D., and Santosa, P.I. (2012) "Indoor Human Tracking Application Using Multiple Depth-Cameras" *Advanced Computer Science and Information Systems (ICACSIS).* IEEE, 307-312.

[34] Vijayanagar, K.R., Loghman, M., and Kim, J. (2012) "Refinement of Depth Maps Generated By Low-Cost Depth Senors" *SoC Design Conference (ISOCC).* IEEE, 355-358.

[35] Vis, I.F.A. and Roodbergen, K.J. (2008) "Positioning of goods in a cross-docking environment" *Computers & Industrial Engineering*, 54(3), 677–689.

[36] Weng, S-K, Kuo, C-M., and Tu, S-K. (2006) "Video object tracking using adaptive Kalman filter." *Journal of Visual Communication and Image Representation* 17(6), 1190-1208.

[37] Yu, W. and Egbelu, P.J. (2008) "Scheduling of Inbound and Outbound Trucks in Cross Docking Systems with Temporary Storage," *European Journal of Operational Research*, 184, 377-396.