**國立臺灣科技大學**



**工業管理系**

**碩士學位論文**

**學號：M10701819**

**以貨就人揀貨系統於不同單品數量與訂單指派下之產量分析**

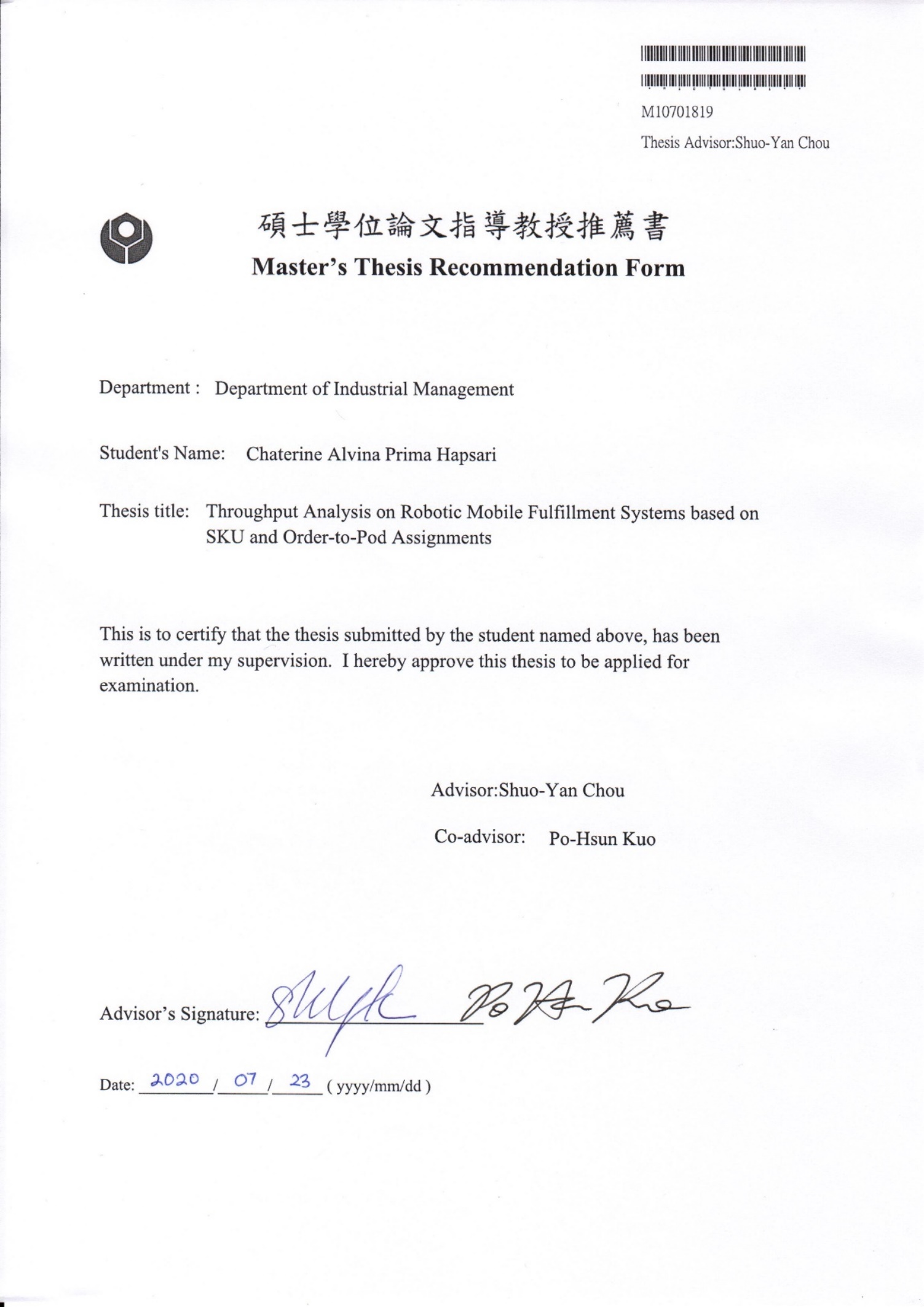
**Throughput Analysis on Robotic Mobile Fulfillment Systems based on SKU and Order-to-Pod Assignments**

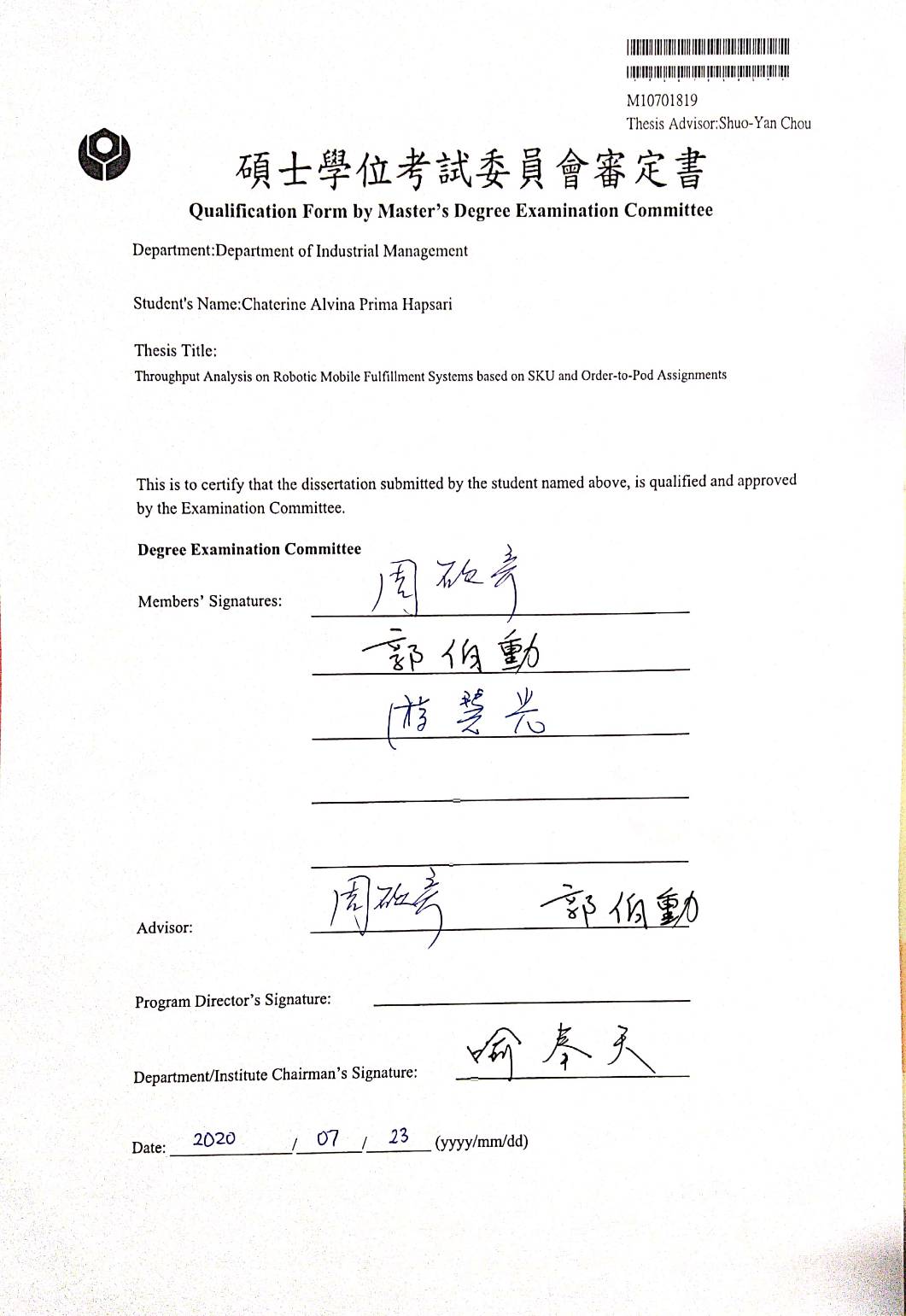
**研 究 生： Chaterine Alvina Prima Hapsari**

**指導教授： 周碩彥 博士**

**郭伯勳 博士**

中華民國一零九年七月





# Abstract

The Robotic Mobile Fulfillment System, a well-known automated parts-to-picker system, has changed the way e-commerce companies, especially Amazon, handle their order fulfillment. Through this approach, mobile robots play the role to pick the movable racks, so-called pods, to the workstation and return it to the storage area. The system performance based on the throughput rate in this system is the indicator of how well the system fulfills the order. Various decision rules influence the throughput rate in the system. Nevertheless, the order-to-pod assignment becomes the critical assignment as it is the first assignment that directly influenced by the external input, the customer order. Furthermore, customer order data also needed to be evaluated to let the system has a better response on order fulfillment.

This research thus focuses on the order-to-pod assignment under consideration of customer order trends to improve the throughput rate. The proposed decision rule for order-to-pod assignment is the likelihood rate. This value is carried out by each SKU in the pod as the decision to select the best pod by predicting which pod tends to carry frequent SKUs to be ordered. The analysis is conducted to compare the baseline scenario with random decision rule and improved scenario with the likelihood rate decision rule for order-to-pod assignment in terms of the throughput rate. Moreover, the simulation with different number of SKUs in a pod is also conducted to understand whether it will influence the throughput rate or not.

The findings of this research are improved scenario for order-to-pod assignment boost 18% of the throughput rate. Moreover, allocating 10 SKUs in a pod produces the highest throughput rate compared to other numbers of SKUs. The total delivered pods also reduce when allocating a higher number of SKUs in a pod.

**Keywords**: Robotic Mobile Fulfillment System, Simulation, Order to Pod Assignment, Order Batching, Agent-Based Modeling

# Acknowledgement

First of all, my highest praise goes to Jesus Christ.

I would like to express my sincerest gratitude to my advisor, Prof. Shuo-Yan Chou, who has supported and guided me throughout my research and thesis. His ideas, kindness, and advice that have encouraged and motivated me to enhance my work further and achieve a great outcome. I would also like to acknowledge Prof. Po-Hsun Kuo as my co-advisor and Prof. Tiffany Hui Kuang Yu as my thesis defense committee for their encouragement, insightful comments, evaluation, and suggestions for my research.

I also would like to give my appreciation to all my labmates in the Center of IoT Innovation - Information Technology Application and Integration (CITI-ITAI) laboratory, especially Kiva teammates; Agnes, Prana, Tina, and Edwin for their friendliness and support during my study in NTUST. Another big gratitude and hug go to my best friends; Eugene, Zula, Lynda, Rina, Ar Ruum, Fitri, and Monica, who give tons of love, support, patience, and irreplaceable memories on this roller coaster journey in Taiwan.

Furthermore, I express my gratitude to my mother and little brother for providing me with unfailing support and continuous encouragement throughout my years of study. This accomplishment would not have been possible without them.

Last but not least, I would like to express special gratitude to Drestanta Nindya, for his continuous encouragement, support, love, patience, advice during my study in Taiwan.

Chaterine Alvina Prima Hapsari (凱薩琳)

Taipei, July 2020

# Table of Contents

[Abstract iii](#_Toc518975633)

[Acknowledgement iv](#_Toc518975634)

[Table of Contents v](#_Toc518975635)

[List of Figures vii](#_Toc518975636)

[List of Tables viii](#_Toc518975637)

[Chapter 1 Introduction 1](#_Toc518975638)

[1.1 Background and Motivation 1](#_Toc518975639)

[1.2 Problem Formulation](#_Toc518975640) 3

[1.3 Research Objective 3](#_Toc518975641)

[1.4 Scope and Limitation 3](#_Toc518975642)

[1.5 Organization of the Thesis 4](#_Toc518975643)

[Chapter 2 Literature Review 5](#_Toc518975644)

[2.1 Modeling Approach 5](#_Toc518975645)

[2.1.1 Simulation 5](#_Toc518975653)

[2.1.2 Queuing Network 6](#_Toc518975654)

[2.1.3 Deterministic Optimization 7](#_Toc518975655)

[2.2 Warehouse Design 8](#_Toc518975646)

[2.3 Throughput 9](#_Toc518975647)

[2.4 Research Gap 10](#_Toc518975647)

[Chapter 3 Modeling 14](#_Toc518975649)

[3.1 System Model 14](#_Toc518975650)

[3.1.1 Flow Process of RMFS 14](#_Toc518975653)

[3.1.2 System Architecture 17](#_Toc518975654)

[3.1.3 Warehouse Layout 18](#_Toc518975655)

[3.2 System Configuration 21](#_Toc518975658)

[3.3 Simulation Configuration 22](#_Toc518975658)

[3.2.1 Simulation based on Decision Rules for Order-to-Pod Assignment 22](#_Toc518975651)

[3.2.2 Simulation based on Number of SKUs in a Pod 25](#_Toc518975652)

[Chapter 4 Result and Discussion 26](#_Toc518975663)

[4.1 Result of Simulation based on Decision Rules for Order-to-Pod Assignment 26](#_Toc518975665)

[4.1.1 Baseline Scenario 26](#_Toc518975666)

[4.1.2 Improved Scenario 27](#_Toc518975667)

[4.1.3 Comparison between Baseline Scenario and Improved Scenario 28](#_Toc518975666)

[4.2 Result of Simulation based on Number of SKUs in a Pod 29](#_Toc518975665)

[4.2.1 Throughput Result for Different Number of SKUs in a Pod 29](#_Toc518975666)

[4.2.2 Comparison between Different Number of SKUs in a Pod 29](#_Toc518975667)

[Chapter 5 Conclusion and Future Research 32](#_Toc518975673)

[5.1 Conclusion 32](#_Toc518975674)

[5.2 Future Research 33](#_Toc518975675)

[References 34](#_Toc518975677)

# List of Figures

Figure 3.1 Sequence Diagram of RMFS 15

Figure 3.2 System Architecture of RMFS 17

Figure 3.3 RMFS User Interface in NetLogo 19

Figure 3.4 Warehouse Layout of RMFS 20

Figure 4.1 Behavior on Random Decision Rule 27

Figure 4.2 Behavior on Likelihood Rate Decision Rule 28

Figure 4.3 T-test Result of Baseline and Improved Scenario 29

Figure 4.4 ANOVA Test Result of Different Number of SKUs in a Pod 30

Figure 4.5 Grouping Information Using Tukey Test 30

Figure 4.6 Tukey Simultaneous Test 31

Figure 4.7 Total Delivered Pods of Different Number of SKUs in a Pod 31

# List of Tables

[Table 2.1 Research Gap 11](#_Toc47647071)

[Table 3.1 Overview of Decision Rules in The Research 23](#_Toc47647072)

[Table 3.2 Pseudocode of Likelihood Rate Rule on Order-to-Pod Assignment 23](#_Toc47647072)

[Table 4.1 Average Throughput Rate for Different Number of SKUs in a Pod 29](#_Toc47647073)

# Chapter 1 INTRODUCTION

## Background and Motivation

In this era with high internet usage, the e-commerce industry has gained its benefit as it is shown on its annual growth rate and projected to grow 8.1% from 2020 to 2024 [1]. Its growth has shaped the industry to offer shorter and shorter shipping lead-time from 2-days shipping, next day shipping, or even same-day shipping as their form of competition to carrier company in order to handle the delivery as their in-house operation [2]. Handling with tight delivery schedules is not the only challenge faced by the e-commerce industry, specifically Business to Customer (B2C) segment. They also have to cope with high volatile demand, which its demand usually in a small quantity but offers in large assortment [3]. It is shown that the operation in the e-commerce industry reflects on how customer demand is.

In order for the e-commerce industry to fulfill the customer order, the whole operation inside the company should be well managed, especially the operations within their warehouse, where the products are received, stored, sorted, picked, packed, and shipped. Order picking is known as the extensive labor operation in the warehouse and high-cost operation, which spend approximately 55% of the total operating expense [4]. Thus, order picking becomes the critical operation for improvement, as the company has to perform efficiently and effectively on fulfilling customer demand.

The implementation of automation in the order picking process is one method to boost warehouse performance. Among various types of automated picking systems, The Robotic Mobile Fulfillment System (RMFS) innovated by Kiva System, and then acquired by Amazon Robotics, is known as a suitable solution to handle the e-commerce challenges by its flexibility and scalability [5]. Robotic Mobile Fulfillment System, as an automated parts-to-picker system, not only improves the productivity rate but also reduce the pickers traveling time compared to traditional picker-to-parts system [5]. The concept of this system is assigning mobile robots to move throughout the warehouse to bring movable shelf, so-called, pod, which contains inventory back and forth between the storage area and workstation [6].

A system can be evaluated whether it is a good or bad system based on the result of the performance measures. The performance measures can be categorized in terms of time, quality, cost, and productivity, and the selection of evaluated performance measures differ in each case following their own goal [7]. This Robotic Mobile Fulfillment System also has several performance measures, but this performance measure, so-called unit throughput rate, is highly correlated with other performance measures. As the system performance is following the determined decision rules, the unit throughput rate in this system is greatly affected by the decision rules on the pick order assignment [8]. Pick order assignment (POA) is the assignment for selecting pick orders from the backlog to the picking station [9].

Although there are numerous researches mainly evaluate the system based on throughput, either in the dimension of time or productivity rate, the analysis of decision rules for pick order assignments in the part-to-picker system has not yet considered the customer order trend. Thus, this research focuses on improving the throughput rate on the Robotic Mobile Fulfillment System by considering the customer orders trend for pick order assignment. Different from previous research, the pick order assignment in this research is the assignment for selecting the best pod to carry the order to the picking station, which later is described as an order-to-pod assignment. This difference is based on the consideration that the first destination of the robot is the pod, while the picking station is the second destination. Thus, the pod selection becomes the main focus of this research.

Meanwhile, the customer orders should be transformed into a variable as the decision to select the pod. That variable is named as the likelihood rate, which is the indicator of how likely each SKU (Stock Keeping Units) is ordered. This value is carried out by each SKU in the pod as the decision to select the best pod by predicting which pod tends to carry more orders to the picking station. This method is following the concept of class-based storage by measuring the class based on the demand frequency of the product for storage assignment policy [4]. Thus, the proposed decision rule for order-to-pod assignment in this research is based on the likelihood rate. For comparison, the baseline scenario is generated by implementing a random decision rule for the order-to-pod assignment. This research will compare the proposed decision rule with the baseline scenario in terms of the throughput rate.

Furthermore, previous research has mentioned that the number of pod visiting the picking station decreases as the number of products in the pod increases [10]. It is expected that the increasing number of products in the pod can increase the throughput rate. Thus, this research also evaluates whether the number of SKUs allocated in a pod will affect the throughput rate.

## Problem Formulation

According to the research background, the problems formulated in this research are as follows:

1. How is the SKU likelihood rate as the decision rule for an order-to-pod assignment can improve the throughput rate?
2. How is the number of SKUs allocated in a pod influence the throughput rate?

## Research Objective

According to the research background and the problem formulation, the objectives of this research are as follows:

1. Find better decision rules for order-to-pod assignment in order to improve the throughput rate
2. Find the best configuration for the number of SKUs allocated in a pod

## Scope and Limitation

The scope and limitation of this research are addressed as follows:

1. The system performance in this research is only evaluated based on the throughput rate
2. The Robotic Mobile Fulfillment System is modeled as an agent-based simulation using NetLogo as the agent-based simulation platform integrated with Python as the tool for computational
3. The order picking process in this system is covering the picking process and replenishment process
4. The replenishment process in this system is only implemented virtually, without having a realistic simulation to the replenishment station
5. The mobile robot does not consider motion properties, such as acceleration and deceleration. It only has a constant speed.

## Organization of the Thesis

The organization of this study is divided into five major sections. In the first section, Chapter 1 presents the background and motivation of the study, problem formulation, research objectives, research scope, and limitation. The literature review and modeling are explained in Chapter 2 and Chapter 3, respectively. Chapter 4 describes the results and discussion. In the last section, Chapter 5 presents the conclusion and future research opportunities.

# Chapter 2 LITERATURE REVIEW

## Modeling Approach

The selection of the modeling approach should be determined first in order to know how to solve the problem. This subsection will explain several modeling approaches and its case related to the topic of the research.

### Simulation

Reflecting on the innovation of the Kiva system, the development of a simulation environment focusing on a massively multi-vehicle system, so-called, Alphabet Soup has been introduced. This simulation assigns the robots to carry buckets containing letters from the letter receiving stations to the word-assembly stations. This simulation can be an overview of how is the work on multi-agent systems, resource allocation, vehicle coordination, and operation research [11].

Managing a new parts-to-picker system, Robotic Mobile Fulfillment System, a simulation framework called “RAWSim-O” (Robotic Automatic Warehouse Simulation (for) Optimization) is introduced. This simulation is the extensive work of Alphabet Soup but implemented in a different system. This simulation can be a great tool to analyze the effect of multiple rules applied in each decision problem on the Robotic Mobile Fulfillment System. Moreover, this simulation has been tested on the integrated, simple robot prototypes based on vacuum cleaning robots [9].

Another approach to evaluate the design and performance of the system is called Agent-Based Modelling and Simulation (ABMS). As there are numerous Agent-Based Modelling and Simulation (ABMS) software tools, a comparison structure of each tool is presented with a description as follows: software tool name, license/availability, source code, type of agent-based on its interaction behavior, coding language, compiler, model development effort, modeling strength, and scalability level, and ABMS scope or application domain. These are several examples of ABMS software tools reviewed in this research; FlexSim, MASON, Mathmatica\* (Wolfram), Mesa, NetLogo, and SimEvents (MATLAB\*). Among numerous ABMS software tools, NetLogo is known as an open-source software tool with simple/easy model development effort and medium to high scalability level using Scala code as the source code [12].

### Queuing Network

Queuing network is classified into three categories, which are open queuing network (OQN), closed queuing network (CQN), and semi-open queuing network (SOQN). Previous research focuses on modeling the Robotic Mobile Fulfillment System in the form of semi-open queuing networks with backordering by the lost-customer system. This method aims to see the relationship between the number of robots and the average waiting time for a task. The average waiting time is reduced significantly, with only adding one more robot, while it becomes stagnant by adding more robots in the system [13].

Previous research builds semi-open queuing networks (SOQN) to solve the problem. It uses a two-phase approximate approach to compare the decision rules on assignment, i.e., random, handling-speed-based, near-optimal, and optimal assignment rules [14]. Similar to previous research, this research introduces a new type of semi-open queuing networks (SOQN): cross-class matching multi-class SOQN. The research finds the optimal solution of three decision variables, i.e., the number of pods per product, the ratio number of pick stations to replenishment stations, and the replenishment level per pod [15].

The combination of single class semi-open queuing network (SOQN) and closed queuing network (CQN) also can be implemented to analyze the system performances, such as maximum order throughput, average order cycle time, and robot utilization. This model is used by considering the type of orders, i.e., single-line order and multi-line orders, and either using storage zones or not. The result of this method can estimate accurately on the given system performances [16].

The implementation of the queueing network, specifically closed queuing network (CQN) model, also can be used for robot assignment strategy by considering a different type of storage zones and characteristics of the robot. The research result shows that the implementation of a single storage zone reduces the throughput time up to one-third of the initial value when using pooled robots. However, the expected replenishment time increases up to three times. Whereas, implementing multiple zones for the activity of the robot to the storage zone with dedicated and shortest queue produces more significant throughput compared to random strategy [17]. Meanwhile, other research also implements a closed queuing network (CQN) model for class-based storage but adding pod storage policies in the model, i.e., random open location storage and closest open location storage. The research result shows that the closest open location pod storage policy increases the system throughput for all item classes but does not use the storage spaces efficiently compared to random location pod storage policy [18].

Different from other previous researches, an open queuing network (OQN) is built for understanding the RMFS using two different roles, either as dedicated robots or pooled robots. The result shows the optimal number and velocity of the robots in terms of total throughput time [19].

As the robot in RMFS is powered by battery, a study focuses on battery management issues by evaluating the strategy either by battery charging or swapping. The model is built using a semi-open queuing network (SOQN) to estimate the system performance. The result shows that battery management affects throughput significantly. It turns out that inductive charging outperforms other battery recovery policies, while battery swapping produces 4.88% throughput time lower than plug-in charging [20].

### Deterministic Optimization

Several kinds of research implement deterministic optimization to solve the RMFS problem. One of the research implements a mixed-integer programming (MIP) model to solve order processing in a picking station [21]. Another research builds a deterministic model to find the optimal place for returning pod to the storage area, so-called, Pod Repositioning Problem (PRP). The finding of this research is introducing an algorithm named Tetris algorithm, which is considered robust and flexible enough to be applied up to larger instances [22].

The implementation of deterministic optimization, specifically on new mixed-integer programming (MIP) model, can be used to analyze the system performance while applying decision on the assignment of pods to stations and orders to the station. Different from previous research that implements assignment of pods to stations and orders to station sequentially, this particular research extends the works by integrating those two assignments. Furthermore, this research allows splitting order by letting the demanded items for the same order to be delivered to the different picking station. The decisions for integrating two assignments and allowing split orders improves the performance, specifically the throughput by 46% [23].

Order batch picking optimization is known as one of the problems should be tackled for improving the operational in e-commerce warehouse, specifically on manual picking system. This particular research integrates four algorithms, i.e., location interval distance algorithm, location selection algorithm, routing algorithm, and order batching algorithm to solve this problem considering different storage scenarios. The implementation of this approach considerably has excellent performance to solve the problem with different sizes, both in solution quality and computation efficiency [24]. Another approach to solve order batching problem is by implementing association-based clustering considering the customer demand patterns. The idea of this approach is to discover the association between orders to set in the same batch and combined it with 0-1 integer programming to maximize the association between orders within each batch. This proposed approach outperforms other methods, i.e., FCFS and other existing heuristics approach [25].

## Warehouse Design

One of the criteria to have an excellent performance in the RMFS is the warehouse design. Thus, there are several kinds of research conducted related to warehouse design. RMFS is categorized as an automated warehouse system with robot pickers. Its objectives are affected by several decision variables; one of them is physical layout configurations. It is reemphasizing that the number of aisles, depth of aisles, number of cross aisles, and number of tiers affect the picking decisions and the workload of the pickers. It is also mentioned that automation helps to provide saving in space and labor [26].

Following the storage assignment problem, there are several decisions needed based on the elements in the warehouse. The pods on storage areas in RMFS are organized into many pod batch with a specific size. The size of the pod batch will influence the number of cross aisles. The result shows that the optimal width of the shelf block decreases with the width-to-length ratio of the system [14]. Considering the width-to-length ratio of the storage area itself, the result of maximum order throughput is not sensitive to different ratios. On the contrary, the maximum order throughput is affected by the location of the workstation. Moreover, implementing the storage zone or not also influences the result of order throughput if the location of the workstation is changed [16].

As there are a large number of products handled by e-commerce, the inventory allocation in a rack is one of the vital issues that need to be handled. Allocating diverse SKUs stored in each rack will profoundly impact the picking performance and number of robots. Moreover, the implementation of shared storage policy by scattering an SKU not in a single rack but multiple racks will reduce the rack movement which results in the lower amount of the robot and gain a higher chance of having demanded item closer to picking station [21].

Similar to previous research, another research mentions that spreading the inventory to multiple pods will massively improve the throughput performance. Moreover, this research also evaluates the optimum ratio between the number of picking stations and replenishment station, which is 4 to 2. Considering the replenishment process, the optimal level of inventory in a pod to be replenished is before it is empty, 50% of the total inventory in a pod [15].

On a smaller scope, the inventory for each pod also needs to be organized. One previous research aims to decide which product to be stored in which pod on storage assignment problem. One pod will consist of which products are often ordered together in order to maximize product similarity. The finding in this research is as the number of products in a pod in increasing, the number of pod visits the picking station tends to be reduced [10].

## Throughput

Many performance metrics can be evaluated on Robotic Mobile Fulfillment System. Each performance measure is influenced by multiple rules applied in every decision problem. The decision problem on the robot movement plays an essential role in determining the path to move on reaching the destination. Research develops a path planning algorithm by considering kinematic constraints, i.e., acceleration, deceleration, and turning time. The research shows that implementing WHCAv\* produces the best performance metrics, i.e., throughput, path length, and search time [27].

Another research analyzes decision rules for RMFS. Those are pick order assignment, replenishment order assignment, pick pod selection, replenishment pod selection, and pod storage assignment. Seven performance measures in RMFS are as follows: unit throughput rate, pick order throughput rate, order turnover time, distance traveled per robot, order offset, the fraction of orders that are late, pile-on, and the pick station idle time. The result shows that the pick order assignment is the most affecting decision rules for the throughput rate. It improves up to twice the throughput rate [8].

Throughput performance also can be boosted by doing pod repositioning, specifically active repositioning. There are two methods to do the pod repositioning, either parallel while the system is active or when the system downtime, i.e., night period. However, applying a greedy approach to search the repositioning move produces unnecessary moves to obtain the desired inventory well-sorted value [28].

## Research Gap

This section shows the difference between this research and other researches. The research gap also shows the contribution of this research compared to the previous researches (see Table 2.1). Based on the research gap, it is shown that the previous researches for the parts-to-picker system are focusing on the storage policy, layout design, and pod and robot assignment, but do not have a comprehensive discussion for assigning an order to pod. Meanwhile, there are researches discuss order batching on manual picking systems, which are the same as an order-to-pod assignment in the parts-to-picker system. Thus, the contribution of this research is focusing on order-to-pod assignment but considering the customer demand trend.

Table 2.1 Research Gap

| **Reference** | **Order Picking System** | **Research Issue** | **Performance Measure** | **Method** |
| --- | --- | --- | --- | --- |
| [4] | Manual order picking system | Optimal layout design, storage assignment, routing, order batching and zoning of the manual order picking system | - | Literature review |
| [8] | Parts-to-picker | Decision problems and its effect on the performance of RMFS | Throughput, order due time | Simulation |
| [9] | Parts-to-picker | Simulation framework for RMFS | System’s overall efficiency | Simulation |
| [10] | Parts-to-picker | Storage assignment based on product similarity and order batching problem | Number of pod visit | Deterministic |
| [13] | Parts-to-picker | Lost customer approximation with back-ordering | Throughput and idle time | SOQN |
| [14] | Parts-to-picker | Assignment rule of workstations to robots considering handling-speed of pickers | Throughput time | SOQN |
| [15] | Parts-to-picker | The optimal number of pods per product, ratio number of pick station and replenishment station, replenishment level per pod | Order throughput time | SOQN |
| [16] | Parts-to-picker | Optimal workstation location, length-to-width of storage area, average order cycle time for multi-line orders, zoning policy | Order throughput, average order cycle time, robot utilization | SOQN |
| [17] | Parts-to-picker | Multiple or single storage zone and robot assignment (dedicated or pooled robots) | Throughput time | CQN |
| [18] | Parts-to-picker | Class-based storage and in combination with two pod storage policies within the zone | System throughput | CQN |
| [19] | Parts-to-picker | The optimal number of robots and average speed considering dedicated or pooled robots | Throughput time | OQN |
| [20] | Parts-to-picker | Battery charging and swapping strategies | Throughput time, cost | SOQN |
| [21] | Parts-to-picker | The batch of picking orders, processing sequence, and arrival sequence of racks delivered by the mobile robot | Picking performance | Heuristics |
| [22] | Parts-to-picker | The algorithm on pod repositioning problem | Cost, computational time | Heuristics |
| [23] | Parts-to-picker | Integrated pick order assignment and pick pod selection for multiple stations, and allowing split orders | Pod visits to stations | Simulation |
| [24] | Manual order picking system | Order batch picking optimization considering different storage scenarios | Total travel distance | Heuristics and metaheuristics |
| [25] | Manual order picking system | Order batching approach considering customer demand pattern | Total travel distance | Data mining, integer programming |
| [26] | Automated order picking system | System analysis, design optimization, and operations planning and control for automated and robotic handling system | - | Literature review |
| [27] | Parts-to-picker | Path planning algorithm considering kinematic constraints | Throughput, path length, search time | Simulation |
| [28] | Parts-to-picker | Active pod repositioning | Throughput | Simulation |
| This research | Parts-to-picker | Order-to-pod assignment considering customer demand trend | Throughput rate | Simulation |

# Chapter 3 MODELING

This chapter explains how the system is modeled and built-in this research. The explanation of this chapter consists of the system model and simulation framework.

## System Model

The system model built in this research follows the concept of the Robotic Mobile Fulfillment System. There are subsections to describe further the system model.

### Flow Process of RMFS

In order to have a better understanding of the system, Figure 3.1 is provided to show the flow process in the Robotic Mobile Fulfillment System. The elements in this system are pod, mobile robot, picking station, and replenishment station, while order represents the input that enters the system. Following the concept of agent-based simulation, the movable agent in this system, while the other elements are categorized as the static agent. Furthermore, the warehouse acts as the environment of this system.

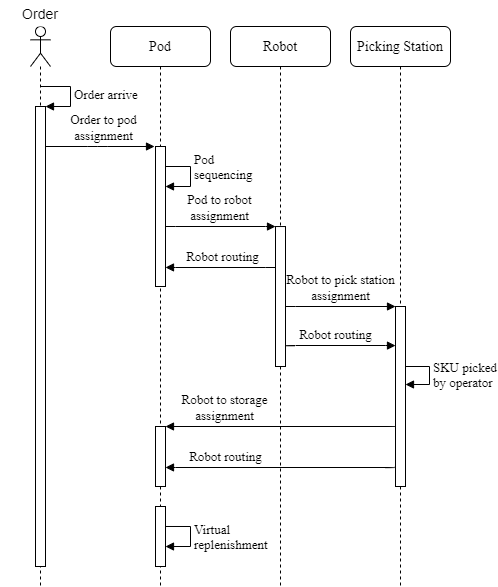


Figure 3.1 Sequence Diagram of RMFS

The full cycle of Robotic Mobile Fulfillment starts from the order arrival until the mobile robot has returned the pod to the storage area, and do virtual replenishment for inventory in a pod that reaches the threshold. The detailed explanation for each activity in this system is as follows:

* *Order arrival*

Order, as the primary input of this system, enters the system continuously following a specific distribution for its inter-arrival time.

* *Order-to-pod assignment*

The activity in this stage is the activity for assigning the order to one selected pod that carries the required SKU of the order. This assignment is triggered immediately when the orders have entered the system.

* *Pod sequencing*

Right after the pod has been chosen, a pod list is updated in sequence based on the earliest due date of the order carried by pod. This pod list indicates the pod priority to be picked first compared to the others.

* *Pod-to-robot assignment*

The activity in this stage is the assignment for finding a mobile robot to pick a pod from the storage area. This assignment is bundled with the robot routing as the input. The mobile robot candidates for this assignment are any robot that currently available or will be available from its previous job for returning pod to the storage area.

* *Robot routing*

This activity is for determining the robot routes from its initial location to the destination point. The distance calculation from this activity becomes an input for the assignment, which requires the mobile robot movement.

* *Robot-to-pick station assignment*

The activity in this stage is the assignment for selecting pick stations as the mobile robot’s destination point to deliver the pod. This assignment is bundled with the robot routing as the input.

* *Item picked by an operator*

The activity when humans as the operator do pick SKU(s) matched with the order from the pod and put it into the bin(s).

* *Robot-to-storage assignment*

The activity in this stage is the assignment for selecting an unoccupied storage location to store back the pod. This assignment is also bundled with the robot routing as the input.

* *Virtual replenishment*

The activity in this stage represents the actual replenishment process in this system. This activity is activated only when there is any inventory in the pod that reaches the threshold.

### System Architecture

The system architecture consists of three integrated platforms, i.e., NetLogo, Python, and Microsoft Excel. NetLogo is an open-source, and agent-based simulation platform created by Uri Wilensky and developed by Northwestern University [29]. It is known as a software tool that has simple and easy on model development with medium to a large scale of computational modeling strength [12]. Python is the programming language that is utilized to do the computational. The system architecture and the interaction between those three platforms can be seen in Figure 3.2.

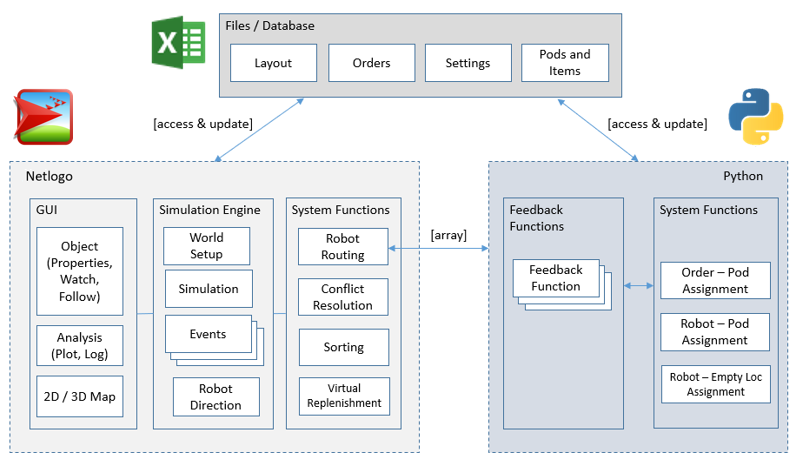


Figure 3.2 System Architecture of RMFS

The detailed explanation for this architecture is as follows:

* Microsoft Excel provides the inputs and the output to/from both NetLogo and Python in the form of CSV files, i.e., layout design, orders, settings, pod, and items.
* NetLogo acts as GUI to visualize the simulation, observe the agent’s behavior through its simulation, and show the output of the simulation. Figure 3.3 shows the user interface on NetLogo. NetLogo also plays the role of a simulation engine to set up all settings of the simulation, show the event change and robot movement during the simulation run. The system functions built-in NetLogo is the part integrated with Python to obtain the feedback on handling the robot routing and assignment.
* Python provides the feedback function based on the input received from NetLogo and returns the computational result to NetLogo, i.e., the assignment result.

### Warehouse Layout

As this research builds a simulation of the order picking process in the Robotic Mobile Fulfillment System, this subsection explains the warehouse layout configuration built-in NetLogo. Figure 3.4 shows the warehouse layout used in this research.

The warehouse design of this research is similar to the previous research but in the rotated view and smaller scale [16]. The warehouse-size built in this simulation is 35 and 45 in width and length, respectively. The picking location is located on the upper side of the warehouse, whereas the replenishment station is on the lower side. This simulation allocates five stations equally for both picking station and replenishment station. The middle area of the warehouse represents the storage area for storing the movable shelf, so-called pods. The total area provided to store the pods are 550 locations. The pods are organized in a batch with a size of 5x2. Each pod batch is positioned with the shorter dimension facing the workstation. The consideration of this configuration is to have more aisles toward/outward the workstation, which is expected provides shorter distance compared to other configurations.

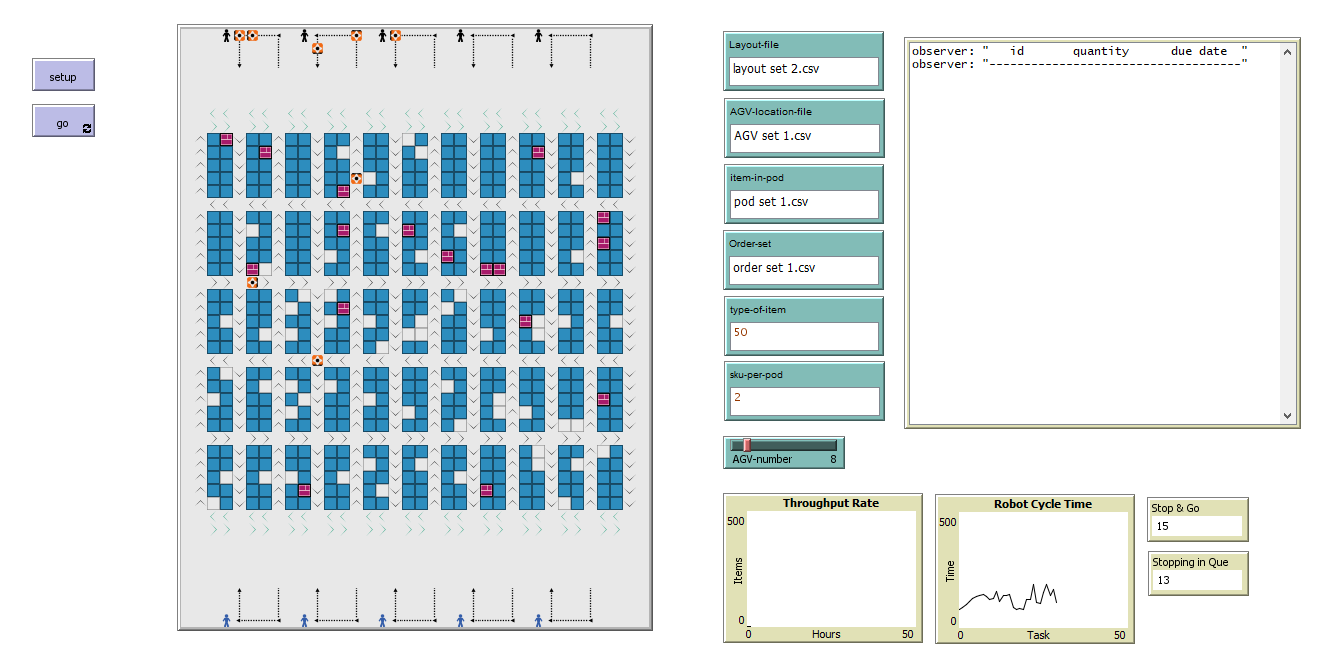


Figure 3.3 RMFS User Interface in NetLogo

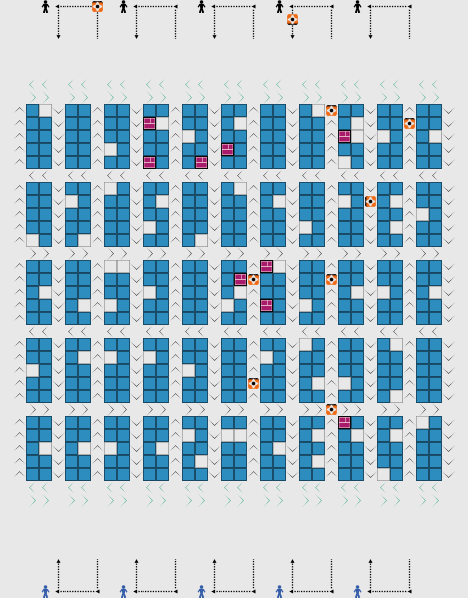


Figure 3.4 Warehouse Layout of RMFS

There are several icons illustrated in this warehouse to distinguish different elements. The light square indicates the empty location, dark square indicates the available pod, whereas the striped square illustrates the selected pod. The composition of the empty location is allocated 10% of the total storage capacity, which its location is spreading randomly throughout the warehouse. The illustration of the mobile robot is shown as the only icon spread in aisles and picking station area. The mobile robot can move in the aisle following the direction with different direction applied on each neighboring aisles. Moreover, a two-lane highway is provided near the workstation with a different direction for route option toward/outward the workstation.

## System Configuration

The system configuration explains about the initial setting of the simulation and set up the parameter in the simulation. The detailed explanation for the configuration is as follows:

1. **Simulation**

* The simulation runs continuously for 12 hours of simulation time. The simulation length represented in NetLogo is 43,200 ticks (tick is NetLogo’s unit time). The simulation length is used for observing the pattern of the simulation.
* The number of replication in this simulation is ten replications. The number of replication is used for lessening the effect of randomness.

1. **Items**

* There are 800 different types of SKUs in the system.
* Each SKU is stored randomly in the pod throughout the storage area.

1. **Order**

* The order inter-arrival time follows an exponential distribution with parameter λ = 30 seconds (ticks).
* The order enters the system in single-line order or 1 SKU per order with a single quantity.
* The order’s properties are item type, item quantity, and due date.

1. **Pod**

* Each pod can carry multiple SKUs. For the baseline scenario, one pod consists of 2 SKUs.
* The pod’s properties are pod ID, pod coordinates, item type, and item quantity.

1. **Mobile robot**

* Each robot can only carry one pod at once.
* At the initial state, the mobile robot status is idle.
* The initial robot location is randomly placed throughout the warehouse.
* The mobile robot’s properties are mobile robot ID, coordinates, assigned pod ID, and the availability status.
* The number of mobile robots is set to be thirty-five units.

1. **Picking station**

* The queue capacity in the picking station is six slots.
* The cycle time for picking items follows a Poisson distribution with parameter μ = 6 seconds (ticks) per unit SKU. Thus, the service time in a picking station is the multiplication between cycle time and the number of delivered SKUs.
* At the initial state, all the picking station is idle.

1. **Virtual replenishment**

* The checking time for virtual replenishment occurs every 100 seconds (ticks).
* The threshold for replenishment is 50% of the initial quantity stored in the pod
* The inventory is replenished as much as the inventory reaches the initial quantity stored in the pod.

## Simulation Configuration

This subsection describes how the simulation conducted in this research is. There are two simulation configuration conducted in this research, i.e., based on the decision rules and based on the number of SKUs in a pod.

### Simulation based on Decision Rules for Order-to-Pod Assignment

This subsection explains the simulation conducted based on the decision rules for the order-to-pod assignment. Moreover, the decision rules applied for other assignments remain the same while applying different decision rules for order-to-pod assignment. There are two scenarios applied in this research, i.e., baseline scenario and improved scenario. Table 3.1 provides an overview of the decision rules for each activity in the system based on the given scenarios.

Table 3.1 Overview of Decision Rules in The Research

|  |  |  |
| --- | --- | --- |
| **Activity** | **Baseline** | **Improved** |
| Order-to-pod assignment | Random | Likelihood rate |
| Pod-to-robot assignment | Nearest | Nearest |
| Robot routing | Simple rule | Simple rule |
| Robot-to-pick station assignment | Nearest and queue length | Nearest and queue length |
| Robot-to-storage assignment | Nearest | Nearest |
| Virtual replenishment | Random | Random |

The detailed explanation of decision rules for each activity in this research is as follows:

1. **Order-to-pod assignment**

* *Random*

This decision rule chooses any pod carrying the demanded SKU randomly.

* *Likelihood rate*

This decision rule considers the value of the likelihood rate carried by each SKU in the system to select the best pod. The likelihood rate indicates how likely a particular SKU to be ordered. The higher its value, the higher chance that an SKU will be ordered again. As previously mentioned that each pod consists of multiple SKUs, the value of the likelihood rate on other SKU(s) besides the required SKU becomes the criteria to select the pod. The chosen pod is the one with the highest likelihood rate of other SKU(s) in the same pod as the required SKU (see Table 3.2).

Table 3.2 Pseudocode of Likelihood Rate Rule on Order-to-Pod Assignment

|  |
| --- |
| **Pseudocode of Likelihood Rate Rule on Order-to-Pod Assignment** |
| Set j as the SKU type  For j = 0 to number of total SKUs in the system  Likelihood rate (j) = total order quantity of SKU j / total order quantity  Next SKU type  Set A = , I as the ordered SKU  If I ⸦ A  The chosen pod =  End if |

1. **Pod-to-robot assignment**

At this activity, both baseline and improved scenarios implement the nearest decision rule. This decision rule works by choosing the robot with the nearest distance from the pod’s current location.

1. **Robot routing**

The robot routing for both baseline and improved scenarios implement the simple rule. The routing is developed based on two events, i.e., when the mobile robot is carrying the pod (deliver a pod to a picking station or returning pod to storage area) and when it is not carrying the pod (about to pick the next pod). The idea of this routing is to obtain the actual distance based on its route.

1. **Robot-to-pick station assignment**

This assignment implements a decision rule based on the nearest and queue length for both baseline and improved scenario. The combination of these decision rules in this assignment is selecting the nearest distance of the picking station from the mobile robot’s current location with the shorter queue length.

1. **Robot-to-storage assignment**

The nearest decision rule in this assignment is applied to both baseline and improved scenarios. The selection of an empty location in this assignment is based on the nearest distance from the mobile robot’s location in the picking station to the possible empty location in the storage area.

1. **Virtual replenishment**

This activity is activated when the inventory reaches the threshold to be replenished. The pod with the inventory to be replenished is randomly selected.

### Simulation based on Number of SKUs in a Pod

Besides changing the different decision rules on order-to-pod assignment, this research conducts simulation based on the number of SKUs in a pod. For this type of simulation, the decision rule applied for each assignment in the system follows the decision rules on the baseline scenario. There are five different scenarios applied in this research. The baseline scenario for this simulation allocates 2 SKUs in a pod. In contrast, the other scenarios allocate 5 SKUs, 10 SKUs, 15 SKUs, and 20 SKUs in a pod, respectively.

# Chapter 4 RESULT AND DISCUSSION

This chapter explains the result of the simulation based on the decision rules for order-to-pod assignment and the simulation based on the number of SKUs in a pod. Furthermore, the comparison among different scenarios is explained for each simulation.

## Result of Simulation based on Decision Rules for Order-to-Pod Assignment

This subsection describes the simulation result of two scenarios, i.e., baseline scenario and improved scenario, and its comparison. The detailed explanation can be seen below:

### Baseline Scenario

As previously mentioned, the decision rules for order-to-pod assignment in the baseline scenario is random. The result of the baseline scenario shows that the average throughput rate is 410 units/hour. Following the behavior in the simulation, the implementation of random will make the location of selected pods (with the icon of the striped square) is scattered around the storage area, although the decision rule for robot-to-storage assignment is nearest (see Figure 4.1).

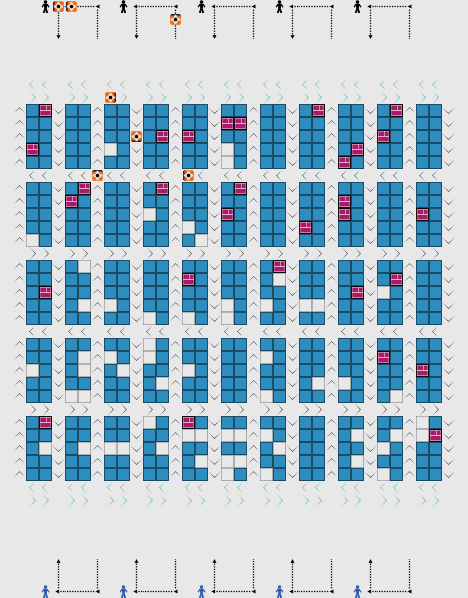


Figure 4.1 Behavior on Random Decision Rule

### Improved Scenario

The improved scenario implements the likelihood rate as the decision rule on the order-to-pod assignment. This scenario produces 485 units/hour on average for the throughput rate. The behavior of this scenario in the simulation shows that the location of the selected pods (with the icon of the striped square) are all closer to the picking station (see Figure 4.2). It happens because this scenario already considers which SKUs are frequently ordered for the pod selection, which causes a higher chance of the next selected pod is in close distance to the picking station.

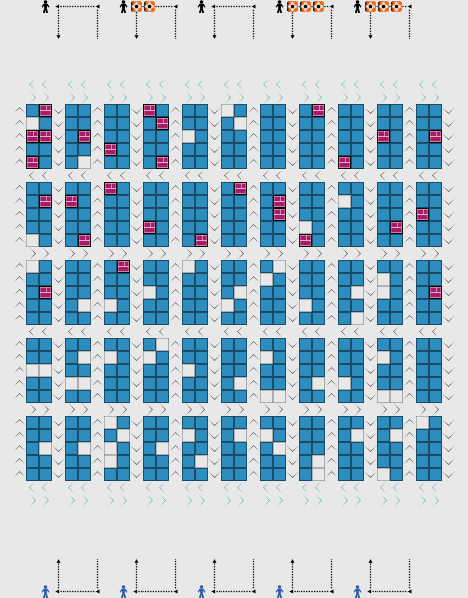


Figure 4.2 Behavior on Likelihood Rate Decision Rule

### Comparison between Baseline Scenario and Improved Scenario

The throughput rate result between the baseline scenario and the improved scenario is analyzed using a two-sample T-test. The selection of using a two-sample T-test is to check whether the mean differs significantly between two groups, i.e., baseline scenario result and improved scenario result, in which its data are statistically independent. The result of the two-sample T-test has rejected the hypothesis with the p-value less than 0.05 that the means are significantly different between the baseline scenario and the improved scenario (see Figure 4.3). Furthermore, the implementation of an improved scenario for the order-to-pod assignment can boost 18% of the throughput rate.

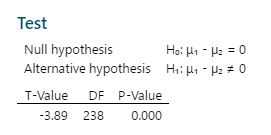


Figure 4.3 T-test Result of Baseline and Improved Scenario

## Result of Simulation based on Number of SKUs in a Pod

This subsection describes the simulation result of five scenarios and the comparison. The detailed explanation can be seen below:

### Throughput Result for Different Number of SKUs in a Pod

As previously mentioned, there are five different scenarios applied in this research. The baseline scenario for this simulation allocates 2 SKUs in a pod. In contrast, the other scenarios allocate 5 SKUs, 10 SKUs, 15 SKUs, and 20 SKUs in a pod, respectively. The average throughput rate for each scenario can be seen in Table 4.1.

Table 4.1 Average Throughput Rate for Different Number of SKUs in a Pod

|  |  |  |
| --- | --- | --- |
| **Scenario** | **SKUs** | **Throughput** |
| Baseline | 2 SKUs | 409.8 |
| Scenario #1 | 5 SKUs | 488.2 |
| Scenario #2 | 10 SKUs | 500 |
| Scenario #3 | 15 SKUs | 474.1 |
| Scenario #4 | 20 SKUs | 477.4 |

### Comparison between Different Number of SKUs in a Pod

Comparing the average throughput rate for different number of SKUs in a pod, the throughput rate is increasing from the scenario of 2 SKUs to 10 SKUs with the highest value on the scenario of 10 SKUs, but dropping down on scenario of 15 SKUs and 20 SKUs (see Table 4.1). Furthermore, the result for all scenarios is evaluated using a statistical test, i.e., one-way ANOVA and Tukey’s range test. The selection of these tests is to determine whether the means of the groups differ. The one-way ANOVA test result has rejected the null hypothesis with the p-value is less than 0.05 (see Figure 4.4) and concludes that not all of the population means are equals. Based on the grouping information using the Tukey test, it shows that 5 SKUs and 10 SKUs have a statistically higher mean than 2 SKUs (see Figure 4.5).

Moreover, Figure 4.6 shows the confidence interval for the difference between the means of all scenarios. Both results between 2 SKUs and either 5 SKUs or 10 SKUs are in the range without including zero, which indicates that the difference is statistically significant. Though both 5 SKUs and 10 SKUs are statistically significant compared to 2 SKUs (baseline scenario), the highest average throughput rate is achieved by implementing 10 SKUs in a pod. Besides, the total delivered pods are also evaluated to know how its results in different scenarios. Figure 4.7 indicates that an increasing number of SKUs in a pod requires a smaller number of pods delivered to the picking station. This finding proves what has been stated by the previous research regarding the number of pod visits [10].

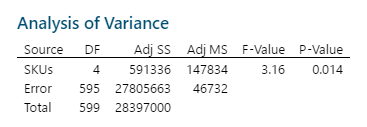


Figure 4.4 ANOVA Test Result of Different Number of SKUs in a Pod

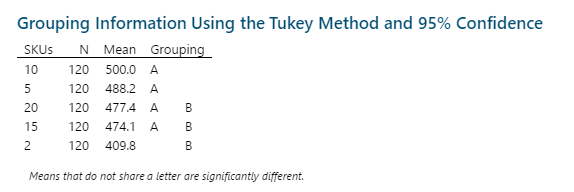


Figure 4.5 Grouping Information Using Tukey Test

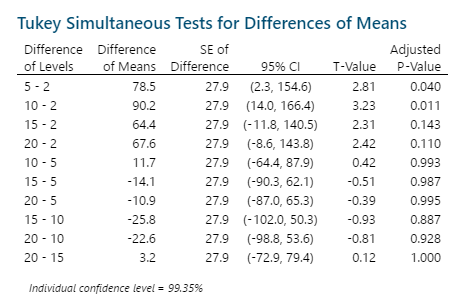


Figure 4.6 Tukey Simultaneous Test

Figure 4.7 Total Delivered Pods of Different Number of SKUs in a Pod

# Chapter 5 CONCLUSION AND FUTURE RESEARCH

## Conclusion

This research is focusing on the order-to-pod assignment with the consideration of the customer demand trend in the Robotic Mobile Fulfillment System. The selection of order-to-pod assignment is based on the previous research mentioned that the pick order assignment is the assignment with the highest impact on the throughput rate [8]. The previous research is choosing a pick order for the picking station. In contrast, this research is assigning an order to the pod. Furthermore, the order fulfillment system in an e-commerce warehouse reflects how customer demand is. Thus, understanding the customer demand trend as the indicator to do the assignment is one of the key points to enhance the system performance in the warehouse.

The research conducts two different simulations, i.e., simulation based on the decision rules for order-to-pod assignment and simulation based on the number of SKUs in a pod. For the simulation based on the decision rules for order-to-pod assignment, this research proposes the implementation of the likelihood rate for the decision rule on the improved scenario. The concept of this decision rule is to assign the order to the pod with the highest value of the likelihood rate of other SKUs in the same pod of the demanded SKU. This improved scenario will be compared with random decision rules on the baseline scenario. Meanwhile, the simulation based on the number of SKUs in a pod is conducted with five different scenarios, which apply the decision rules from the baseline scenario.

For the simulation based on the decision rules for order-to-pod assignment, the throughput rate result between the baseline scenario and improved scenario is analyzed using a two-sample T-test. The result indicates that the means between scenarios are significantly different. The average throughput rate between two scenarios also proves that the improved scenario can boost 18% of the throughput rate. Furthermore, the result analysis for simulation based on the number of SKUs in a pod is conducted with one-way ANOVA and Tukey’s range test. One-way ANOVA test result shows that not all population means are equals, while Tukey’s range test result shows both 5 SKUs and 10 SKUs have statistically higher means compared to 2 SKUs.

Moreover, allocating 10 SKUs in a pod produces the highest average throughput rate compared to the other scenarios. Meanwhile, the number of the delivered pod is reducing when having a higher number of SKUs in a pod. It shows there is an improvement when allocating a higher number of SKUs in a pod. Thus, the best SKU configuration in this research is allocating 10 SKUs in a pod.

## Future Research

There are various ways for improving the throughput rate in the Robotic Mobile Fulfillment System, as the variation of decision rule and parameters in this system will influence the throughput rate. For the assignment part, the selection of different decision rules on other assignments could be implemented. For instance, consider an excellent way to deal with the queue in picking stations for robot-to-pick station assignment.

The storage policy in the warehouse also can be another future research. Several suggestions for this part are SKU allocation in a pod, such as considering different pod has a different number of SKUs, considering the SKU popularity and pod allocation in the storage area considering its carrying SKU.

Another idea for future research is focusing on the replenishment process. It includes the mechanism on the replenishment process, how to assign the robot for the replenishment process, or both replenishment and picking process and the integration for the replenishment process and the picking process.



# References

1. Statista. *eCommerce worldwide*. 2020 [cited 2020 03/07]; Available from: <https://www.statista.com/outlook/243/100/ecommerce/worldwide>.

2. Absolunet. *10 eCommerce Trends 2020*. 2020 [cited 2020 03/07]; Available from: <https://10ecommercetrends.com/>.

3. Nils Boysen, R.d.K., Felix Weidinger, *Warehousing in The E-Commerce Era: A Survey.* European Journal of Operational Research, 2019. **277**(2): p. 396-411.

4. Rene de Koster, T.L.-D., Kees Jan Roodbergen, *Design and Control of Warehouse Order Picking: A Literature Review.* European Journal of Operational Research, 2007. **182**(2): p. 481-501.

5. Wulfraat, M. *A Supply Chain Consultant Evaluation of Kiva Systems (Amazonrobotics)*. 2012 [cited 2020 10/07]; Available from: <https://mwpvl.com/html/kiva_systems.html>.

6. John J. Enright, P.R.W. *Optimization and Coordinated Autonomy in Mobile Fulfillment Systems*. in *9th AAAI Conference on Automated Action Planning for Autonomous Mobile Robots*. 2011. California, USA.

7. Francielly Hedler Staudt, G.A., Maria Di Mascolo, Carlos M. Taboada Rodriguez, *Warehouse Performance Measurement: A Literature Review.* International Journal of Production Research, 2015. **53**(18): p. 5524-5544.

8. M. Merschformann, T.L., M.B.M. de Koster, L. Suhl, *Decision Rules for Robotic Mobile Fulfillment Systems.* Operations Research Perspectives, 2019. **6**.

9. M. Merschformann, L.X., H. Li, *RAWSim-O: A Simulation Framework for Robotic Mobile Fulfillment Systems.* Logistics Research, 2018. **11**(8).

10. Xi Xiang, C.L., Lixin Miao, *Storage Assignment and Order Batching Problem in Kiva Mobile Fulfillment System.* Engineering Optimization, 2018. **50**(11): p. 1941-1962.

11. Hazard, C.J., P.R. Wurman, and R. D'Andrea. *Alphabet Soup: A Testbed for Studying Resource Allocation in Multi-vehicle Systems*. 2006.

12. Sameera Abar, G.K.T., Pierre Lemarinier, Gregory M.P. O'Hare, *Agent Based Modelling and Simulation Tools: A Review of The State-of-Art Software.* Computer Science Review, 2017. **24**: p. 13-33.

13. Otten, S., et al., *Lost-customers approximation of semi-open queueing networks with backordering -- An application to minimise the number of robots in robotic mobile fulfilment systems.* arXiv: Probability, 2019.

14. Bipan Zou, Y.Y.G., Xianhao Xu, Zhe Yuan, *Assignment Rules in Robotic Mobile Fulfillment Systems for Online Retailers.* International Journal of Production Research, 2017. **55**: p. 6175-6192.

15. Tim Lamballais Tessensohn, D.R., Rene B.M. De Koster, *Inventory Allocation in Robotic Mobile Fulfillment Systems.* IISE Transactions, 2020. **51**(1): p. 1-17.

16. T. Lamballais, D.R., M.B.M. De Koster, *Estimating Performance in a Robotic Mobile Fulfillment System.* European Journal of Operational Research, 2017. **256**: p. 976-990.

17. Debjit Roy, S.N., Rene de Koster, Ivo Adan, Jacques Resing, *Robot-Storage Zone Assignment Strategies in Mobile Fulfillment Systems.* Transportation Research Part E, 2019. **122**: p. 119-142.

18. Nigam, S., et al. *Analysis of Class-based Storage Strategies for the Mobile Shelf-based Order Pick System*. 2014.

19. Yuan, Z. and Y. Gong, *Bot-In-Time Delivery for Robotic Mobile Fulfillment Systems.* IEEE Transactions on Engineering Management, 2017. **64**: p. 83-93.

20. Zou, B., et al., *Evaluating battery charging and swapping strategies in a robotic mobile fulfillment system.* Eur. J. Oper. Res., 2018. **267**: p. 733-753.

21. Nils Boysen, D.B., Simon Emde, *Parts-to-Picker Based Order Processing in a Rack-Moving Mobile Robots Environment.* European Journal of Operational Research, 2017. **262**: p. 550-562.

22. Krenzler, R.K., L. Xie, and H. Li, *Deterministic Pod Repositioning Problem in Robotic Mobile Fulfillment Systems.* ArXiv, 2018. **abs/1810.05514**.

23. Lin Xie, N.T., Ruslan Krenzler, Hanyi Li, *Introducing Split Orders and Optimizing Operational Policies in Robotic Mobile Fulfillment Systems.* European Journal of Operational Research, 2020.

24. Peng Yang, Z.Z., Huijie Guo, *Order Batch Picking Optimization Under Different Storage Scenarios for E-commerce Warehouses.* Transportation Research Part E: Logistics and Transportation Review, 2020. **136**.

25. Mu-Chen Chen, H.-P.W., *An Association-based Clustering Approach to Order Batching Considering Customer Demand Patterns.* Omega, 2005. **33**: p. 333-343.

26. Kaveh Azadeh, R.d.K., Debjit Roy, *Robotized and Automated Warehouse Systems: Review and Recent Developments.* Transportation Science, 2019. **53**(4): p. 917-945.

27. Marius Merschformann, L.X., Daniel Erdmann *Multi-Agent Path Finding with Kinematic Constraints for Robotic Mobile Fulfillment Systems*. 2018.

28. Merschformann, M. *Active repositioning of storage units in Robotic Mobile Fulfillment Systems*. in *OR*. 2017.

29. Steven F. Railsback, V.G., *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Second ed. 2019, Princeton, New Jersey: Princeton University Press.