



國立臺灣科技大學

工業管理系

碩士學位論文

學號：M10701814

以貨就人揀貨系統優化機器人行動交
通管制與協調之研究

Traffic Control and Coordination
for Robot Motion Optimization on
Robotic Mobile Fulfillment System

研 究 生：Agnes Yosephine Kosasih

指導教授：周碩彥 博士

郭伯勳 博士

中華民國一零九年七月



M10701814

Thesis Advisor: Shuo-Yan Chou



碩士學位論文指導教授推薦書
Master's Thesis Recommendation Form

Department : Department of Industrial Management

Student's Name: Agnes Yosephine Kosasih

Thesis title: Traffic Control and Coordination for Robot Motion Optimization on
Robotic Mobile Fulfillment System

This is to certify that the thesis submitted by the student named above, has been
written under my supervision. I hereby approve this thesis to be applied for
examination.

Advisor: Shuo-Yan Chou

Co-advisor: Po-Hsun Kuo

Advisor's Signature:

Date: 2020 / 07 / 23 (yyyy/mm/dd)



M10701814

Thesis Advisor: Shuo-Yan Chou

碩士學位考試委員會審定書

Qualification Form by Master's Degree Examination Committee

Department: Department of Industrial Management

Student's Name: Agnes Yosephine Kosasih

Thesis Title:

Traffic Control and Coordination for Robot Motion Optimization on Robotic Mobile Fulfillment System

This is to certify that the dissertation submitted by the student named above, is qualified and approved by the Examination Committee.

Degree Examination Committee

Members' Signatures:

周啟奇

郭伯勳

游慧光

Advisor:

周啟奇

郭伯勳

Program Director's Signature:

Department/Institute Chairman's Signature:

喻奉天

Date: 2020 / 07 / 23 (yyyy/mm/dd)

ABSTRACT

Robotic Mobile Fulfillment System (RMFS) is a parts-to-picker system that is well known to be implemented on e-commerce business. Despite its advantages to make the picking process more efficient by reducing ineffective picker activities such as traveling and searching, it consumes too much energy to move the Automated Guided Vehicle (AGV) as a transporter. Thus, traffic control and coordination are implemented on the routing policy to reduce AGV energy consumption.

NetLogo, an agent-based modeling simulation, will serve as a tool to model the picking process. Different from the previous study, where path planning is used for AGV routing, prevent a collision, and distance estimation for assignment, this study establishes a simple routing for AGV. Using three times Manhattan calculation for distance calculation, the assignment process can be improved more. Collision and deadlock prevention is solved using the agent-based modeling theory, where agents can communicate with each other and make a decision locally.

Implementing traffic policy in a warehouse intersection is done by storing the traffic condition at each intersection. Rather than deciding by its information and moving in FCFS manners, AGV will communicate with its surrounding agents and area to make a better decision in the traffic situation. Aisle's direction and robot condition will also be taken into consideration. Implementing this traffic policy proves to reduce the number of stop-and-go in traffic significantly by 11.4%, therefore minimizing AGV energy consumption.

Keywords: Robotic Mobile Fulfillment System (RMFS), Industry 4.0, Simulation, Routing Problem, Traffic in Warehouse, Energy Conservation

ACKNOWLEDGEMENT

Firstly, I would like to extend my sincerest gratitude to my advisor, Prof. Shuo-Yan Chou, who has supported and guided me throughout my research and thesis. His ideas, kindness, advice, and passion always inspire and motivate 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 would also like to thank all my lab mates in Information Technology Application and Integration (ITAI) laboratory for their friendliness and support every single day during these past two years. I would also like to give tons of thanks to my dearest classmates, roommates, and friends that have been with me through my journey in NTUST.

Furthermore, I must express my very profound gratitude to my parents and my siblings for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Last but not least, my most profound appreciation and praise goes out to Jesus Christ for letting me achieve another of my life accomplishments.

Agnes Yosephine Kosasih

Taipei, July 2020

TABLE OF CONTENTS

| | |
|---|------|
| Abstract | iii |
| Acknowledgement..... | iv |
| Table of Contents | v |
| List of Figures | vii |
| List of Tables..... | viii |
| 1. Chapter 1 Introduction | 1 |
| 1.1 Background and Motivation | 1 |
| 1.2 Objective and Limitation | 3 |
| 1.3 Methodology | 4 |
| 1.4 Organization of the Thesis | 4 |
| 2. Chapter 2 Literature Review | 6 |
| 2.1 Robotic Mobile Fulfillment System (RMFS)..... | 6 |
| 2.2 Agent-Based Modelling (ABM)..... | 9 |
| 2.3 Traffic Light Control and Coordination | 11 |
| 3. Chapter 3 Methodology | 13 |
| 3.1 Process Flow..... | 13 |
| 3.2 Simulation | 15 |
| 3.2.1 Layout | 16 |
| 3.2.2 Parameters and Assumptions | 17 |
| 3.3 NetLogo and Python Integration | 19 |
| 4. Chapter 4 Result and Discussion | 21 |
| 4.1 Baseline Simulation..... | 21 |
| 4.1.1 Decoding | 21 |
| 4.1.2 AGV Routing | 22 |
| 4.1.3 Distance Calculation | 30 |
| 4.1.4 Assignment..... | 32 |
| 4.1.5 Collision and Deadlock Prevention..... | 35 |
| 4.2 Traffic Policy..... | 37 |
| 4.3 Result Comparison | 40 |
| 5. Chapter 5 Conclusion and Future Research..... | 44 |

| | | |
|-----|-----------------------|----|
| 5.1 | Conclusion..... | 44 |
| 5.2 | Future Research | 45 |
| | References | 46 |

LIST OF FIGURES

| | |
|--|----|
| Figure 1.1 Picking System Classification [3]..... | 1 |
| Figure 2.1 RMFS Pick Order and Replenishment Process [13]..... | 6 |
| Figure 2.2 Hierarchical Overview of Decision Problem in RMFS [13] | 7 |
| Figure 2.3 Steps of Decisions to be Done Triggered by Received Order [15]..... | 8 |
| Figure 2.4 Structure of Agent-Based Modeling [26] | 10 |
| Figure 3.1 Sequence Diagram | 14 |
| Figure 3.2 Simulation Agents..... | 16 |
| Figure 3.3 Simulation Layout | 17 |
| Figure 3.4 System Architecture..... | 19 |
| Figure 4.1 Aisle Direction..... | 22 |
| Figure 4.2 Scenario One Starting and Ending Intersection..... | 24 |
| Figure 4.3 Scenario One Ending Intersection for Special Case | 25 |
| Figure 4.4 Scenario One Case One | 26 |
| Figure 4.5 Scenario One Case Two | 26 |
| Figure 4.6 Scenario One Case Three..... | 27 |
| Figure 4.7 Scenario One Case Four (Special Case) | 27 |
| Figure 4.8 Scenario Two Starting and Ending Intersection | 28 |
| Figure 4.9 Scenario Two Case One | 29 |
| Figure 4.10 Scenario Two Case Two..... | 29 |
| Figure 4.11 Scenario Two Case Three (Special Case) | 30 |
| Figure 4.12 Distance Calculation for General Case..... | 31 |
| Figure 4.13 Distance Calculation for Special Case..... | 32 |
| Figure 4.14 Assignment of the Previous Study..... | 33 |
| Figure 4.15 Assignment of This Study..... | 34 |
| Figure 4.16 Deadlock Condition | 36 |
| Figure 4.17 Deadlock Prevention..... | 37 |
| Figure 4.18 Traffic Control Importance | 38 |
| Figure 4.19 Traffic Detection in The Simulation | 39 |
| Figure 4.20 Boxplot of Number Stop-and-Go | 41 |
| Figure 4.21 Normality Test for Baseline Simulation Data..... | 42 |
| Figure 4.22 Normality Test for Traffic Policy Simulation Data | 42 |
| Figure 4.23 Two-Sample T-Test | 43 |

LIST OF TABLES

| | |
|---|----|
| Table 3.1 Parameters and Assumptions..... | 18 |
| Table 4.1 AGV Routing Scenarios | 23 |

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

In the past few years, with the headway in technology, the trend starts to shift from physical stores towards online stores or so-called e-commerce. The rise of e-commerce business changes not only customer behavior of purchasing items but also the entire order fulfillment process along the supply chain [1]. Therefore, it requires a new logistic approach, especially about the internal order picking system and the outbound delivery process [2]. Order Picking System (OPS) itself is classified into five main groups [3], which shows in Figure 1.1:

- "Picker-to-parts" system
- "Pick-to-box" system
- "Pick-and-sort" system
- "Parts-to-picker" system
- "Completely automated picking" system (e.g., dispensers)

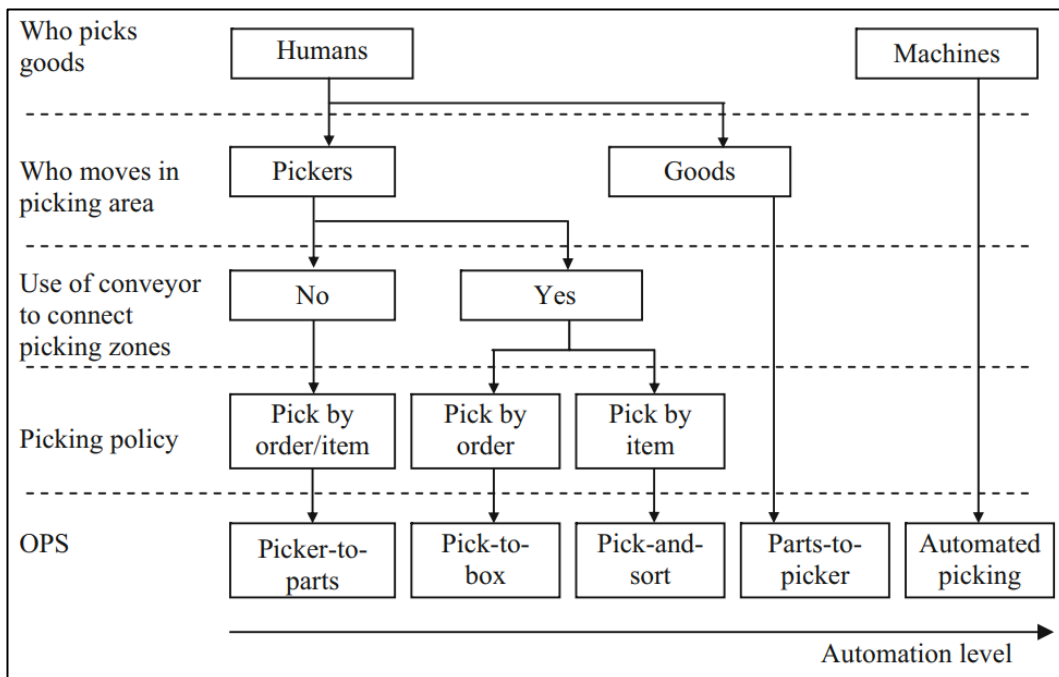


Figure 1.1 Picking System Classification [3]

The most traditional OPS is the picker-to-parts system, where pickers will walk or drive to pick the part throughout the aisle in the warehouse [4]. However, high demand from customer orders with a wide variety of items drives e-commerce to transform its warehouse order picking system into parts-to-picker. The development of warehouse automatic order picking system has overgrown to support the excessive demand of customers. A big boost has been given by the AVS/R (autonomous vehicle-based or shuttle-based storage and retrieval) systems or so-called Robotic Mobile Fulfillment System (RMFS) [5]. RMFS helps e-commerce to deal with peak demand during the holiday season (e.g., Black Friday) [6]. This system uses an Automated Guided Vehicle (AGV) to move the rack, which contains ordered items to the picker on the picking station, resulting in an automated picking process. Kiva robotic system is the first company that introduces this system that later on is acquired by Amazon. However, soon after, Amazon closes the information relating to the Kiva system.

Compare to the picker-to-parts system where workers play an active part to pick the item, parts-to-picker (specifically RMFS) use modern technology to reduce worker picking activities. RMFS is a perfect collaboration between humans and robots by taking advantage of both strengths. Initially, pickers need to do the picking process, including traveling and searching to reach the ordered items scattered in different pods throughout the warehouse. This traveling and searching are non-productive activities that consume more time and human energy [7]. One picker needs to travel back and forth through different aisles to complete an order with two (or more) different items.

On the contrary, AGV can carry pods that weight up to 450 kg (smaller pods) – 1300 kg (larger pods) [5]. Omitting not only waste of searching time, but also traveling time and picker energy. Also, making the picking activity more reliable, since machine or robot will work at a constant pace to maintain the service time and having a lower chance of getting wrong items [8, 9]. Although AGV speed is 3 miles/hours [10] (approximately the same with human walking speed), AGV works in an orderly manner and avoid collision with an excellent path planning through a software help. Moreover, a laser pointer that illuminates the pick location of the ordered item in the pod is helping picker to pick the item without searching [11].

Therefore, omitting the probability of false route choice, time to search for the items, and fatigue factors that may be caused by pickers. Hence, reducing the time significantly and increasing the throughput rate (unit of item deliver per several amounts of time) [6].

Despite its advantages, replacing the picker with robot consume too much energy. In its trip to move the rack, AGV performs several motions such as turning, lifting, stopping, and moving (includes accelerating and decelerating), which consume a lot of energy. AGV uses a battery as a source of energy. Through its movement, energy is transformed into heat energy then into mechanical energy within the engine. The mechanical energy is going to be converted into kinematical energy when AGV is accelerating. Once the AGV reaches its maximum speed, it will only consume energy to maintain the speed (overcome friction with the floor). While deceleration also consumes energy through its attempt to transform kinematic energy into heat through the break. That is why most of the vehicle consumes less fuel or energy while operating at a steady speed than in stop-and-go traffic [12].

In this study, a traffic policy will be implemented in an attempt to reduce stop-and-go in the traffic. Simulation platform will be used as a tool to monitor the AGV traveling time and stop-and-go number. A well-known traffic light control and coordination (an automated system for street traffic light) will be proposed to make the traffic well-organized, thus reducing stop-and-go. Furthermore, a way to avoid deadlock and collision is implemented to prevent a traffic jam in the warehouse. Overall, this study will focus on reducing energy by minimizing stop-and-go in traffic with minimum traveling time.

1.2 Objective and Limitation

This study aims to reduce the energy consumption of AGV by reducing its stop-and-go in traffic. Without a traffic control policy, AGV will move in first come first serve (FCFS) manners, meaning that whoever occupied that intersection first will force the others to stop to avoid a collision. Through the proposed traffic policy, AGV will move in more orderly manners, allowing it to have a priority of which one should move first rather than in random order. This proposed policy will serve

in similar to traffic light control and coordination, a system that is implemented in commonly seen street traffic light. Thus, minimizing stop-and-go by omitting uncertainty of movement between AGV.

Additionally, a simple rule in AGV routing is applied to achieve a minimum traveling distance in an attempt to minimize energy consumption. As known, besides accelerating and decelerating, moving will also consume a lot of energy. Proposing this simple rule in AGV routing will enable it to have a minimum path without doing path planning, which also reduces the computational time for calculation.

The limitation of this study is replenishment, and the charging process is not yet to be considered, thus focusing mainly on the order picking process. The attempt to minimize stop-and-go is not applied in picking station queues. It only applied in traffic on the intersection of the aisle. Turning time, lifting time, acceleration, and deceleration is also not considered in this study.

The contribution of novelty in this thesis is establishing a simulation platform, including a mechanism of traffic control and routing. Focusing on minimalizing the AGV movement, this system also enables to reduce energy consumption. This proposed system of traffic and routing will be explained further in Chapter 3.

1.3 Methodology

The study begins with an observation of the RMFS, its process flow, and each decision AGV needs to make. A simulation platform is built according to those automated warehouse order picking process. With this simulation platform, the proposed AGV routing rule will be implemented first as a baseline. Improvement will be made after that by implementing a traffic policy in the baseline. AGV stop-and-go and traveling distance data will be collected from the simulation. Finally, a statistical test will be done to prove the improvement by implementing the proposed system.

1.4 Organization of the Thesis

This study consist of five chapters. The organization of it is as follows:

Chapter 1 presents the problem background and motivation of the study, as well as the objective, limitation, and methodology; Chapter 2 explains the literature review of related work; Chapter 3 describes the simulation platform, its process flow, and parameters; Chapter 4 introduces the proposed improvement, including the result of it; and Chapter 5 wrap up the entire research with a conclusion and future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Robotic Mobile Fulfillment System (RMFS)

Robotic Mobile Fulfillment System (RMFS) is an automated parts-to-picker system that often implemented in the e-commerce warehouse. In RMFS, mobile robots are used to transport movable racks (so-called pods) that contains the inventory, back and forth between the storage area and the workstations, thus eliminating the need for the pickers to walk and search for the inventory [7, 13-15]. Firstly introduced by Kiva System, and was acquired by Amazon, this system proved to increase throughput capacity, reduce operational cost, and improve flexibility [16, 17].

The process flow of RMFS starts when orders enter the system. The system will search for pods that contain ordered items and ask a robot to pick the pod. The robot will bring it to a picking station, and after picker finishes picking the ordered items, the robot will bring the pod to the empty location. Pods which items quantity reach certain levels [18], will trigger the system to assign a robot to bring that pod into replenishment station. After the replenishment process finished, the robot will bring the pod back to the warehouse. The full process is shown in Figure 2.1 (arrow shows the movement of the robot and pod) [13].

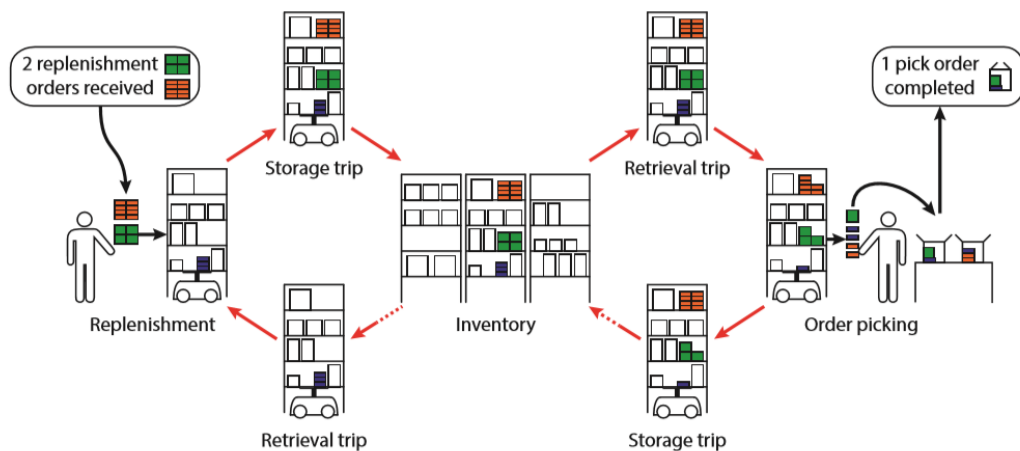


Figure 2.1 RMFS Pick Order and Replenishment Process [13]

RMFS is a flexible system, where pods do not have a fixed location, it can be located anywhere in the warehouse, as long as it is empty and not in the aisle [19]. The items stored in pods also vary with different loads and types [20], which means different types of items may be in the same pods, while the same type of item may also be located in different pods or scattered throughout the warehouse [21]. These open up a choice for the system to decide which pod to assign when orders arrive, and a location to return the pod. The system also needs to select a particular robot to pick selected pod. Those decision problems are classified as operational problems that need to be done in real-time (Figure 2.2).

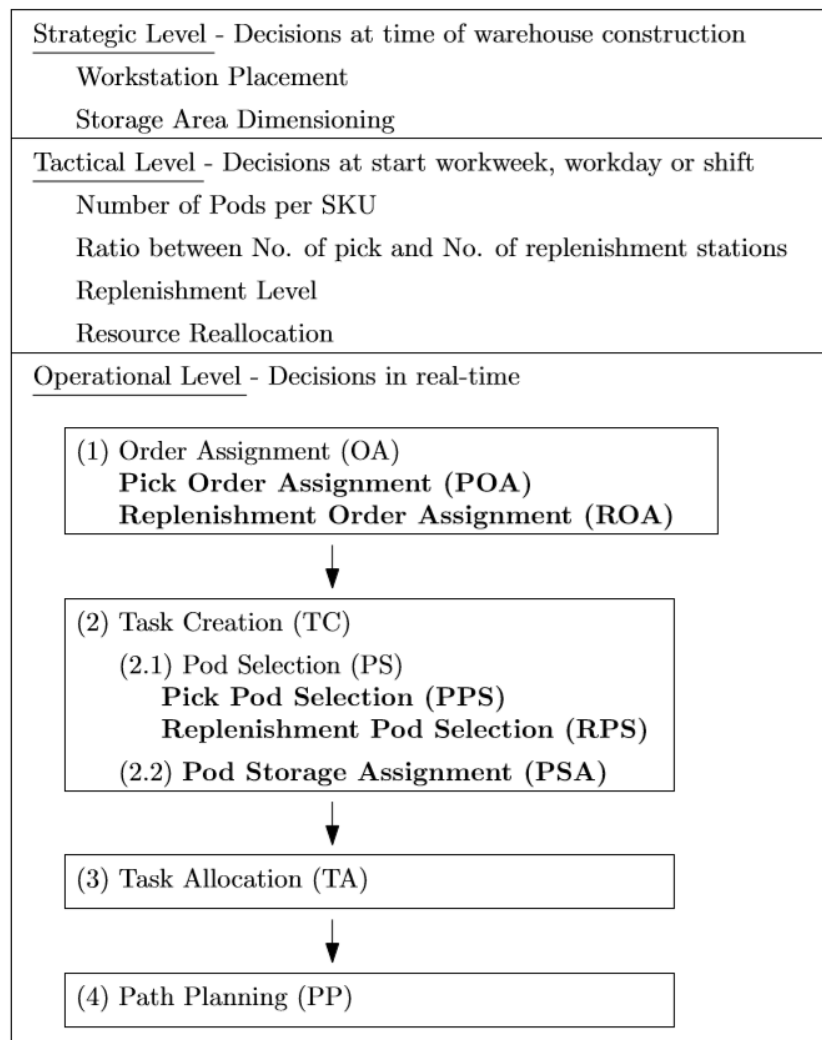


Figure 2.2 Hierarchical Overview of Decision Problem in RMFS [13]

As mentioned in Figure 2.2, robot path planning is also considered as an operational level decision problem. This problem also called the Multi-Agent Path Finding (MAPF) problem, which is a problem where a set of an agent (robot in this case) need to find a collision-free path for all agents with a different starting point to a different destination point [22, 23]. Some use a heuristic algorithm to solve these MAPF problems, and some use a metaheuristic algorithm [22]. It aims to find the most optimum path with the most minimum distance with no collision in the path.

Acceleration, deceleration, and also turning time are important factors that need to be considered in MAPF problems. It calculates the location for each agent in real-time so that in each timestamp, no robot is allowed to be in the same location. If the path makes robots to have a collision or deadlock, a different path that may have a longer distance will be selected. However, when there is no path left that has no collision or deadlock, the robot needs to wait for some time for the path to be free [22].

Selecting the best decision in an operational level decision problem will help to optimize the whole order picking and replenishment process to be effective and efficient. After that, deciding the number of pods needed, working hours of workers, and so on that mentioned in tactical level problem can be done. And finally, solving the strategic level of problems such as how big the warehouse is, the layout, and else can be determined. Thus, the overall goals of increasing throughput and minimizing the operational cost can be achieved [13].

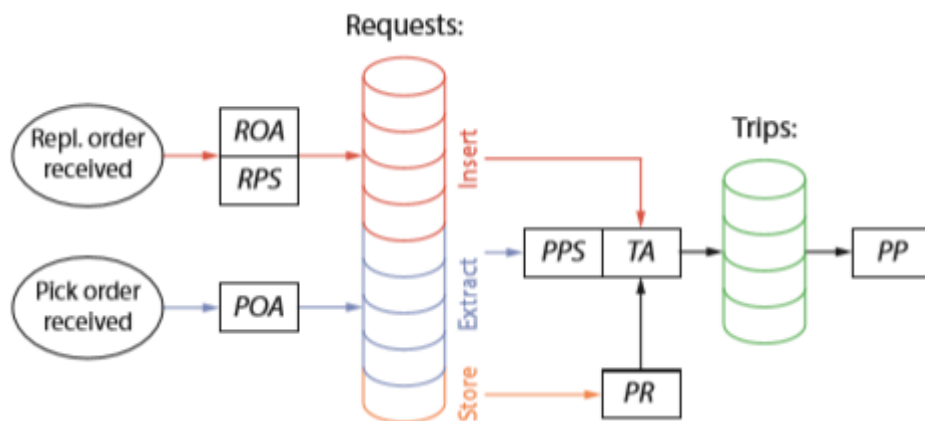


Figure 2.3 Steps of Decisions to be Done Triggered by Received Order [15]

Some research used a simulation framework to implement different decision problems to search for the best way to improve throughput and minimizing the operational cost. Figure 2.3 shows the decision-making steps when the order is coming in. The following decision problem is used in the decision-making steps [15] :

- Replenishment Order Assignment (ROA): assignment of replenishment order to replenishment stations
- Replenishment Pod Selection (RPS): a selection of pods to store one replenishment order
- Pick Order Assignment (POA): assignment of pick orders to pick station
- Pick Pod Selection (PPS): a selection of pods to fulfill orders at the picking station
- Task Allocation (TA) (for robots): assignment of tasks from the system
- Pod Repositioning (PR): assignment of an available storage location to a pod that needs to be brought back to the storage area
- Path Planning (PP) (for robots): planning of the paths for the robots to reach the destination

2.2 Agent-Based Modelling (ABM)

Agent-based modeling is a simulation modeling technique in which the system is modeled as a collection of autonomous decision-making entities called agents. It is best to simulate individual uncertain behavior, the interaction between them, and the adaption or learning process of individuals [24]. ABM becomes more critical as the complexity of the model increase, where the traditional approach is no longer as applicable as before [25].

A conventional agent-based model has three main elements, which are agents, agent relationships, and the environment [26]. Agents represent some specific social actors (individuals or groups) in the virtual environment [27]. It can interact with each agent (in Moore Neighborhood¹) and the environment itself on purpose to get some information and make a particular decision (Figure 2.3).

¹ a two-dimensional square lattice and is composed of a central cell and the eight cells that surround it

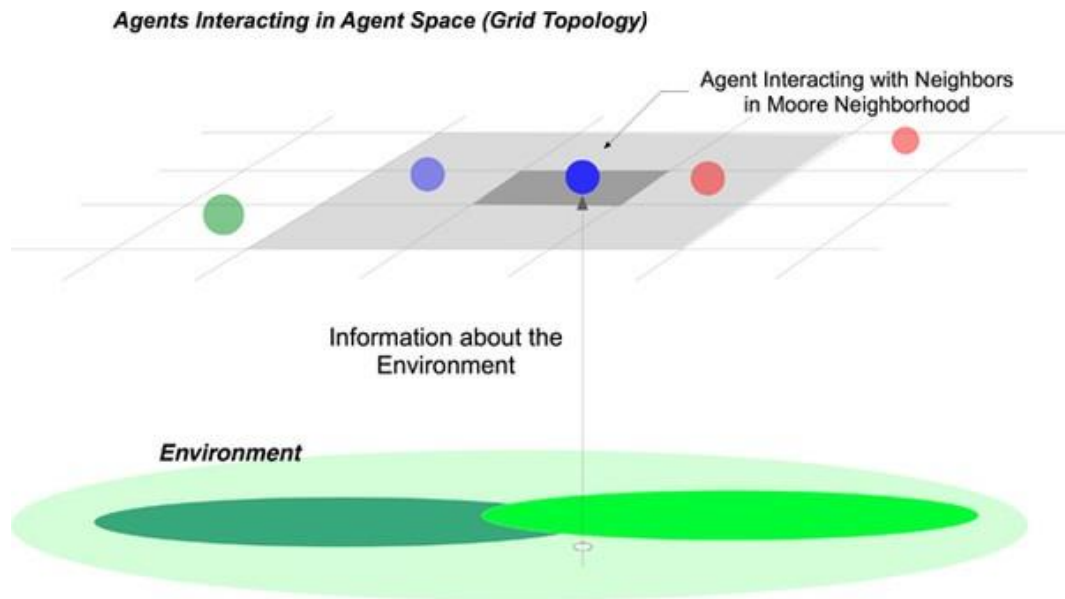


Figure 2.4 Structure of Agent-Based Modeling [26]

Each component in ABM serves a different purpose and has different behavior. All components detail describes as follow:

- Agents

Agents have an autonomous characteristic, where it can move on its own without external direction due to the situation it encounters. Thus, making it able to make a decision independently related to the information that each of the agents obtains. It can interact with each other and exchange some information that may lead to some changes. The different decision each agent execute will make it have various state over time [28].

- Agent relationships

An agent can change information locally with each other under specific conditions to improve its performance. The interaction is limited by surrounding neighbors of an agent, just as in the real-world system. A link is often used to represent a relationship between agents. Information usually spreads between agents rather than globally available in the system. Thus, it made ABM well known as a decentralized system instead of a centralized system [26].

- Environment

The environment is a virtual world where agents move or perform its decision, including where they interact with each other [27]. It may consist of a global variable, some structure, and location along with its specific information that agents can access. It may also collect agent behaviors and interactions. Thus, the state of ABM is the collective state of all the agents and the environment [28].

ABM simulation is often used to simulate the supply chain model, where agents serve as the customer and retailer, also using a link in ABM to represent the network between them [29]. This simulation also implemented in traffic control simulation. The complexity of traffic makes the use of ABM important, cause it can display human behavior. Through the visualization of it, particular congestion can be seen easily with different cases and initialization [30].

2.3 Traffic Light Control and Coordination

In large cities, with an increase in traffic and congestion, controlling the traffic becomes more and more critical. It will cause a queuing phenomenon for the best case, while for the worst one, it will degrade the use of the available infrastructure and reducing the throughput [31]. Thus, traffic signal systems are invented, using traffic light as a tool of signaling, not only controlling vehicles but also pedestrians around it [32]. There are some control strategies classifications for traffic signaling [31]:

- Fixed-time strategy and traffic-responsive strategy

In fixed-time strategy, several signal plans corresponding to different divisions of time are predetermined based on the historical traffic flow data [33]. Traffic light (red light or green light) may be longer or shorter due to the data of people habit in that area (e.g., working hours). This strategy seems to be inefficient because traffic is continuously changing caused by accident or uncertainty cases. Thus, the traffic-responsive strategy closes this gap by utilizing real-time traffic data to optimize the green time of a traffic light [34], so the time of green light will continuously change in

response to the current traffic.

- Isolated strategy and coordinated strategy

In an isolated strategy, traffic coordination will focus on the optimization of a single intersection, which is only suitable under undersaturated traffic conditions. Otherwise, coordinated control considers all the intersection and the entire urban area to optimize the whole system [34].

- Undersaturated and oversaturated

Undersaturated is a traffic condition where the queues of vehicles are only created due to red traffic light and dissolve during the green ones. While oversaturated is a condition with increasing queues. In many cases, these queues will reach the upstream intersections, making it becomes harder to handle. Hence, only a few strategies can be applied for the oversaturated condition.

CHAPTER 3

METHODOLOGY

The basic concept of RMFS that is shown in Figure 2.1 will also be implemented in the proposed simulation, without considering about replenishment. Netlogo is used as a simulation platform with the support of Python as an optimization platform. The simulation sequence, layout, assumption, and parameters will be explained in this chapter.

3.1 Process Flow

The order picking process starts with an order that comes to the system. Assuming the picking station can handle any orders, the assignment between orders to a picking station is omitted. Instead of assigning the order to the picking station, each order will be assigned to the pod. Different from the previous related work, where the order will be assigned to the picking station first, then the required pod will be called to fulfill this order. It is rather inefficient and makes unnecessary limitations because some assume that one picking station can only open up to three boxes to place an ordered item that might not be necessary to consider. Instead of doing that, the direct assignment from order to pod increase the possibility of optimizing the order picking process. Figure 3.1 shows the sequence of the whole process from the order is coming until it is fulfilled.

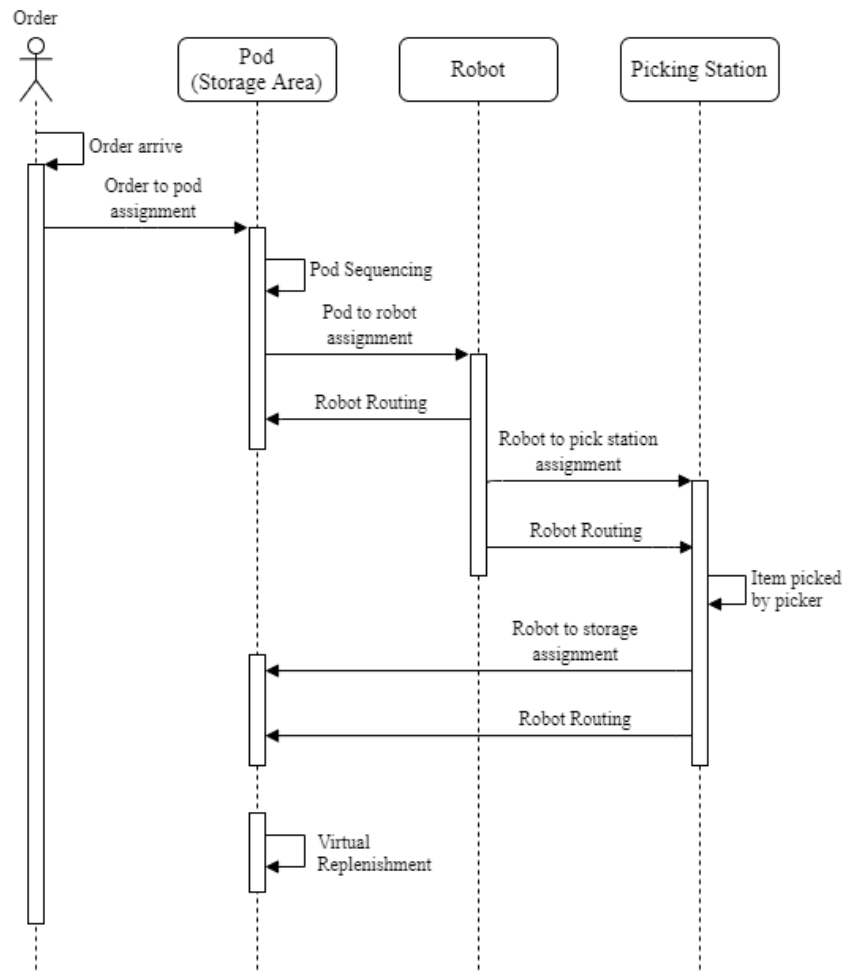


Figure 3.1 Sequence Diagram

Order will arrive continuously, and as soon as it arrives, it will be assigned to the pod immediately. The selected pod will contain information about items that the picker needs to pick according to the assigned order to fulfill it. Each selected pod can contain multiple orders. The earliest due date of those orders will be the due date of the selected pod. The list of the selected pod will be sequenced continuously according to the earliest due date, every time the order is assigned to a pod. Each selected pod due date will also be updated before sequencing the list of the pod.

Then the selected pod will be assigned to an available AGV. Taking two times the number of available AGV from the selected pod list, the assignment will be conducted between AGV and selected pod. This assignment will be based on the nearest distance. However, when the due date on a particular pod reaches one hour,

it will become a priority order. Thus, any available AGV will have to pick this pod without considering the distance between them. Applying a priority in order is essential to avoid any order exceeding its due date, which may cause a cost penalty to the company.

This study establishes a simple routing for the AGV, eliminating the need to use path planning. A simple calculation for robot routing will reduce the computational time of doing path planning. Taking an example of solving the path planning using brute force, it takes $O(n \cdot n!)$ time complexity [35], which makes it impossible when n is equal to 20 or even bigger. Notation n here represent the number of pod and AGV that are considered in the assignment process. Therefore, making the assignment very limited to a certain number of pod and AGV. However, establishing simple routing will make it possible to consider a large number of AGVs and pods.

Replenishment is not yet to be considered in the model, meaning that there will be no trip to replenish the pod. However, to refill the item in the pod that continually reduced due to the order picking process, virtual replenishment is used. Pods that reach particular safety stock will trigger count down to the fixed lead time. When the waiting time is done, items will be refilled instantly.

3.2 Simulation

The simulation platform is used to visualize the order picking process, capturing the movement of AGV, and possible collision also congestion. NetLogo that is well known as an agent-based modeling simulation, is used as the simulation platform. Using agent-based modeling simulation, NetLogo enables the agent to communicate with each other. AGVs can share the information and make a decision based on that without the intervention of the central system or can be said local decisions.

Agents that are used in this simulation is divided into static agents and moving agents. A static agent is an agent that stays on a specific location to perform tasks or provide information for other agents. On the contrary, a moving agent can move towards a determined destination and perform tasks during its movement. Figure 3.2 shows the main agents on this simulation and its attributes.

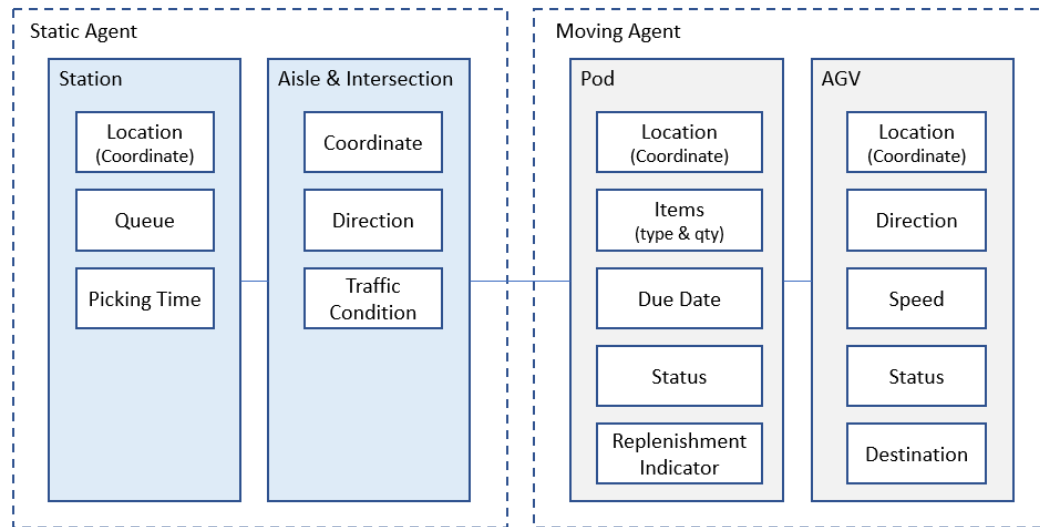


Figure 3.2 Simulation Agents

Attributes can be generated to store different information about the agents. Each agent has different attributes, which also might be different for each individual due to its current status. This attribute also can be updated continuously. Agents can also share information with other agents and with the simulation world or so-called global system. Sharing information between agents and the system helps to solve the problem that agents might encounter.

3.2.1 Layout

The simulation layout represents the warehouse on e-commerce, which has a picking station, replenishment station, also the storage area. The storage area will contain pods, empty storage locations, and the aisle where AGV moves. Figure 3.3 shows the simulation layout on NetLogo.

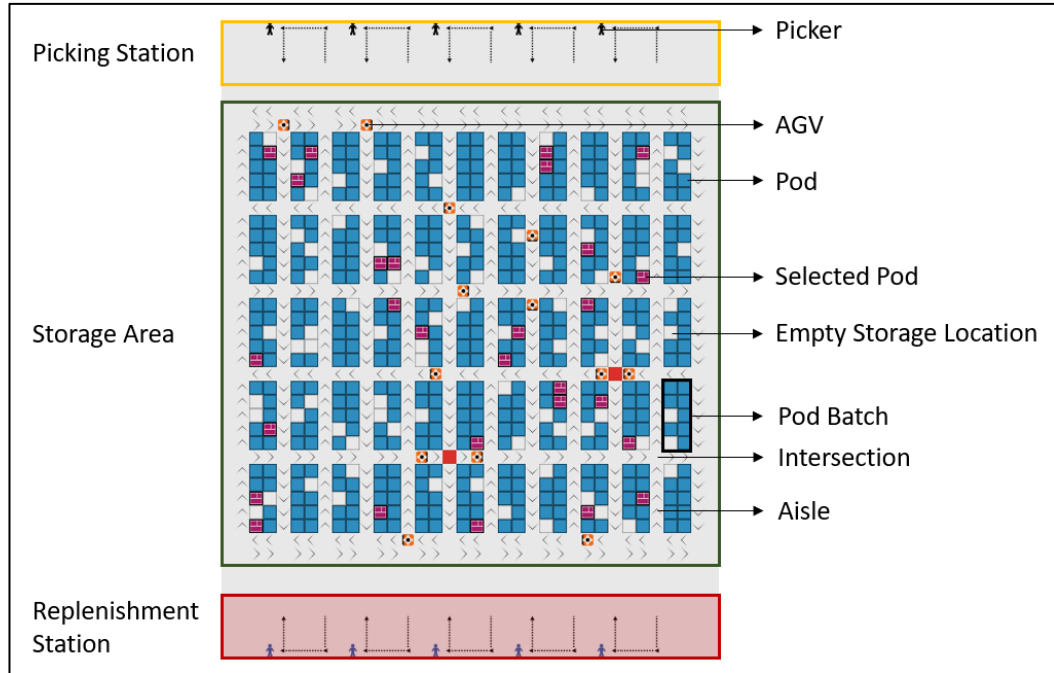


Figure 3.3 Simulation Layout

On top of the layout, there are picking stations with a picker on each station and queuing track. Below it, there is a storage area which contains pods that contain items in each. Different shape of the pod indicates the selected pod with orders that need to be picked by AGV later. There is also an empty storage location as a different option where AGV can place back the pod after it finished being picked. The aisle is determined to have a one-way direction. However, below the pod, there are no directions. On the aisle, AGV only has options to move forward or stop, while on the intersection, it can turn according to the direction of the next aisle it wants to go. On the bottom of the layout, there are replenishment stations that is not yet to be considered.

3.2.2 Parameters and Assumptions

Due to the limited access of information of the Kiva system, some of the system and parameter is assumed. Parameters and assumptions of the simulation are shown in Table 3.1.

Table 3.1 Parameters and Assumptions

| Parameter | Value |
|----------------------------|--|
| Simulation | |
| Run Length | 3 hours |
| Replication | 30 replication |
| Layout | |
| Inventory area | 550 storage location |
| Inventory capacity | 495 pod (90% of total) |
| Empty Storage Area | 55 empty storage (10% of total) |
| Pod Batch | 2 x 5 blocks |
| Aisles | 12 vertical aisles, 4 horizontal aisles |
| Hallway | 10 hallways (5 between each station) |
| Items | |
| SKU on pod | 2 SKU |
| SKU variations | 50 type |
| Quantity per SKU per pod | Poisson distribution, $\lambda = 15$ |
| Orders | |
| SKU frequency / popularity | Poisson distribution, $\lambda = 20$ |
| Interarrival | Exponential distribution, $\lambda = 0.05$ |
| AGV Movement | |
| Robot speed | 1 m/s |
| Time to lift and store pod | 1 s |
| Handling time of items | Poisson distribution, $\lambda = 15$ |
| Replenishment | |
| Safety Stock | 50% of total items on a pod |
| Lead Time | 100 s |
| Stations | |
| Pick station capacity | 5 stations |
| Queuing per station | 5 AGV |

The previous study [18] mentioned that it is not stable to replenish the pod when it is empty, meaning that it may cause a low throughput rate. Doing replenishment every time a pod visits the picking station (some item being pick there) make replenishment process utilization low. Because it only replenishes a few items but wastes a lot of traveling time. Therefore, replenish the pod when it is half empty is more stable and better than when it is empty or every time it visits the picking station.

3.3 NetLogo and Python Integration

The main software for running this simulation is NetLogo, where visualization and output are shown. Some functions to solve the problem will also be executed directly on NetLogo. Its advantage as agent-based modeling, which makes the agents able to communicate with each other, is beneficial to solve some problems. However, NetLogo expertise is on agent-based simulation, which is a bit hard to perform some optimization algorithm. Thus, Python is used as optimization software to support the calculation of the heuristic approach that is used. The system architecture where NetLogo and Python collaborate is shown in Figure 3.4.

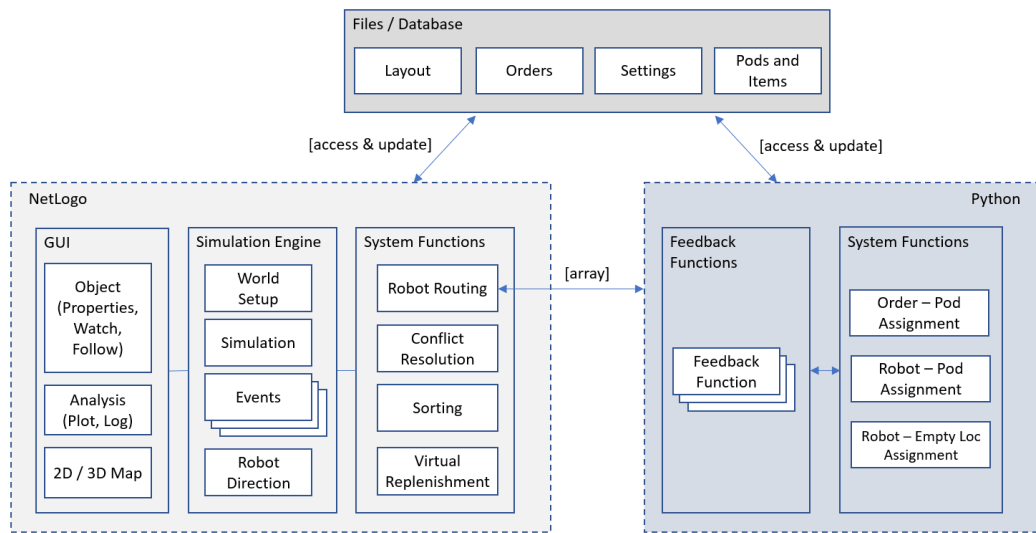


Figure 3.4 System Architecture

NetLogo can call the function and get feedback directly from Python. It can send an array as input for Python function, while Python also can send the result through an array to NetLogo. However, when the information is too large, a file or database is more convenient to use. This simulation uses Excel as a software to store and transfer the data. Storing data on Excel also helps to keep a history of the entire simulation process. Both NetLogo and Python can access and update the data on Excel directly.

Graphical User Interface (GUI) on NetLogo helps to visualize the entire order picking process, including the AGV and pod movement. The output will be shown as a plot or number on the GUI and will be updated automatically. GUI on

NetLogo also provides information about agents that can easily be accessed through selecting the agent by clicking it. This information is also updated automatically while the simulation is running.

CHAPTER 4

RESULT AND DISCUSSION

In this chapter, the baseline of the simulation will be explained in detail, including the routing and assignment. Implementation of the traffic policy as an improvement also will be shown. Furthermore, the result of improvement, as well as the analysis, will be described.

4.1 Baseline Simulation

The detail of baseline simulation, especially about the AGV routing and assignment, will be emphasized further. AGV, as a simulation agent, needs to know the whole map (layout), including where to turn since the aisle is determined to have a one-way direction. After knowing the information, AGV can decide a routing by itself. A distance calculation can be done to be used on the assignment part.

4.1.1 Decoding

AGV needs to understand the layout of the simulation. Giving an intelligent to the AGV, it needs to differentiate the pod or storage location and the aisle on the warehouse. Each patch (a block with specific coordinate) will be given an attribute as information of that location. That attribute contains a purpose for that location, such as a storage area where the pod and empty location is, aisles, intersections, and stations. AGV will ask the attribute of every patch first before performing a movement. It will prevent the AGV to drop a pod on the aisle or moving in the wrong aisle direction. Figure 4.1 and Figure 4.2 shows how AGV read aisle direction based on the patch coordinate.

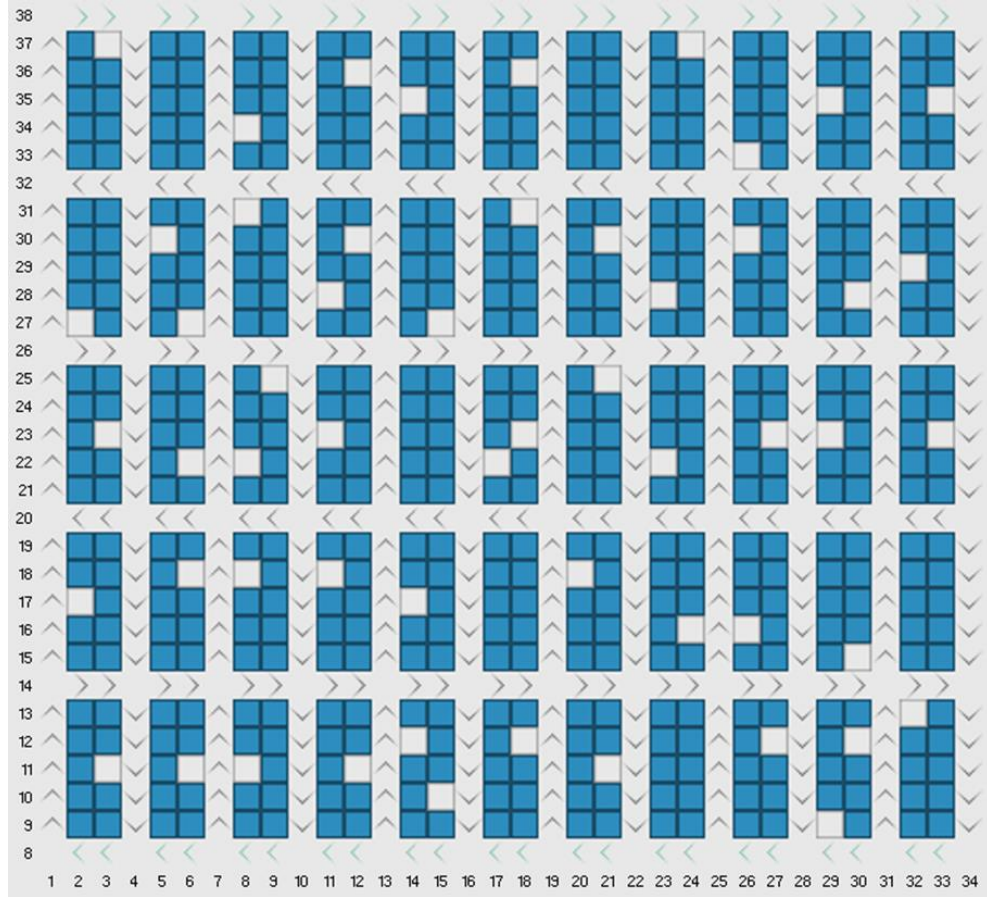


Figure 4.1 Aisle Direction

The Y-coordinate of the patch will determine the right and left direction of the aisle. There are replenishment stations below the storage area, which is why the Y-coordinate seems to start not from zero in Figure 4.1 and Figure 4.2. The directions are determined by Y-coordinate mod four. If it is equal to two, then the aisle is going to the right, and if it is equal to zero, then it is going to the left. The X-coordinate is determining up and down direction by odd and even. Odd X-coordinate is up direction aisle, and the even one is down direction aisle. Not only reading the coordinate of the patch, but AGV also needs to ask the attribute of the patch and make sure it is not a storage area or station.

4.1.2 AGV Routing

AGV needs to search for its route when moving from the initial location to the determined destination. The routing differentiates into three scenarios, which are routing to pick the pod, to deliver it to the picking station, and to bring the pod

back to the storage area. Each scenario has different things that need to be considered before determining the routing. Table 4.1 shows different scenarios of AGV routing.

Table 4.1 AGV Routing Scenarios

| <i>Scenario</i> | <i>1</i> | <i>2</i> | <i>3</i> |
|--------------------|----------------|-----------------|--------------------------|
| <i>Process</i> | Picking Pod | Bring Back Pod | Deliver Pod to be Picked |
| <i>Initial Loc</i> | Empty Location | Picking Station | Selected Pod |
| <i>Destination</i> | Selected Pod | Empty Location | Picking Station |
| <i>Carry Pod</i> | No | Yes | Yes |
| <i>Routing</i> | Underneath Pod | Aisle Only | Straightforward |

The third scenario, when AGV carries the pod to the picking station, is straightforward. AGV only needs to identify the aisle it is in, whether it is upward or downward. If it downward, AGV needs to take turns in the nearest intersection, searching for the upward aisle then go directly to the picking station. Because the routing on the third scenario is simple without complicated cases there, it will not be analyzed further.

Through all the cases of the scenarios, a pattern of AGV movement can be seen and analyzed. Establishing a new rule for AGV routing also made the route automatically be generated by Netlogo as the simulation platform. Four steps are established to determine this AGV routing. These steps will also be used to calculate the distance later. Those four steps are:

- Determine the scenario (whether it can move underneath the pod or not)
- Set the starting intersection
- Set the ending intersection
- Identify if it is a special case or regular case

The first step is to determine the scenario or the state of AGV. It can be done by asking its attribute since AGV will record the process it is doing at the current time. It also can be done by seeing the AGV coordinate, whether it is at the picking station or the storage area. Different scenarios will affect the setting of the starting intersection, also the ending intersection.

The first scenario, where AGV is in the empty location and about to pick the selected pod, which has a specific location, is allowed to go underneath the pod. AGV is allowed to go underneath the pod since it does not carry any pod. However, to minimize the traffic and avoid unsolvable deadlock underneath, AGV is just allowed to go on the horizontal direction underneath the pod within just one pod batch.

Setting the starting intersection for the first scenario considers the pod it needs to take. If the destination is on the top of the AGV, it will search for the upward aisle. Otherwise, if it is on the bottom, it will search for the downward aisle. Going underneath the pod gives AGV the flexibility to choose a different aisle direction without making a u-turn within one pod batch. Figure 4.2 shows how AGV go to the desired aisle directly by just turning around or go straight underneath the pod. Then, the starting intersection will be the first one it meets.

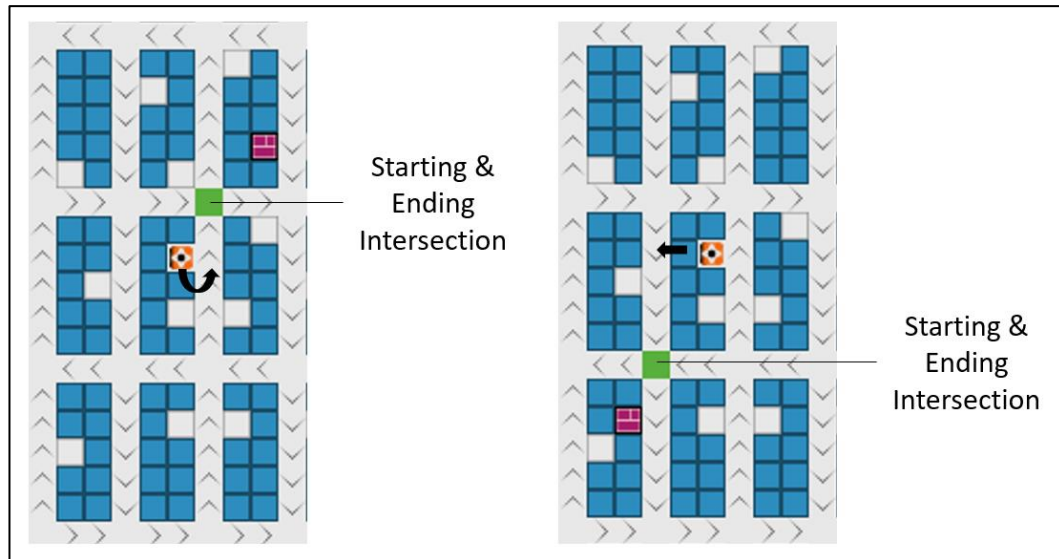


Figure 4.2 Scenario One Starting and Ending Intersection

Figure 4.2 is a simple case where the ending intersection is determined mainly based on the AGV position. Since the AGV can go underneath the pod, the aisle near the selected pod does not need to be considered. However, there is a special case which made the ending intersection different. Figure 4.3 shows the ending intersection for the special case.

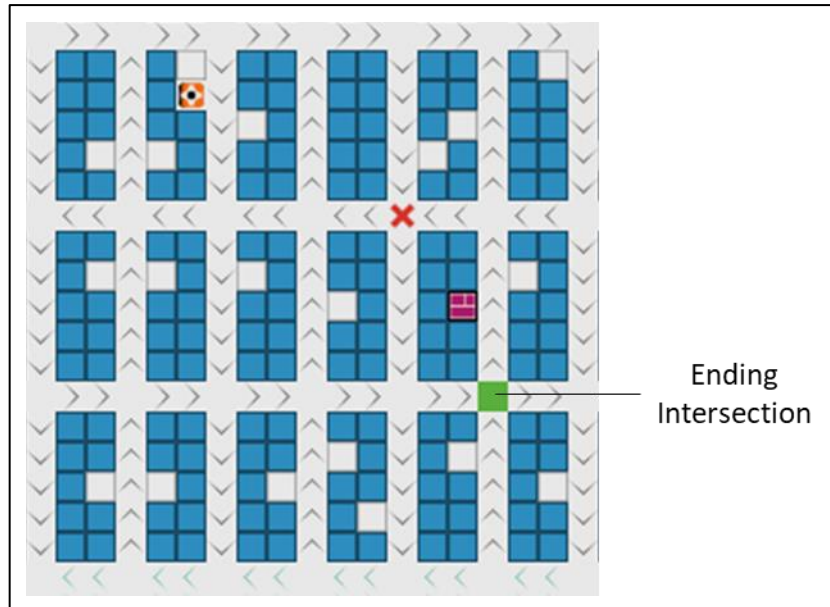


Figure 4.3 Scenario One Ending Intersection for Special Case

Consider only the position of the AGV, ending intersection should be on the intersection with the X mark in Figure 4.3. However, the ending intersection is changed because the movement of AGV is not matched with the aisle direction. AGV should be moving to the right, but the aisle direction is going to the left. The other reason is there only an aisle between the AGV and the selected pod. Therefore, there is no option to turn in another aisle. Exhausting all the cases of AGV routing for scenario one, it is summarized in four cases. The cases for scenario one are shown in Figure 4.4, Figure 4.5, Figure 4.6, and Figure 4.7.

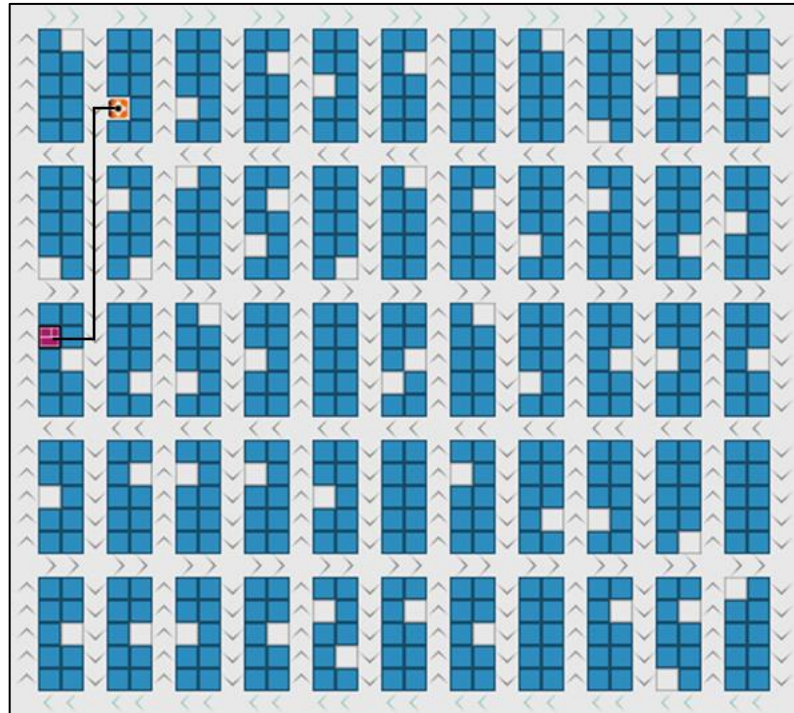


Figure 4.4 Scenario One Case One

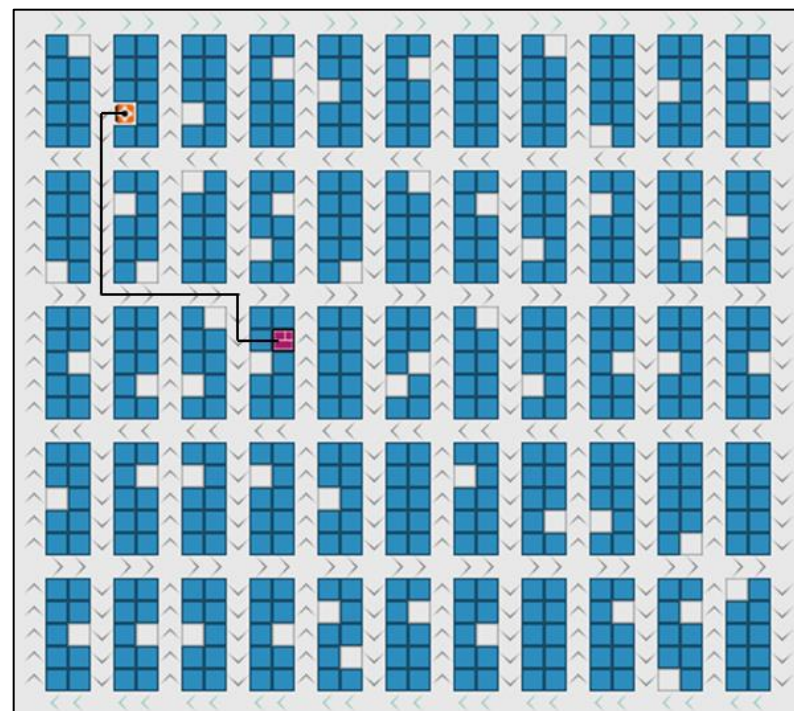


Figure 4.5 Scenario One Case Two

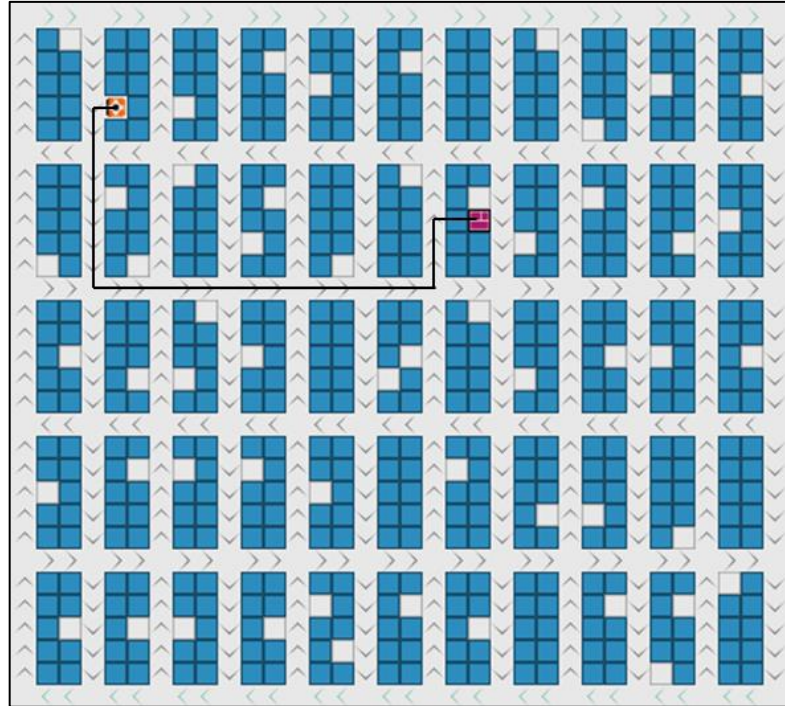


Figure 4.6 Scenario One Case Three

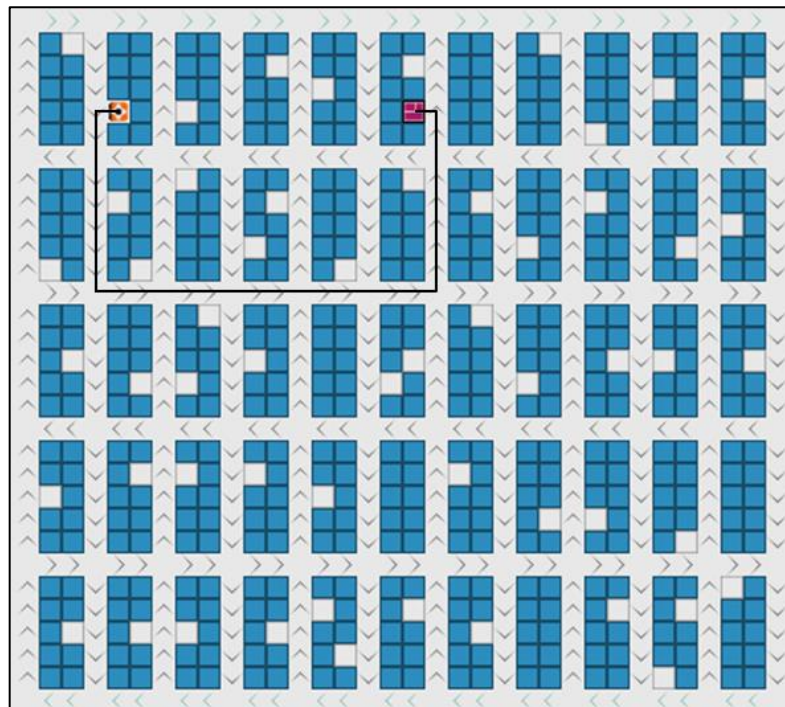


Figure 4.7 Scenario One Case Four (Special Case)

The second scenario is when AGV brings the pod back to the storage area, after the pod its carry is being picked at the picking station. AGV can not go

underneath the pod because it is still carrying the pod. Go underneath will cause another pod hitting the pod it is carrying. Thus, the movement will be limited in the aisle and intersection.

The way of determining the starting intersection is different from the first scenario. In the second scenario, AGV always starts from the picking station queue on the top of the layout. Therefore, the starting intersection will always be the intersection below it straightly. The ending intersection mainly considers the aisle next to the selected empty location. Figure 4.8 shows the starting and ending intersection for the second scenario.

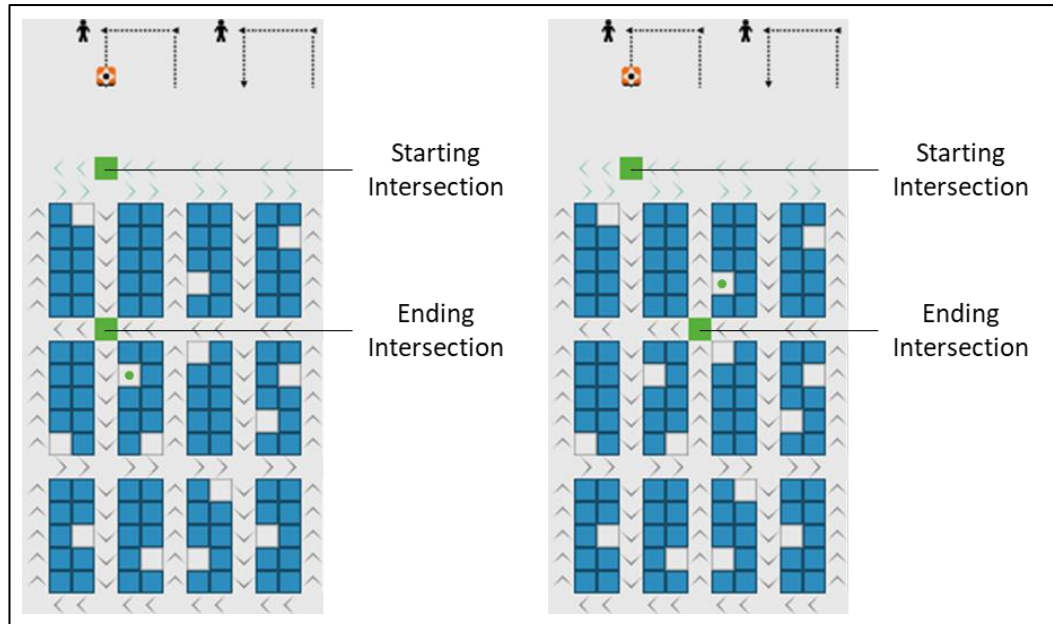


Figure 4.8 Scenario Two Starting and Ending Intersection

Different from the first scenario, where determining the ending intersection depends on several factors, the second scenario has a fixed rule to determine the ending intersection. However, this scenario also has a special case where AGV needs to make a u-turn before arriving on the selected empty location (marked by a dot in Figure 4.8). All the possible routing will be exhausted and summarized into three cases. Figure 4.9, Figure 4.10, and Figure 4.11 show three cases of scenario two. Figure 4.11 also shows a special case in this scenario.

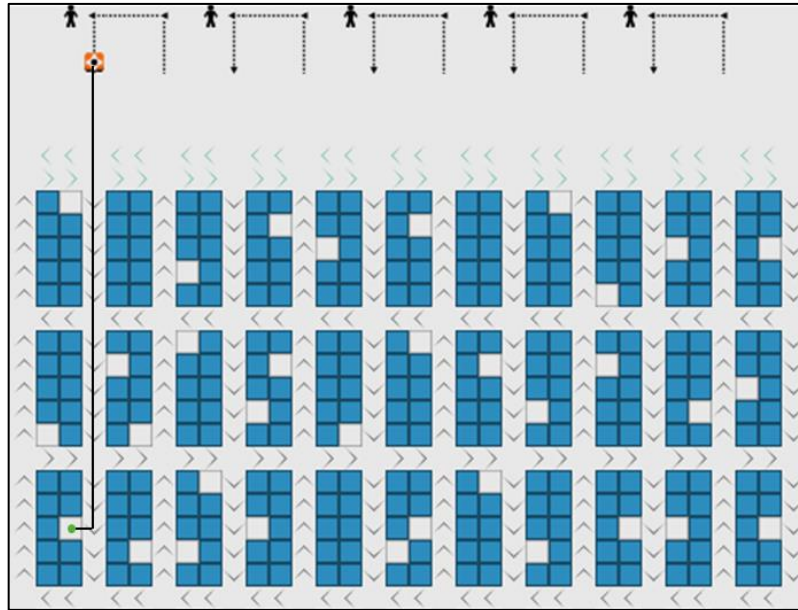


Figure 4.9 Scenario Two Case One

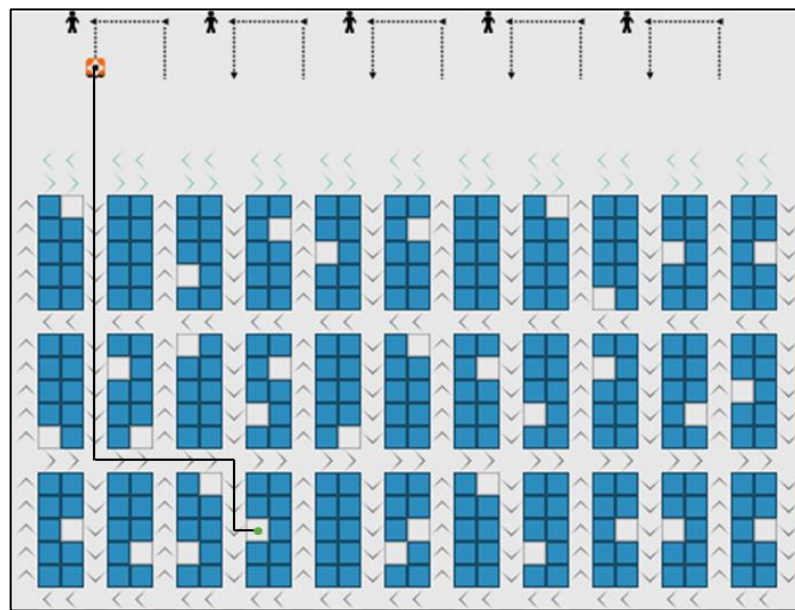


Figure 4.10 Scenario Two Case Two

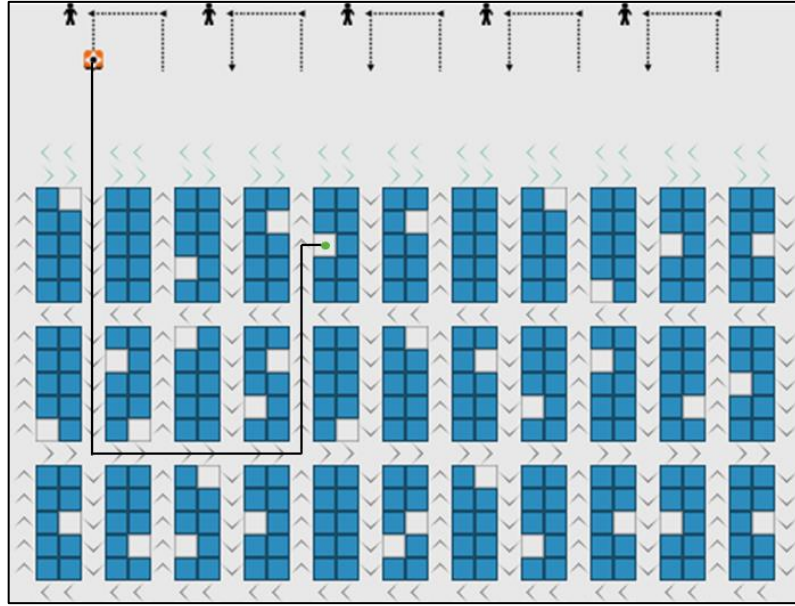


Figure 4.11 Scenario Two Case Three (Special Case)

Special case for scenario two occurs under two conditions. The first one is when the aisle next to the empty location is in the upward direction. The second one is the aisle below that upward aisle is on the opposite with AGV desired movement. In Figure 4.12, the aisle below the destinate pod batch is going to the left, making AGV need to go on extra miles to turn to the right.

4.1.3 Distance Calculation

A formulation of a simple distance calculation is established based on all the cases for both scenarios. Manhattan distance calculation is a fundamental distance function that is used in this calculation. The distance calculation is done by doing a calculation of Manhattan distance three times, then adding the result together. The origins and destinations that will be calculated using the Manhattan distance are:

- AGV origin location and the starting point
- The starting point and the ending point
- The ending point and destination (selected pod or selected empty location)

Different from the general case, the special case needs additional distance because of its different routing. On the special case for both scenarios, AGV needs

to makes a u-turn. Therefore, the distance of the u-turn needs to be calculated to get a precise distance. The difference between the general case and special case is shown in Figure 4.12 and Figure 4.13.

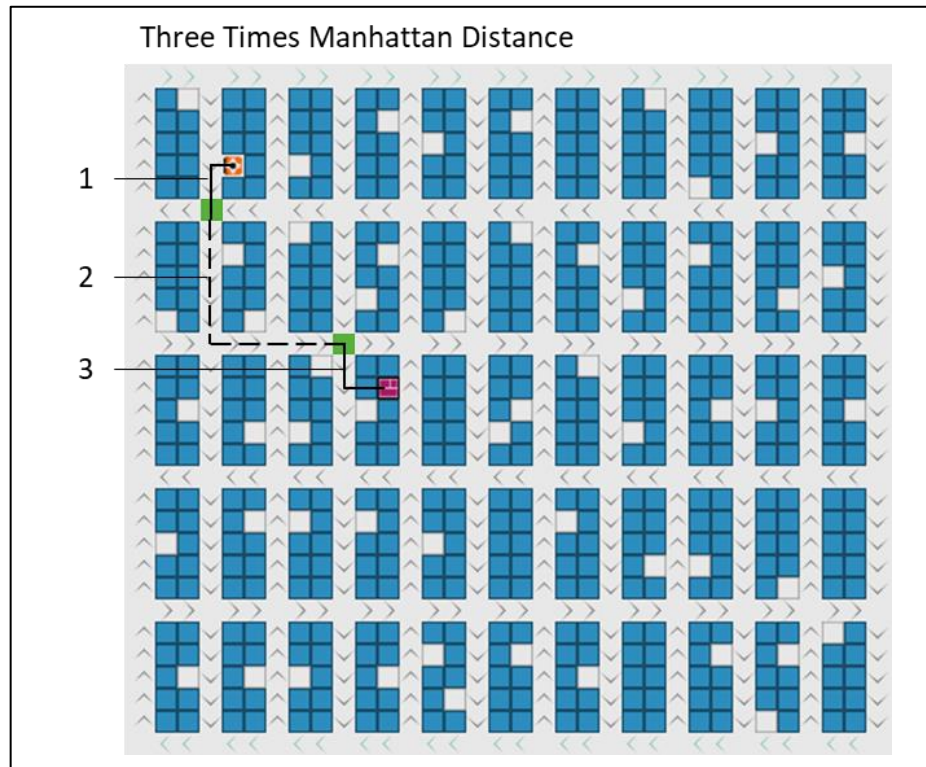


Figure 4.12 Distance Calculation for General Case

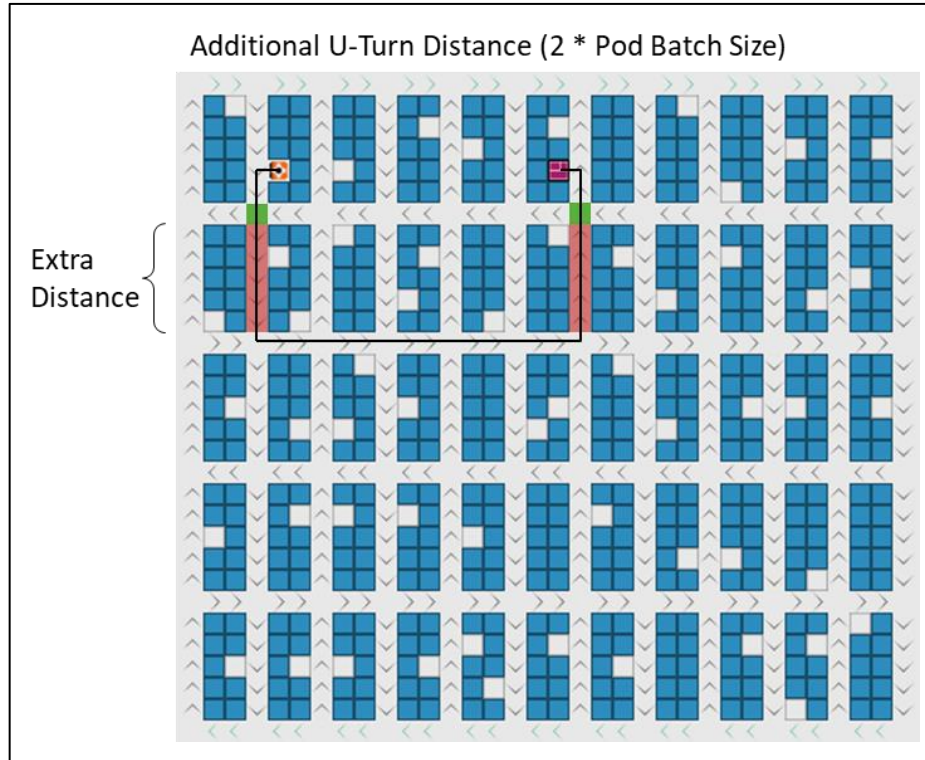


Figure 4.13 Distance Calculation for Special Case

The additional distance that needs to add to the distance calculation is two times the pod batch size. Therefore, as mentioned before, types of AGV routing cases need to be identified first before calculating the distance. After calculating a precise distance between AGV and its destination, an assignment can be done for picking the pod and bring the pod back.

4.1.4 Assignment

Different from the previous study, where the assignment is done between one picking station and several numbers of pods. The AGV is usually serving only one specific picking station. For the nearest assignment rule of the previous study, distance is calculated using path planning to estimate the traveling time. Illustration of the assignment from the previous study is shown in Figure 4.14.

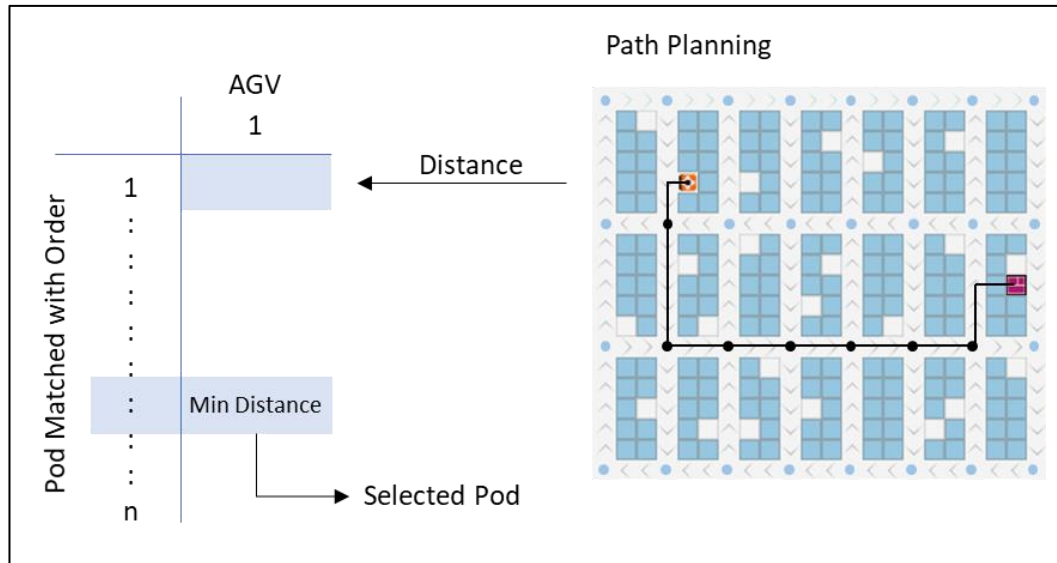


Figure 4.14 Assignment of the Previous Study

The assignment is done to search for the pod to be picked by a specific AGV. The decision is made by selecting the pod with minimum distance from n number of pods that matched with the order. Calculating distance using path planning is possible because it runs for only n number of times, although path planning will consider every node in the warehouse. However, this study aims to do a better assignment process by considering many available AGV. By considering many available AGV, distance can be minimized furthermore. Figure 4.15 shows how the assignment process is done in this study.

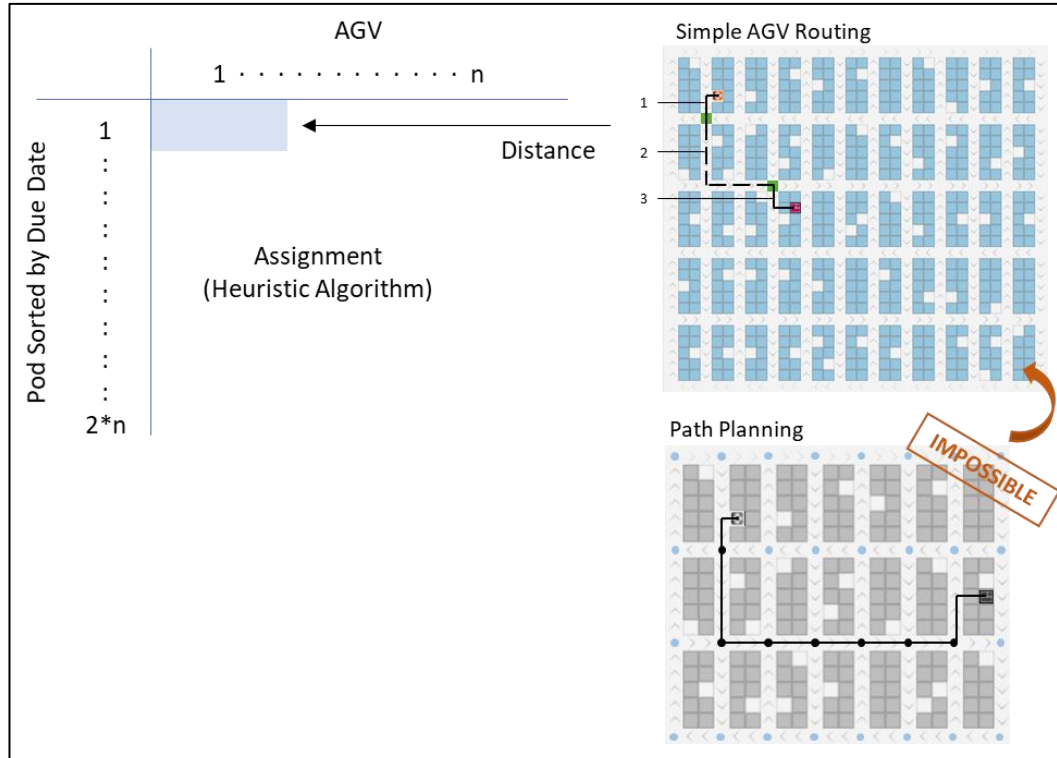


Figure 4.15 Assignment of This Study

The assignment is done for n number of AGV that is available or soon to be available. The pods that are going to be assigned is taken from the list of the selected pod. A list of the selected pod will continuously sort based on its due date, then two times the number of available AGV is taken into consideration. The same process will be done for the assignment to the empty location. The list of empty locations that will be considered is based on every pod batch starting from the nearest one with the picking station. If the number reaches two times n , then it will be stopped. However, if it is less than two times n , the next pod batch will also be taken into consideration.

Here, simple AGV routing to calculate distance through doing three times of Manhattan distance calculation is used instead of using path planning. Using path planning is not possible because path planning itself is considered to be a greedy algorithm. At the same time, a heuristic algorithm, which is also a greedy algorithm, will be used to solve the assignment problem. Doing two steps greedy algorithm will consume too much computational time to solve just one assignment problem at a time. Where in this simulation, the assignment is done continuously every time

AGV is performing another task. Thus, using path planning to calculate the distance will make the running time of the simulation becomes neverending.

The Hungarian method is used to solve the assignment problem in this simulation. Hungarian method is a commonly used method to solve the assignment problem that considers about opportunity cost. Both assignments for AGV to pick the pod and bring the pod back to the storage area will be solved using the Hungarian method. However, the assignment for AGV to the picking station is straightforward, so it does not need to use the Hungarian method to solve it.

4.1.5 Collision and Deadlock Prevention

Having a collision is possible when AGV is moving throughout the warehouse. Collision in the aisle can be happening when there is a stopping AGV, and another AGV which move in the same direction keeps moving. Another case is when two AGV is approaching an intersection at the same time. Therefore, AGV needs to detect whether there is AGV in front of it or not.

The advantage of using NetLogo, which is an agent-based modeling simulation, is each agent can communicate with each other. Using the ability to communicate with each other, AGV will continuously ask the patch in front of it. If an AGV is occupying that patch, the AGV behind it will stop moving so that it will not collide with each other. Similar to that, when AGV is occupying the intersection, the approaching AGV will stop moving.

Using the same logic, AGV is able to stop when queuing and moving underneath the pod. However, the deadlock may occur when AGV is moving underneath the pod because there is no direction underneath the pod. AGV may go to the left while another one is going to the left, causing a deadlock. Figure 4.16 shows the deadlock condition that may occur in the simulation.

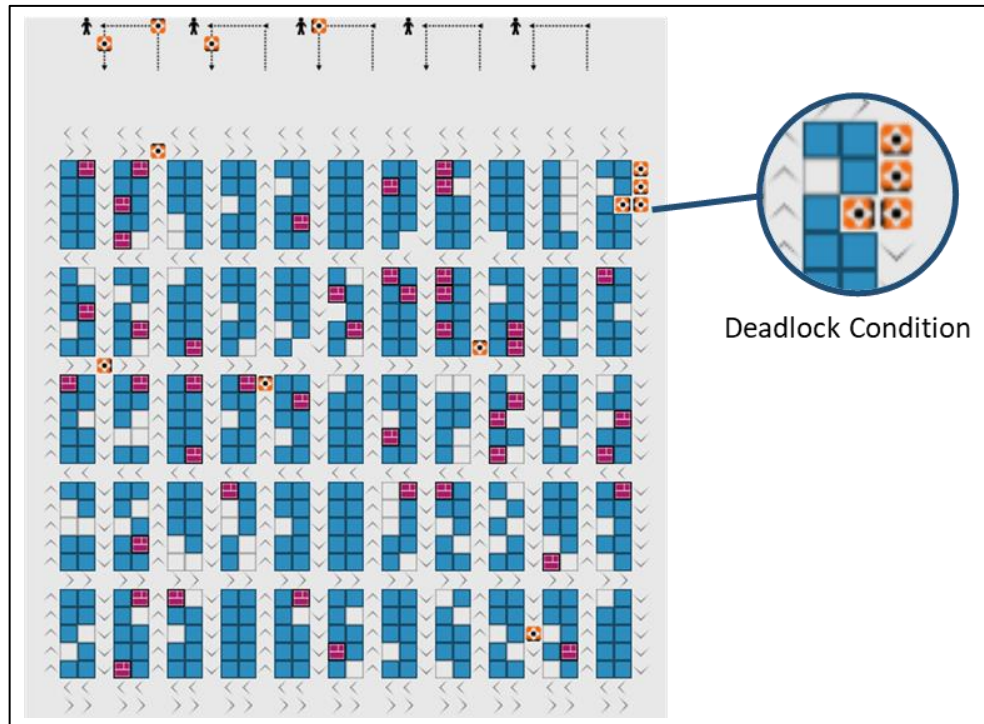


Figure 4.16 Deadlock Condition

Deadlock not only will cause both AGV to stop but also blocking the aisle. Another AGV may be stuck in the same place if the deadlock is not resolved. There is an option of turning around and taking another route to solve the deadlock. But turning around will cause an extra distance for AGV. AGV may also be stuck in another deadlock while turning around in a different direction. Thus, as a benefit for using NetLogo, agents can also give a signal to each other through communication. Figure 4.17 shows how AGV gives signals to each other to prevent deadlock conditions.

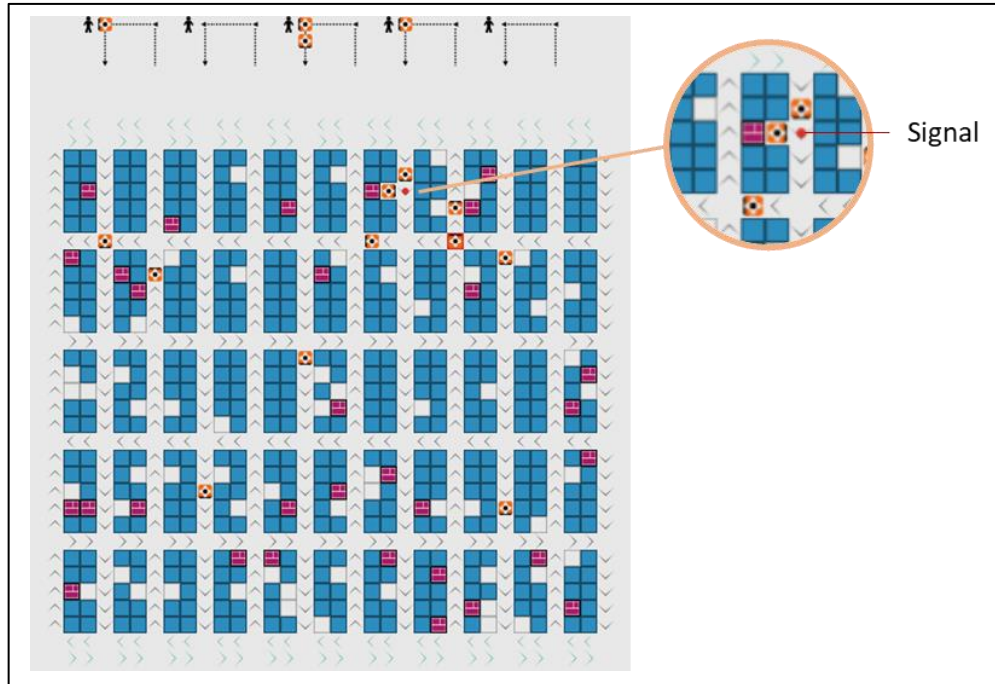


Figure 4.17 Deadlock Prevention

The AGV that is going out from underneath the pod will give a signal in the aisle beside it. The signal will stop approaching AGV to move, which may block the road and cause a deadlock. It will allow the AGV who is below the pod going out first. After the AGV is going out, the signal will disappear, allowing another AGV to move normally.

4.2 Traffic Policy

Traffic policy is implemented in the simulation as an improvement for the baseline simulation. Traffic light control and coordination, which is implemented in the commonly used traffic light, is used as a fundamental principle for this improvement. Then, this principle is developed furthermore to be more suitable for the simulation.

On the baseline simulation, the traffic might happen in the intersection and queuing area. However, traffic in the queuing area is not considered because AGV routing or movement will not be able to minimize the traffic there. The assignment problem is the one that causes AGV to queue.

In the intersection, AGV will move based on the First Come First Serve (FCFS) manners. It means that whoever occupied the intersection first will force the other approaching AGV to stop moving. This situation is seen as a problem to cause traffic. When there is no traffic control, AGV does not know the surrounding traffic in that intersection and making a decision based on its state. Figure 4.18 shows the difference if traffic control is implemented.

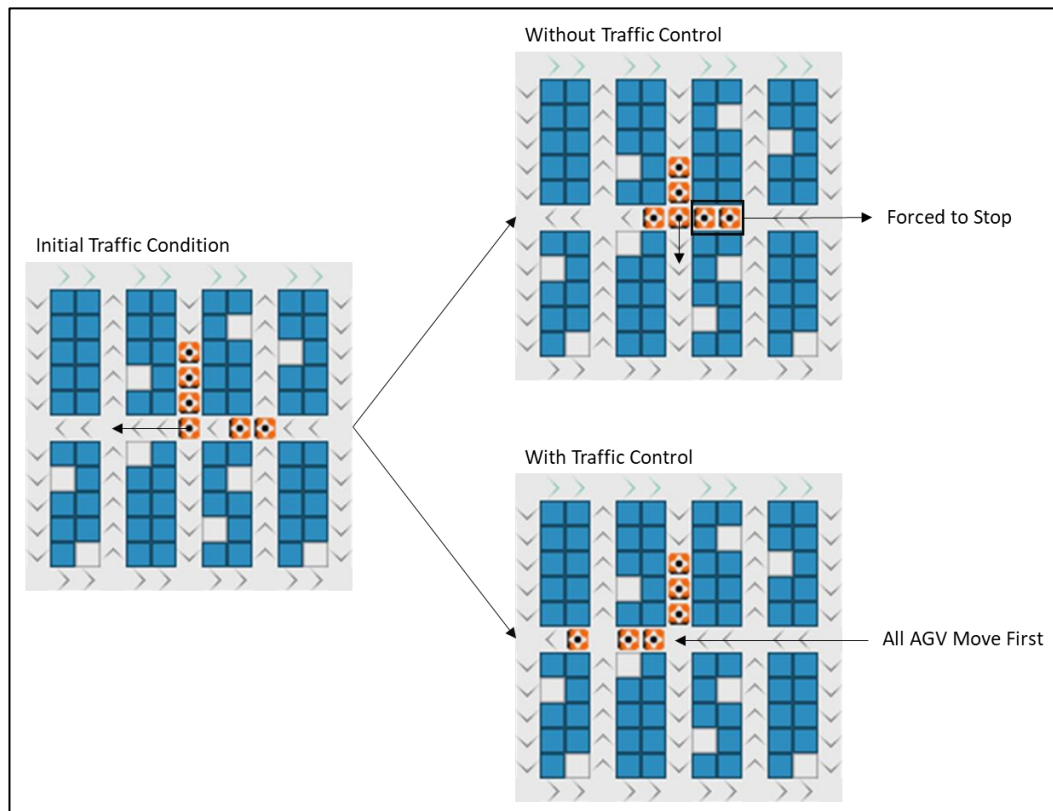


Figure 4.18 Traffic Control Importance

Figure 4.18 shows the importance of implementing traffic control in the warehouse. Without traffic control, AGV will move whenever the intersection is empty, causing a stop-and-go condition for AGV in the other direction. However, implementing traffic control as a policy will help to reduce the number of stop-and-go.

The first step to implement the traffic policy in this simulation is by detecting the traffic. Traffic information will be stored at each intersection. The intersection will make a radius to detect the surrounding AGV, then counting the

approaching AGV. If the approaching AGV in that intersection is more than two AGVs, it will be counted as traffic. The intersection color will turn red to indicate there is traffic there.

Radius to detect the traffic is two patches away from the intersection. It is determined from a distance between the horizontal intersection. The radius is not made in a bigger scope because it will include another intersection, which may affect the traffic policy become more complicated. It is also not made in a smaller scope because it is better to consider more AGV to make the traffic policy more effective to minimize the stop-and-go. Furthermore, before AGVs passing the intersection, it will trigger the traffic control policy, so that it will not move in FCFS manners. Figure 4.19 shows the traffic detection in this simulation.

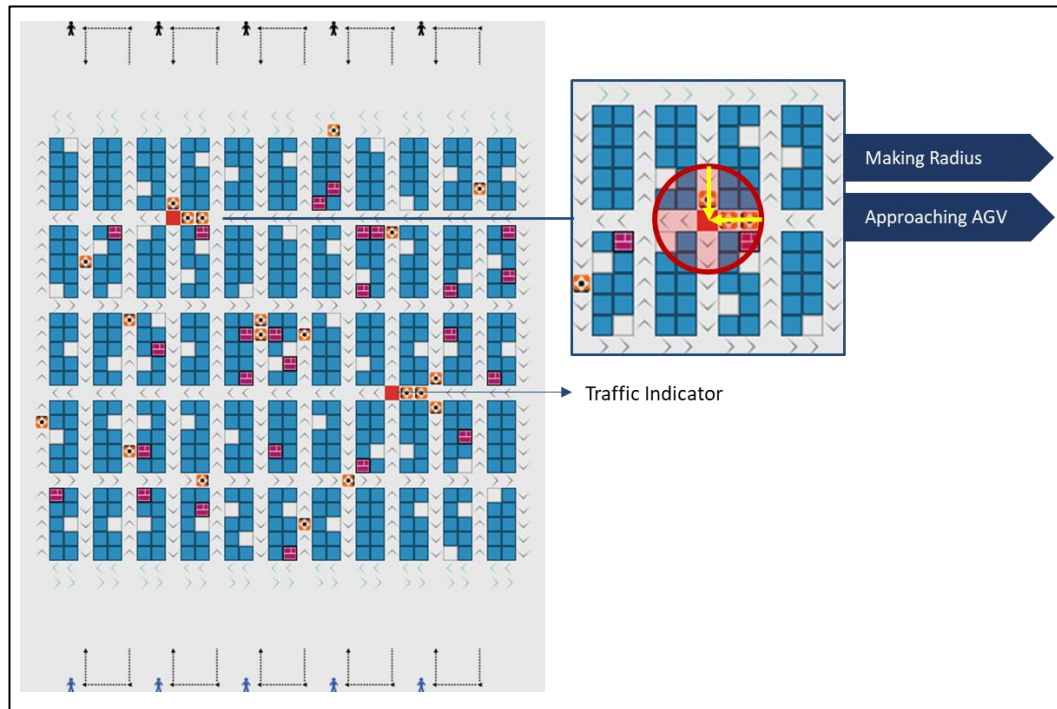


Figure 4.19 Traffic Detection in The Simulation

Traffic control that has been explained before then is developed more. Giving the horizontal aisle the priority, when two AGV in a different direction is moving towards the intersection at the same time, the horizontal one will be prioritized and move first. This rule is established because the horizontal aisle only has two patches between each intersection, while vertical ones have five patches.

Because of this condition, when traffic in the horizontal aisle reached three AGVs, it will block another aisle, causing another traffic. Therefore, AGV on the horizontal direction will be prioritized.

The movement of AGV is another thing to be considered. When AGV is approaching an intersection that indicates there is traffic, AGV will ask another AGV in that radius. When there is any AGV in the horizontal direction, the vertical one will stop to allow the horizontal AGV to move first. However, if the vertical AGV is one patch away from the intersection and the horizontal AGV is still two patches away, the vertical one can move first without blocking the horizontal AGV. But because of the horizontal priority rule, the vertical one will stop, causing unnecessary traffic for the vertical direction.

Solving that problem, AGV will also have an attribute to record whether it is previously moving or stopping. When the vertical direction AGV is previously moving, it is allowed to move first. But when the horizontal queue in intersection reaches another intersection, the horizontal will be prioritized again to move first. Therefore, it will not block the other intersection but will also not causing unnecessary traffic in the vertical direction.

4.3 Result Comparison

The baseline simulation and the improvement simulation that includes traffic policy are compared to observe how traffic policy influence AGV energy consumption. Each simulation is run for three hours, with 30 replications. The output for this simulation is the number of stop-and-go in traffic. Figure 4.20 shows the boxplot comparison of number stop-and-go.

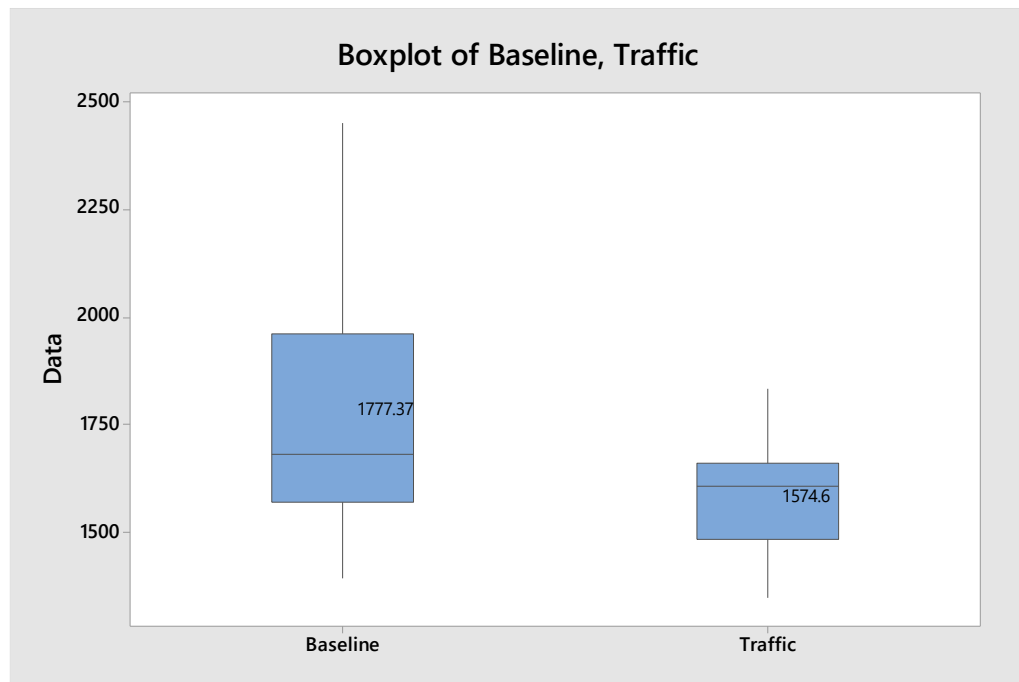


Figure 4.20 Boxplot of Number Stop-and-Go

The number of stop-and-go in traffic is reduced by implementing traffic policy. From the average of 1777.37 stop-and-go situations, it is reduced to 1574.6 on average. Thus, reducing 11.4% stop-and-go condition compared to the baseline simulation. However, it is not yet proven that by implementing this traffic policy will significantly reduce energy consumption through stop-and-go.

A statistical test is conducted to prove the baseline result, and the traffic policy result is significantly different. First, the normality test is done for each data to decide which statistical test is more suitable for the data. Figure 4.21 shows the normality test for baseline data, and Figure 4.22 shows the normality test for traffic policy data.

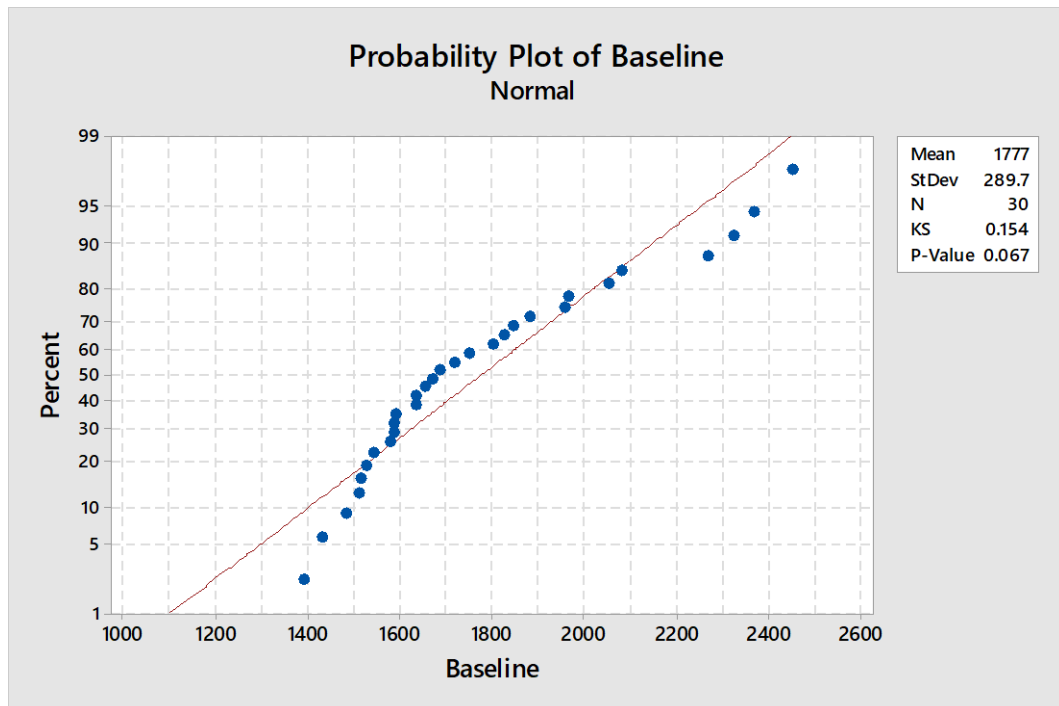


Figure 4.21 Normality Test for Baseline Simulation Data

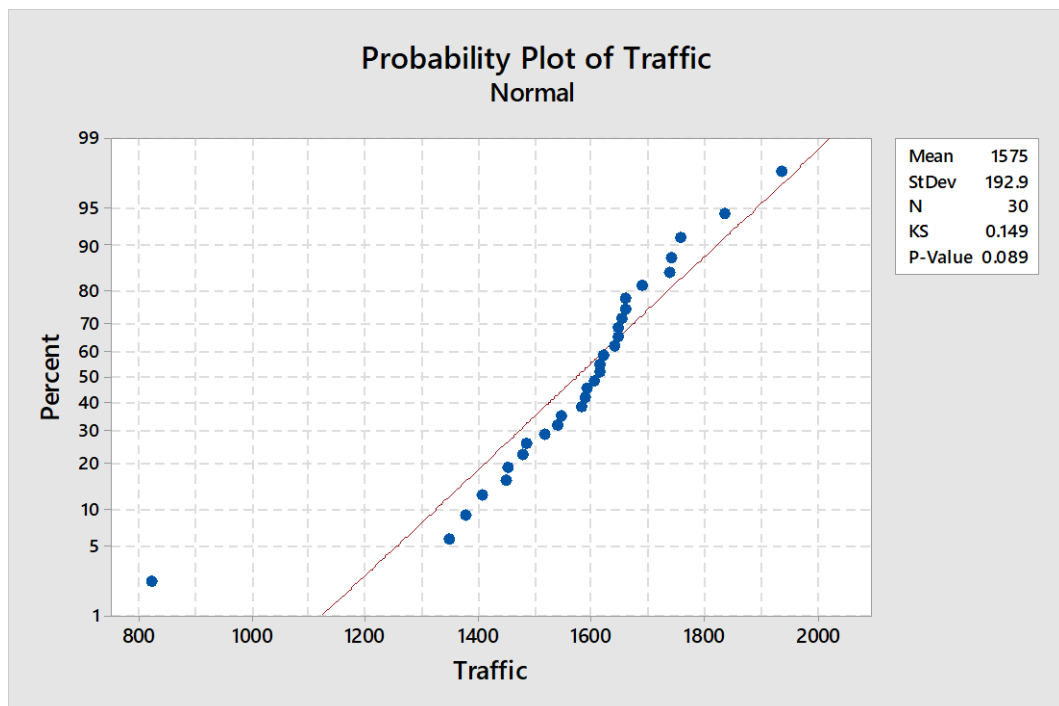


Figure 4.22 Normality Test for Traffic Policy Simulation Data

Figure 4.21 and Figure 4.22 shows that the data from baseline simulation

and traffic policy simulation are normally distributed. It shows from the P-Value, which is greater than α (0.05). Thus, unable to reject the null hypothesis that the data is normally distributed. Therefore, a Two-Sample T-Test can be conducted to test the data since it is normally distributed. Figure 4.23 shows the Two-Sample T-Test of the data.

| Test | | |
|------------------------|----|-----------------------------|
| Null hypothesis | | $H_0: \mu_1 - \mu_2 = 0$ |
| Alternative hypothesis | | $H_1: \mu_1 - \mu_2 \neq 0$ |
| T-Value | DF | P-Value |
| 3.19 | 58 | 0.002 |

Figure 4.23 Two-Sample T-Test

Two-Sample T-Test is conducted assuming that the variances of the two data are equal. Taking the degree of freedom (DF) from each data is total data minus one, DF is determined to be 58 from the total data, which is 60. From the test that is conducted, it is proved that the data is significantly different from each other. The P-Value is less than α (0.05), which means the null hypothesis is rejected. Thus, the mean of the baseline data is not the same as the mean of traffic policy data. Therefore, implementing traffic policy will significantly reduce the number of stop-and-go in the simulation.

CHAPTER 5

CONCLUSION AND FUTURE RESEARCH

5.1 Conclusion

RMFS is a conventional parts-to-picker system that e-commerce implements to fulfill the demand on time, even during the holiday season, with peak demand. It is proved to be more effective since it reduces picker's unproductive activities, such as traveling and searching. However, using an AGV as a transporter consumes too much energy through its motions. Thus, it is important to reduce energy consumption by minimizing stop-and-go in traffic.

The simulation platform is used as a tool to visualize and capture the traffic conditions that may occur. The baseline simulation is built without a traffic policy. Different from the previous study that uses path planning for robot routing and to estimate the distance for assignment, which makes the assignment is limited to a certain number, this study uses a simple rule for routing to improve the assignment process. AGV will determine its starting and ending intersection that will be used to calculate distance. Distance is calculated using three steps of Manhattan distance calculation, which is a simple calculation. Therefore, the assignment can be done with many numbers of AGVs and pods, making the assignment process more effective.

On the baseline simulation, AGV move in FCFS manners, meaning that whoever occupies the intersection first, will force the other AGV to stop and cause traffic. Traffic policy will fix this condition by implementing traffic control and coordination by detecting the traffic and store the information in the intersection. When AGV detects the traffic, it will make a decision not only by its information but considering the surrounding area. The horizontal direction will be given a priority because it has a short distance between each intersection compared to the vertical direction. By implementing this policy for traffic, it can significantly reduce the number stop-and-go in the traffic by 11.4% compared to the baseline simulation.

5.2 Future Research

There is a wide area of research that could be developed from this study. Referring to the simulation itself, the replenishment and charging part could be developed more to be included in the simulation. There are also many problems with replenishment and AGV depleting energy that is related to the charging process that has not been examined yet. Adding factors such as AGV speed (acceleration and deceleration), lifting time, and turning time will make the analysis deeper. Changing the assignment to the picking station based on the queue also can be done to reduce AGV traveling time.

The next level of improvement is to change the setting of the warehouse. Changing the pod batch size, picking station number, replenishment station number, and so on to search for the best configuration for a particular demand pattern will produce a useful result for this development. Determining the optimum number of AGV and changing the demand pattern can also be done in the future. Adjusting the other simulation setting may also affect the simulation and make an unpredictable result leading to a new insight for this scope of research.

REFERENCES

- [1] K. H. Leung, K. L. Choy, P. K. Y. Siu, G. T. S. Ho, H. Y. Lam, and C. K. M. Lee, "A B2C e-commerce intelligent system for re-engineering the e-order fulfilment process," *Expert Systems with Applications*, vol. 91, pp. 386-401, 2018, doi: 10.1016/j.eswa.2017.09.026.
- [2] J. Joong-Kun Cho, J. Ozment, and H. Sink, "Logistics capability, logistics outsourcing and firm performance in an e-commerce market," *International Journal of Physical Distribution & Logistics Management*, vol. 38, no. 5, pp. 336-359, 2008, doi: 10.1108/09600030810882825.
- [3] F. Dallari, G. Marchet, and M. Melacini, "Design of order picking system," *The International Journal of Advanced Manufacturing Technology*, vol. 42, no. 1-2, pp. 1-12, 2008, doi: 10.1007/s00170-008-1571-9.
- [4] R. de Koster, T. Le-Duc, and K. J. Roodbergen, "Design and control of warehouse order picking: A literature review," *European Journal of Operational Research*, vol. 182, no. 2, pp. 481-501, 2007, doi: 10.1016/j.ejor.2006.07.009.
- [5] R. De Koster and K. Azadeh, "Robotized warehouse systems - Developments and Research Opportunities," 2017.
- [6] J.-t. Li, "Design Optimization of Amazon Robotics," *Automation, Control and Intelligent Systems*, vol. 4, no. 2, 2016, doi: 10.11648/j.acis.20160402.17.
- [7] N. Klierer, J. F. Ehmke, and R. Borndörfer, *Operations Research Proceedings 2017: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Freie Universität Berlin, Germany, September 6-8, 2017*. Springer International Publishing, 2018.
- [8] P. Yang, K. Yang, M. Qi, L. Miao, and B. Ye, "Designing the optimal multi-deep AS/RS storage rack under full turnover-based storage policy based on non-approximate speed model of S/R machine," *Transportation Research Part E: Logistics and Transportation Review*, vol. 104, pp. 113-130, 2017, doi: 10.1016/j.tre.2017.05.010.
- [9] K. J. Roodbergen and I. F. A. Vis, "A survey of literature on automated storage and retrieval systems," *European Journal of Operational Research*, vol. 194, no. 2, pp. 343-362, 2009, doi: 10.1016/j.ejor.2008.01.038.
- [10] B. Einstein, "Meet the Drone That Already Delivers Your Packages," vol. 2020, ed: Bolt LTD, 2016.
- [11] M. Wulfraat. "Leadership in Global Supply Chain and Logistics Consulting." MWPVL International Inc. https://mwpvl.com/html/kiva_systems.html (accessed 6/4/2020, 2020).
- [12] K. VanGelder, *Fundamentals of Automotive Technology Principles and Practice*, D. Kaplan, ed., 2 ed. Vancouver, Washington: Jones & Barlett Learning, LLC, an Ascend Learning Company, p. 1047. [Online]. Available: <https://books.google.com.tw/books?id=gnE1DgAAQBAJ&pg=PA1047&lpg=PA1047&dq=energy+accelerate+decelerate+vs+move&source=bl&ots=q0zYaJlhMw&sig=ACfU3U3ad1NJkg2wcakmPYysVB0BB2EfJw&hl=en&sa=X&ved=2ahUKEwjU7Mrn1ePpAhVoyIsBHQ->

HCy8Q6AEwAHoECACQAQ#v=onpage&q=energy%20accelerate%20decelerate%20vs%20move&f=false. Accessed on: 6/3/2020.

- [13] M. Merschformann, T. Lamballais, and R. De Koster, "Decision Rules for Robotic Mobile Fulfillment Systems," *Operations Research Perspectives*, vol. 6, 01/20 2018, doi: 10.1016/j.orp.2019.100128.
- [14] M. Merschformann, L. Xie, and H. Li, "RAWSim-O: A Simulation Framework for Robotic Mobile Fulfillment Systems," *Logistics Research*, vol. 11, 08/28 2018, doi: 10.23773/2018_8.
- [15] L. Xie, H. Li, and N. Thieme, "From Simulation to Real-World Robotic Mobile Fulfillment Systems," *Logistics Research*, vol. 12, 09/04 2019, doi: 10.23773/2019_9.
- [16] B. Zou, Y. Gong, X. Xu, and Z. Yuan, "Assignment rules in robotic mobile fulfillment systems for online retailers," *International Journal of Production Research*, vol. 55, no. 20, pp. 6175-6192, 2017, doi: 10.1080/00207543.2017.1331050.
- [17] J. Zhang, F. Yang, and X. Weng, "A Building-Block-Based Genetic Algorithm for Solving the Robots Allocation Problem in a Robotic Mobile Fulfillment System," *Mathematical Problems in Engineering*, vol. 2019, pp. 1-15, 2019, doi: 10.1155/2019/6153848.
- [18] T. Lamballais Tessensohn, D. Roy, and R. B. M. De Koster, "Inventory allocation in robotic mobile fulfillment systems," *IIE Transactions*, vol. 52, no. 1, pp. 1-17, 2020/01/02 2020, doi: 10.1080/24725854.2018.1560517.
- [19] M. Merschformann, "Active repositioning of storage units in Robotic Mobile Fulfillment Systems," 01/12 2018.
- [20] D. Roy, S. Nigam, R. de Koster, I. Adan, and J. Resing, "Robot-storage zone assignment strategies in mobile fulfillment systems," *Transportation Research Part E: Logistics and Transportation Review*, vol. 122, pp. 119-142, 2019, doi: 10.1016/j.tre.2018.11.005.
- [21] N. Boysen, D. Briskorn, and S. Emde, "Parts-to-picker based order processing in a rack-moving mobile robots environment," *European Journal of Operational Research*, vol. 262, no. 2, pp. 550-562, 2017, doi: 10.1016/j.ejor.2017.03.053.
- [22] M. Merschformann, L. Xie, and D. Erdmann, "Multi-Agent Path Finding with Kinematic Constraints for Robotic Mobile Fulfillment Systems," 06/28 2017.
- [23] L. Cohen *et al.*, "Rapid Randomized Restarts for Multi-Agent Path Finding: Preliminary Results," presented at the Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, Stockholm, Sweden, 2018.
- [24] E. Bonabeau, "Agent-Based Modeling: Methods And Techniques for Simulating Human Systems," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99 Suppl 3, pp. 7280-7, 06/01 2002, doi: 10.1073/pnas.082080899.
- [25] C. M. Macal and M. J. North, "Tutorial on Agent-Based Modeling and Simulation PART 2: How to Model with Agents," in *Proceedings of the 2006 Winter Simulation Conference*, 3-6 Dec. 2006 2006, pp. 73-83, doi: 10.1109/WSC.2006.323040.

- [26] S. J. Taylor, Ed. *Agent-Based Modeling And Simulation* (The OR Essentials series. Palgrave Macmillan, London, 2014.
- [27] M. Salgado and N. Gilbert, "Agent Based Modelling," 2013, pp. 247-265.
- [28] S. M. N. Arifin, G. R. Madey, and F. H. Collins, *Spatial Agent-Based Simulation Modeling in Public Health: Design, Implementation, and Applications for Malaria Epidemiology*. 2016, pp. 1-288.
- [29] C. Keramydas, D. Bechtsis, and D. Aidonis, "Agent-Based Simulation for Modeling Supply Chains: A Comparative Case Study," *International Journal of Operational Research*, vol. 2, pp. 36-39, 10/01 2016.
- [30] B. Karima, S. Ellagoune, H. Seridi, and H. Akdag, "Agent-based modeling for traffic simulation," 2012, pp. 51-56, 01/01.
- [31] M. Papageorgiou, C. Kiakaki, V. Dinopoulou, A. Kotsialos, and W. Yibing, "Review of road traffic control strategies," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2043-2067, 2003, doi: 10.1109/jproc.2003.819610.
- [32] C. J. Leonard, Kates, "Traffic Signal Systems," Ontario, Canada, 1966.
- [33] J. Liu *et al.*, "Secure intelligent traffic light control using fog computing," *Future Generation Computer Systems*, vol. 78, pp. 817-824, 2018/01/01/ 2018, doi: <https://doi.org/10.1016/j.future.2017.02.017>.
- [34] B.-L. Ye, W. Wu, and W. Mao, "A Method for Signal Coordination in Large-Scale Urban Road Networks," *Mathematical Problems in Engineering*, vol. 2015, p. 720523, 2015/09/16 2015, doi: 10.1155/2015/720523.
- [35] C.-Y. Huang, C.-Y. Lai, and K.-T. Cheng, "CHAPTER 4 - Fundamentals of algorithms," in *Electronic Design Automation*, L.-T. Wang, Y.-W. Chang, and K.-T. Cheng Eds. Boston: Morgan Kaufmann, 2009, pp. 173-234.