MULTI-ROBOT PATH-PLANNING ALGORITHM FOR AUTOMATED PARCEL SORTING CENTRE WITH RING ROAD LAYOUT

BY

JONG QIAN BIAO

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JAN 2023

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Multi-Robot Path-Planning Algorithms for Automated Parcel Sorting Centre with Ring Road Layout

Academic Session: Jan 2023

I ___________________________Jong Qian Biao________________________

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ (Author's signature)

_________________________ (Supervisor's signature)

Address:
Lot 3417, Taman Delight 1B,
Jalan Airport, 98000 Miri,
Sarawak.

Liew Soung Yue
____
Supervisor's name

Date: 16/4/2023

Date: 28/4/2023

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 16/4/2023

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that ***Jong Qian Biao*** (ID No: ***1901846***) has completed this final year project entitled "Multi-Robot Path-Planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout" under the supervision of Prof Ts Dr Liew Soung Yue from the Department of Computer Science, Faculty of Information and Communication Technology, and Ts Wong Chee Siang from the Department of Computer Science, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____

(*Jong Qian Biao*)

DECLARATION OF ORIGINALITY

I declare that this report entitled "**MULTI-ROBOT PATH-PLANNING ALGORITHM FOR AUTOMATED PARCEL SORTING CENTRE WITH RING ROAD LAYOUT**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : 

Name : Jong Qian Biao

Date : 16/4/2023

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr Liew Soung Yue, who gave me this brilliant opportunity to work on this project and spent much of his valuable time supporting and guiding me.

I must also thank my parents, who have provided love, support, and encouragement throughout my university life. I want to thank my friends as well for providing suggestions and encouragement that make me feel possible to finish the whole project on time.

ABSTRACT

Due to the rapid growth of e-commerce, warehouses and sorting centres need to handle a huge number of parcels every day. Some companies are automating their warehouses and sorting centres to cope with the increasing number of parcels. Automated warehouses use automated guided vehicles (AGV) or autonomous mobile robots (AMR) to help sort parcels. These robots work in a highly dynamic environment and are usually implemented in huge numbers. Therefore, it is important to ensure that the robots do not collide with each other and that tasks are distributed fairly and efficiently.

Most recent approaches to the navigation of robots assume that the road system for robots to travel in the indoor environment exhibits a simple grid layout. As a result, although the proposed algorithms by recent papers could solve the collisions among the robots, they have less effect in resolving the traffic jam issue caused by the robots passing through the core part of the road system. Such a traffic congestion phenomenon greatly limits the throughput when the number of robots increases. Since the automated warehouses and sorting centres are usually large and contain hundreds of robots, the recent algorithms with the simple grid layout assumption are doubted to be efficient in the real world.

Therefore, this project proposes a map layout that implements the outer ring road concept in addition to the inner grid layout. The outer ring road allows robots that wish to travel a longer distance from one end to another without the need to pass through the core road system, thus reducing the amount of traffic in high-traffic areas. This layout works because it provides traffic that travels a longer distance and an alternative path towards their destinations, minimising congestion with other robots.

Besides, this project also evaluates and implements the approaches of recent studies and discovers new approaches that could handle the movements of hundreds of robots in real-time while maximising the throughput.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AGV	Automated Guided Vehicle
AMR	Autonomous Mobile Robots
MCMC	Malaysian Communication and Multimedia Commission
SEA	Southeast Asia

Chapter 1

Introduction

Chapter 1 discusses the project background, problem statements, motivations, objectives, scopes, and contributions. First, the project background section briefly explains the key terms and provides some background information regarding the topic. Next, the problem statement section shows various problems with parcel sorting that the current world faces. Then, the motivation section deliberates on the motives that drive the research. Project objectives define the goal of the project, and the scopes are the tasks to be completed. Lastly, the contribution section explains how the project will benefit us.

1.1 Background

E-commerce is the selling and buying of products through online platforms. With the continuous advancement of e-commerce technology and its incredible benefits to businesses, many companies have started adopting e-commerce strategies. Global e-commerce sales are predicted to reach a total amount of $5.5 trillion in the year 2022 and will continue to increase steadily in the future years [1].

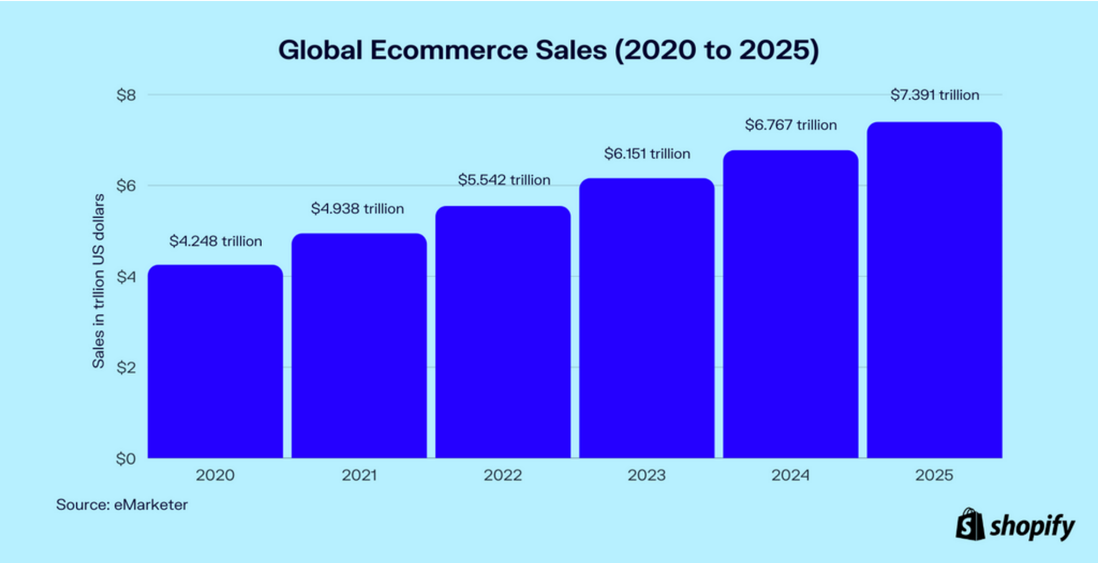


Figure 1.1.1. Global Ecommerce Sales [1]

E-commerce companies must deliver products to their customers whenever orders are received. The primary and most common method that e-commerce companies use to deliver products to their customers is through parcel delivery services offered by parcel delivery companies. The parcels are handled and processed through the parcel delivery chain before they can reach the customers. Figure 1.1.2 illustrates the whole parcel delivery process.

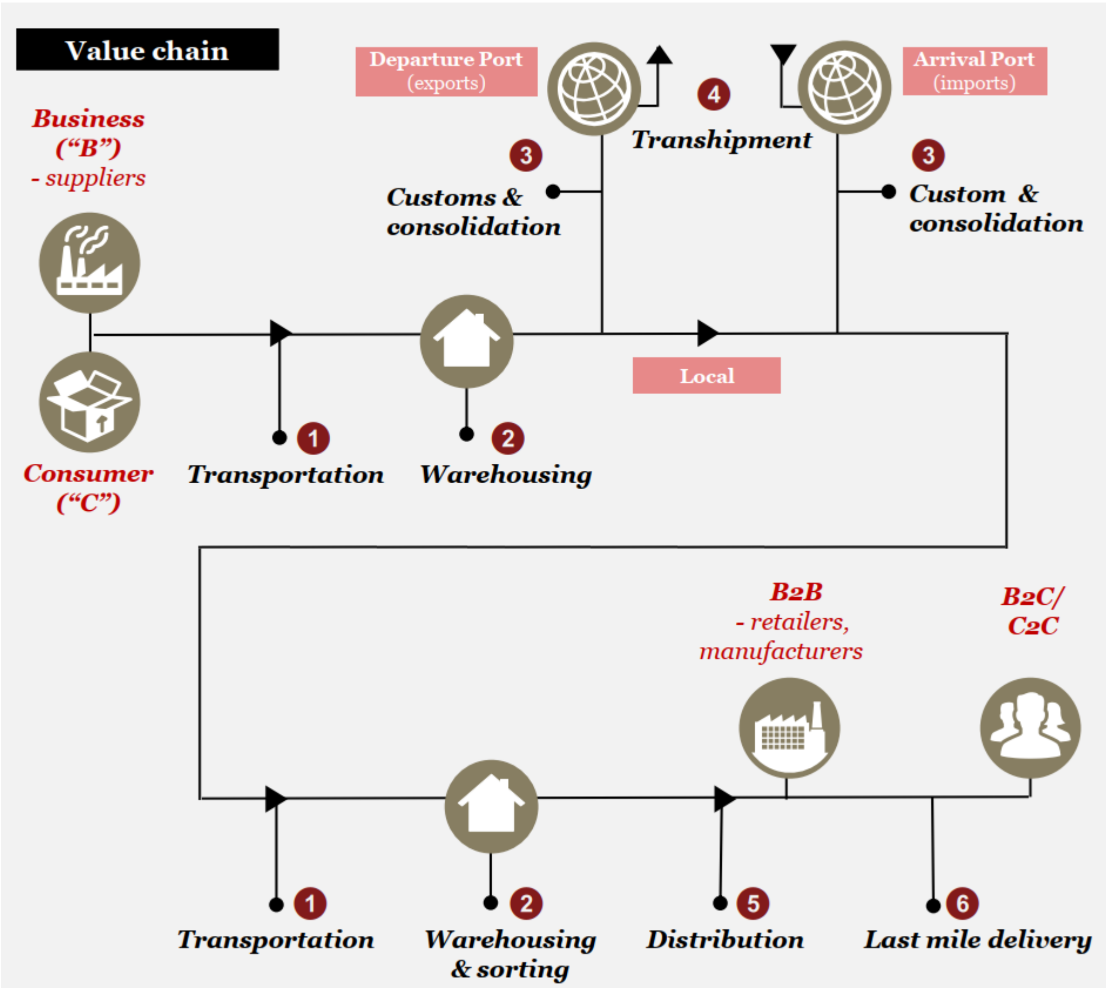


Figure 1.1.2. Parcel delivery chain [2]

Warehouses and sortation centres are components of the parcel delivery chain. The parcels are sorted according to their delivery destination using manual or automated methods. The manual sortation method needs hiring workers to sort the parcels. Conversely, the automated sortation method uses either autonomous guided vehicles (AGV) or autonomous mobile robots (AMR). In comparison, AMRs are superior to AGVs as they offer extra flexibility. AMRs can understand their environment and move dynamically inside the building in predefined zones to pick up and sort the parcels [3]. At the same time, a control framework will be used for communication among the AMRs and coordinates the navigation of each AMR. A task scheduling algorithm is a part of the controlling framework that assigns tasks such as picking up parcels and charging to AMRs. A path-planning algorithm that decides the shortest or the most efficient path for the AMRs is also required.

1.2 Problem Statement

• **Boring and exhausting job**

The parcel sorting job does not require knowledge and experience as a prerequisite, but it takes a lot of manpower if it is manually done. According to [4], workers spend up to 50% of their total picking time travelling in the warehouse. Navigating inside a large building to sort parcels for a whole day is tedious and exhausting. There is a demand for parcel sorters, but it is difficult to find employees that are willing to work as parcel sorters for a long period, resulting in a high turnover rate [5]. This situation reduces the overall profit because companies need to be spending money on training new workers constantly.

• **Inefficiencies and human errors**

Some companies hesitate to automate their warehouse when they can choose to hire more employees [6]. Indeed, human is very slow and inefficient compared to robots. First, humans need sleep and breaks, while robots only need charging and can work continuously for longer. Next, humans navigate slower inside the building and quickly feel tired. Humans also make errors, such as missorting and throwing parcels, while robots have consistent performance and thus make fewer errors. As a result, the manual sorting method lengthens delivery time and increases delivery costs.

• **Collisions between AMRs**

For warehouses and sortation centres that wish to adopt AMRs, developing an efficient framework for controlling hundreds of AMRs is also challenging. In such a congested, dynamic, and unpredictable environment, collisions between AMRs are expected to happen, and a proper framework is required to handle the collisions between AMRs. The simplest motion planning algorithms allow AMRs to adapt to collisions by slowing down or stopping completely [7]. If planned paths turn infeasible due to suddenly emerging obstacles, a new path will be generated for the AMR [7]. However, slowing down or stopping frequently would affect the overall efficiency and throughput of the AMRs.

1.3 Motivation

• **<u>Better use of human resource</u>**

Parcel sorters are no longer needed with the automation of parcel sorting that is dull and repetitive. According to [8], the world is facing a talent shortage where 75% of companies reported difficulties recruiting employees. Hiring employees is costly also. The total labour cost in warehouses could be 50% to 70% of the total budget [9]. Human workforces can be better used on more complicated tasks that robots cannot complete as humans can learn better and faster in solving problems than robots. Companies can organize workshops that enhance the skills and knowledge of some people who could be more intelligent and competitive. Automating simple tasks could force humans to leave their comfort zones, improve themselves and take up more challenging jobs to fill talent shortages.

• **<u>Increasing courier traffic</u>**

According to the Malaysian Communication and Multimedia Commission (MCMC) [10], courier traffic in Malaysia has increased significantly in recent years, as illustrated in figure 1.3.1. This phenomenon is mainly due to the sudden growth and popularization of e-commerce due to the impact of covid-19. Hiring and training more parcel sorters to handle a significant increase in parcels is inefficient, costly, and difficult. Adopting an automated method to sort parcels is way easier for warehouses and sorting centres to expand their scale, thanks to the flexibility of AMRs. AMRs could adapt to growing operating environments, such as adding pick-up, drop-off, and charging points [7]. Increasing or decreasing the number of AMRs does not need a change in the system's overall structure [7].

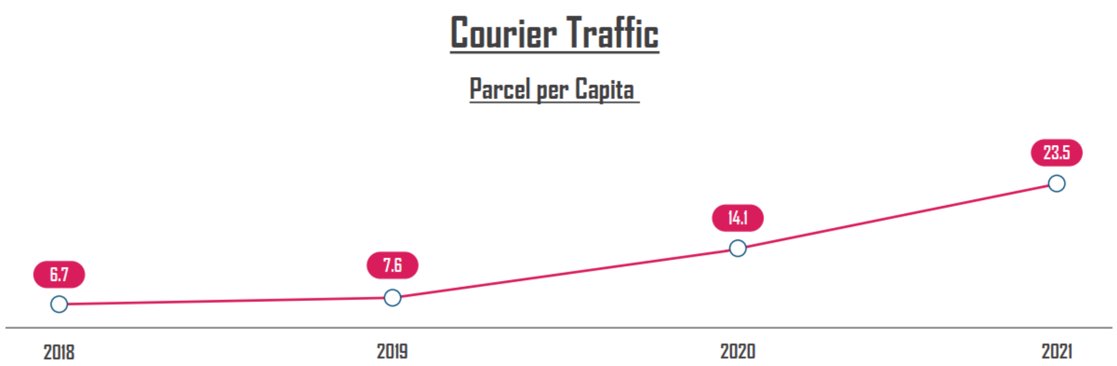


Figure 1.3.1. Courier traffic in Malaysia from 2018 to 2021 [8]

• **<u>Extremely slow parcel delivery speed in Malaysia</u>**

Regarding customer satisfaction, some of the many criteria customers would be concerned about include faster delivery and good item condition. In 2019, MCMC found that most non-business customers are satisfied with on-time delivery, scoring 4.08 out of 5 for their performance. Still, customers expect higher performance from the delivery companies, which is 4.53 [11]. Malaysia is ranked as the slowest country to deliver parcels among Southeast Asia (SEA) countries, taking up to 5.8 days on average for parcels to reach customers [12]. It implies that Malaysia lags far behind other countries in parcel delivery and needs immediate improvements. Improving the parcel sorting speed in sorting centres could perhaps be making some contributions in reducing the transit time.

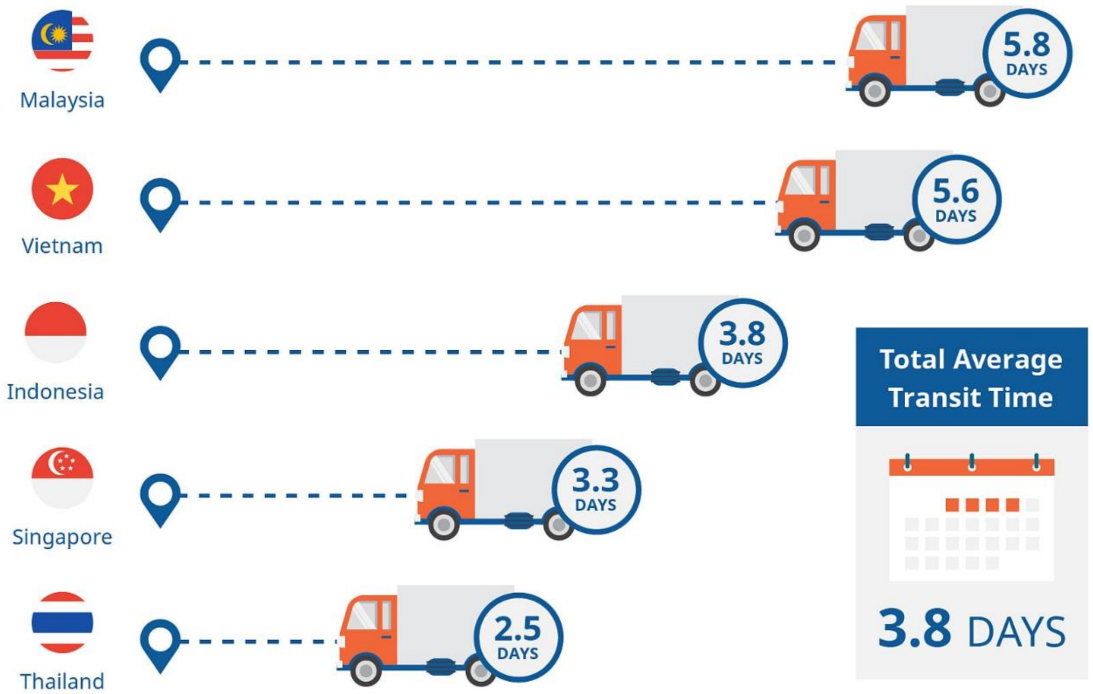


Figure 1.3.2. E-commerce delivery times of SEA countries [12]

1.4 Project Objectives

The main goal of this project is to propose a new map layout that implements the ring road concept, which could reduce the amount of traffic in the central areas of the sorting centres. Besides, a controlling framework with a path-planning algorithm and task-scheduling algorithm to collaborate the parcel sorting robots' movements are also essential. The task-scheduling algorithm should take the robot's position, charging stations, pickup points, and drop-off points as inputs and outputs the path of each robot. Thus, the objectives of this project are:

- To develop a controlling framework to avoid collisions and solve deadlocks among the robots while getting a better throughput especially when the number of robots increases.
- To develop a path planning algorithm that is computationally inexpensive to the extent that it makes real-time operations possible.
- To identify the best map layout design that implements the ring road concept to yield the maximum throughput with the algorithm.

1.5 Project Scope and Direction

The project aims to develop parcel sorting system simulations with a simulation software. The parcel sorting system simulation will require a map layout, controlling framework, task-scheduling algorithm, and a path-planning algorithm. The strategies proposed to improve the throughput will be implemented in the simulations while designing the algorithms. The algorithms implemented should be computationally inexpensive to the extent that makes real-time operations possible because it reflects whether the algorithms could run smoothly in the real world. The project will also compare the throughput of the simulations under different settings and map layouts.

1.6 Contributions

• **For Research Purposes**

This project will implement some of the strategies proposed by other papers and investigate new strategies to be considered while developing the control framework, path-planning algorithm, and task scheduling algorithm. This project will also propose a suitable map layout that reduces the need for robots to slow down or stop completely. The strategy combination should beat the throughput of the existing parcel sorting systems and act as suggestions for developing the similar systems in future studies. The project will also experiment with the settings for different strategies to suggest the best settings that could yield the optimal parcel sorting system.

• **For Application Purposes**

The proposed strategies can be implemented in parcel sorting robots in sorting centres or warehouses in Malaysia. The throughput of the automated parcel sorting systems is expected to be improved compared to the previous strategies. Consequently, the average parcel delivery speed in Malaysia can also be slightly improved. The labour shortage issue can be solved, and labour costs could also be reduced because no parcel sorter is needed. Only minimal human supervision is required to handle unexpected errors that may appear and cause failures.

1.7 Report Organization

The details of the project are shown in the remaining chapters. Chapter 2 reviews some of the recent controlling frameworks and studies their strengths and weaknesses. Next, chapter 3 shows the design and architecture of the algorithms and the controlling framework. Chapter 4 presents a preliminary study of the controlling framework. Lastly, chapter 5 concludes the entire project.

Chapter 2
Literature Review

2.1 Previous works on Automated Parcel Sorting

2.1.1 Development of Collision Free Path Planning Algorithm for Warehouse Mobile Robot [13]

2.1.1.1 Concepts

• **<u>Navigation of Robots</u>**

The paper proposed a collision-free path-finding algorithm for one AGV only. The main task of this algorithm is to prevent the AGV from colliding with the shelves in the warehouse. The warehouse map is represented in graph format, and there are four types of grids: the starting point, goal point, shelves, and paths. The shelves are considered static obstacles, and the task of the AGV is to navigate from the starting point to the goal point without colliding with the shelves. The path grids are the ones the AGV can move onto while navigating to the goal point. The goal point can be the grid containing the shelf that the AGV need to carry or the grid that the AGV needs to unload the shelf if it is carrying one.

The proposed algorithm will first check if the shelves occupy the neighbouring four grids. The algorithm will terminate if shelves occupy all four neighbouring grids because the path cannot be defined. If the shelves do not occupy more than one grid, the algorithm will choose to move to the grid with the lowest Euclidian distance. Euclidian distance is the length of a line segment between the grid and the goal point. The algorithm will guide the robot to move to the empty grid if only one exists. After the AGV moves to the grid, the algorithm will mark the grid as visited and start finding the following grid to move. The robots will not move to the grid marked as visited to avoid looping infinitely. The algorithm will keep running continuously until the Euclidian distance reaches zero, which means that the AGV has reached its destination point.

2.1.1.2 Strengths and Weaknesses

The strength of this paper is that the robot is guaranteed to avoid collisions with the shelves. However, there are many weaknesses found in this paper. The proposed algorithm does not find the shortest path, works only for one AGV, and has a restriction on map layout. The restriction on the map layout will be discussed later in chapter 2.2. Additionally, the paper only considers letting the AGV move through the bottom of the shelves if it carries a shelf.

2.1.2 Efficient Multi-AMR Control Framework for Parcel Sorting Centers [14]

2.1.2.1 Concepts

• **<u>Architecture</u>**

The paper proposed using a two-layer hierarchical architecture controlling framework, where the upper layer of the framework is the server, and the lower layer is the AMRs. The computational tasks are distributed among the server and AMRs to utilize the computational power of AMRs better. In detail, the server is responsible for collecting the environmental state information, processing it, and passing it to the AMRs. The AMRs will plan their own decision according to the local knowledge and global information obtained from the server.

• **<u>Navigation of Robots</u>**

The map is represented in the grid data format. The AMRs have adopted the A* search algorithm to find a possible path on the grid. AMRs can decide where to move, but they need the server's permission. The AMRs will reserve grids, but there is a restriction on the maximum number of grids they can reserve. The default-wait strategy is also implemented, where the robot will remain stationary at the last reserved grid. The grid reservation is based on a first come first serve basis. The paper also introduced a maximum waiting time for grid reservations. If the robots do not get the grid allocation after the maximum waiting time, the robot will put the grid into the forbidden list and plan a new path. If there is no other possible path, the robot will release the grid from the forbidden list and try to find a new path again. The robot will also clear the forbidden list when the robots finish its current task.

• **Pickup and Drop-off Points**

There will be multiple pickup points and drop-off points on the grid. The drop-off points are holes in the real world, and the AMRs can drop the parcels at any grid directly connected to the drop-off points. When the AMRs want to start a pickup task, the server will send a list of the available pickup destinations, and the AMRs can choose any destination within the list. This approach provides extra flexibility to AMRs to decide the destination according to some considerations, such as distance.

• **Charging**

The AMRs will request charging when their battery level is lower than a threshold. The server will reserve a charging station for the AMR upon request, and the charging station will be dedicated to the AMR for a certain amount of time. If there is no available charging station, the AMR will start another task and request for charging again after it has completed the task.

• **Communication**

The AMRs will communicate with the server through access points. The sorting centres are usually large. Therefore, multiple access points are set up to cover all areas. The access points are set up to communicate using different frequency channels. AMRs may communicate with the server through different access points while navigating inside the sorting centres.

2.1.2.2 Strengths and Weaknesses

The strength of this paper is that it has proposed a complete controlling framework, which handles charging, task distributing and information exchange. It has also included a detailed explanation of how the pickup points, drop-off points, charging stations and access points work. Additionally, it has also proposed a complete path-planning algorithm that could avoid collisions and deadlocks. The paper can also achieve an increased throughput when the number of AMRs grows. For the weaknesses, the paper suggested that the global and local information by the AMRs has yet to be fully utilized. Further studies are needed to explore how the AMRs can make better decisions using them.

2.1.3 Collision-Free Route Planning for Multiple AGVs in an Automated Warehouse Based on Collision Classification [15]

2.1.3.1 Concepts

• **Architecture**

The controlling framework is a two-layer hierarchical architecture controlling framework. The upper layer is the server, and the lower layer is the AGVs. The server is responsible for scheduling and routing problems, and the scheduling algorithm used by the server is an improved Dijkstra algorithm. The server will send the planned path to the AGVs, while AGVs will send their location, motion, and power information back to the server. The server communicates with AGVs through wireless communication, and the tasks are assigned to AGVs based on the priority of the tasks.

• **Navigation of Robots**

The height of the shelf bottom is higher than the height of the AGVs. The AGVs will have on-load and non-load states, which implies whether the AGVs are carrying a shelf or not. When AGVs are on load, they cannot pass through the bottom of the shelf. If it is non-load, then they can pass through the shelves. Figure 2.1.3.1.1 illustrates the shelf and the AGV.

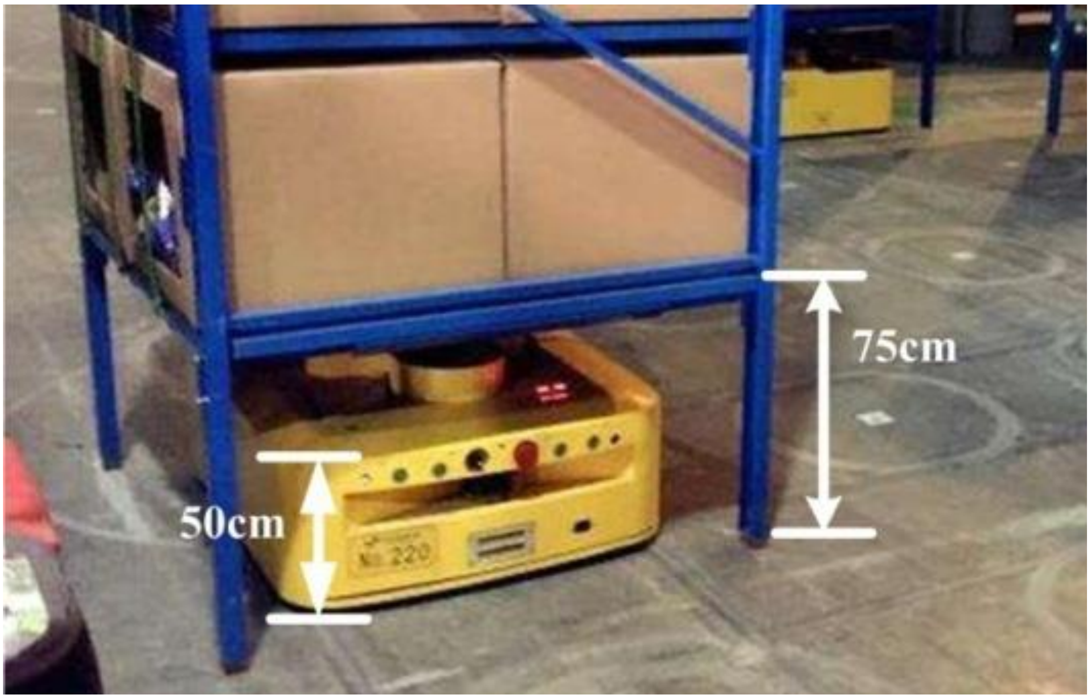


Figure 2.1.3.1. The Shelf and the AGV [14]

• **<u>Collisions</u>**

The paper stated four types of collisions of AGV, including head-on collision, cross collision, node-occupancy collision, and shelf-occupancy collision. A head-on collision happens when two AGVs travel the same path simultaneously but from a different direction. A cross-collision occurs when two AGVs compete for the same node at the intersections and will travel to different paths after passing the intersection. Node-occupancy collision happens when an AGV stops at one of the nodes of the path of another AGV. A shelf-occupancy collision occurs when an on-load AGV cannot pass through a shelf. The paper proposed 4 strategies to solve the collisions, including:

1. Selecting another shortest route.
2. Wait for a short period of time.
3. Modifying the route.
4. Re-dispatching the tasks.

After testing the solutions for solving different collisions, the paper suggested that (1) and (3) are the best solutions for head-on collisions, node-occupancy collisions, and shelf-occupancy collisions. At the same time, (2) is the best solution for cross collisions.

2.1.3.2 Strengths and Weaknesses

The strength of the paper is that it found all types of possible collisions that might happen and proposed solutions for them. Thus, collisions are entirely avoided. Non-loaded AGVs are also allowed to pass through the bottom of the shelves, cutting some distance needed to be travelled. The suggested solutions' weaknesses are that they increase the task completion time and reduce the overall throughput. The paper does not mention their throughput when the number of AGVs rises, but if the AGVs keep changing their route or spending time waiting, it will reduce the overall throughput. It happens because collisions increase when the number of AGVs increases.

2.1.4 Improving the Performance of Multi-AGV Systems with a Dynamic Unlock Algorithm [16]

2.1.4.1 Concepts

• **<u>Deadlocks</u>**

The paper stated four types of deadlocks. The first type of deadlock involves two AGVs travelling on the same path but in a different direction. The second type is crossroad crashes. The third type is an AGV stop on one of the nodes of the path of another AGV. The last type is an AGV stop at the destination node of another AGV. Figure 2.1.3.1 illustrates all four types of deadlocks.

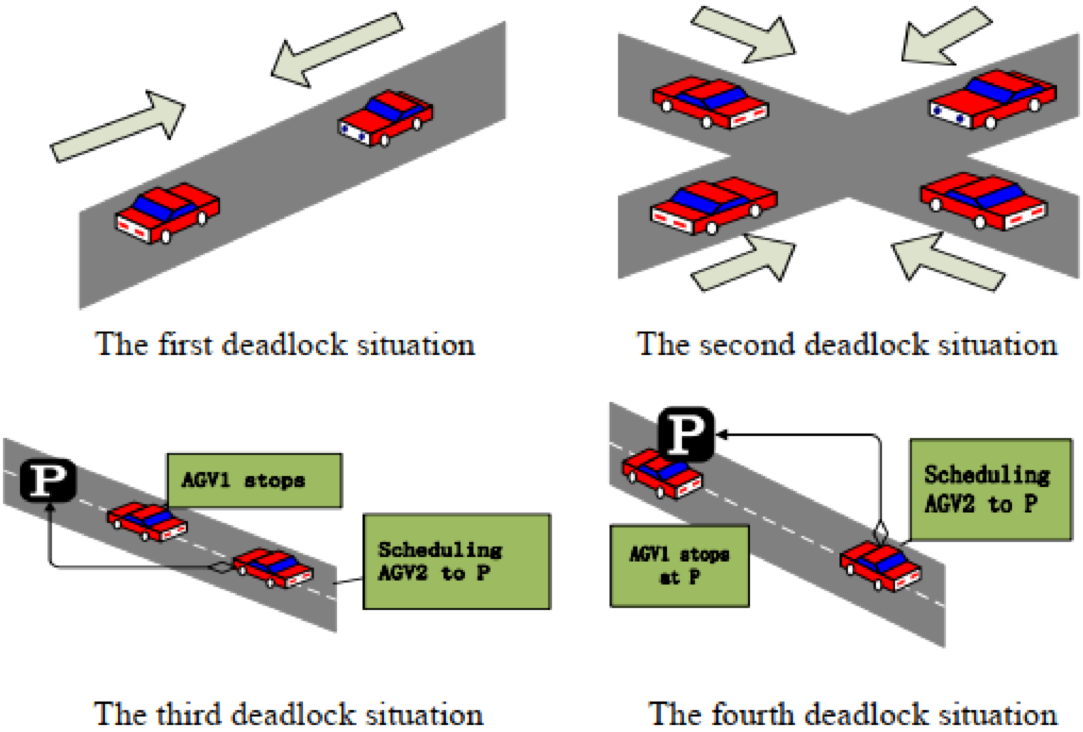


Figure 2.1.4.1. Four Types of Deadlocks [15]

The paper suggested that one AGV stops, and another AGV tries to find another path to solve the first type of deadlock. For the second type of deadlock, the paper mentioned that the existing method could only solve the second type of deadlock. Hence, the paper proposed another algorithm that could solve all four types of deadlocks, which will be discussed later. The third type of deadlock is solved by scheduling the stopping AGV to leave the position. Another solution is researching the path to bypass stopping AGV. Finally, the last type of deadlock could be solved by rescheduling the stopping AGV to another location.

The paper proposed a dynamic unlock algorithm to solve all these four types of deadlocks. The concept of the dynamic unlock algorithm is that the AGVs will reserve all nodes in the assigned path and release the node one by one right after they pass through it. AGVs that wish to pass through the nodes reserved by another AGV will have to wait until the node is released. This concept could solve the first and second types of deadlocks. After the AGVs have done their tasks, they will return to the parking space. Thus, the AGVs will never stop at the nodes for navigation. Therefore, no additional actions are needed to resolve type three and type four deadlocks.

2.1.4.2 Strengths and Weaknesses

Like the previous paper, this paper identified all types of deadlocks that could happen and proposed solutions. The strength of this paper is that deadlocks can be totally avoided. However, the solutions proposed by them are inefficient. Their solutions will cause the system's throughput to drop when the number of AGVs increases. Explanations will be provided later in chapter 2.2.

2.1.5 Comparison of Deadlock Handling Strategies for Different Warehouse Layouts with an AGVS [17]

2.1.5.1 Concepts

- **<u>Map Layout</u>**

The routes of the AGVs are planned based on the shortest distance to the destination point. The paper identified six types of map layouts in which the route can be single lane or double lane, and unidirectional lane or bidirectional lane. Figure 2.1.5.1 illustrates an example of all six different types of map layouts. The paper tested the deadlock avoidance strategy and deadlock detection & resolution strategy on these map layouts.

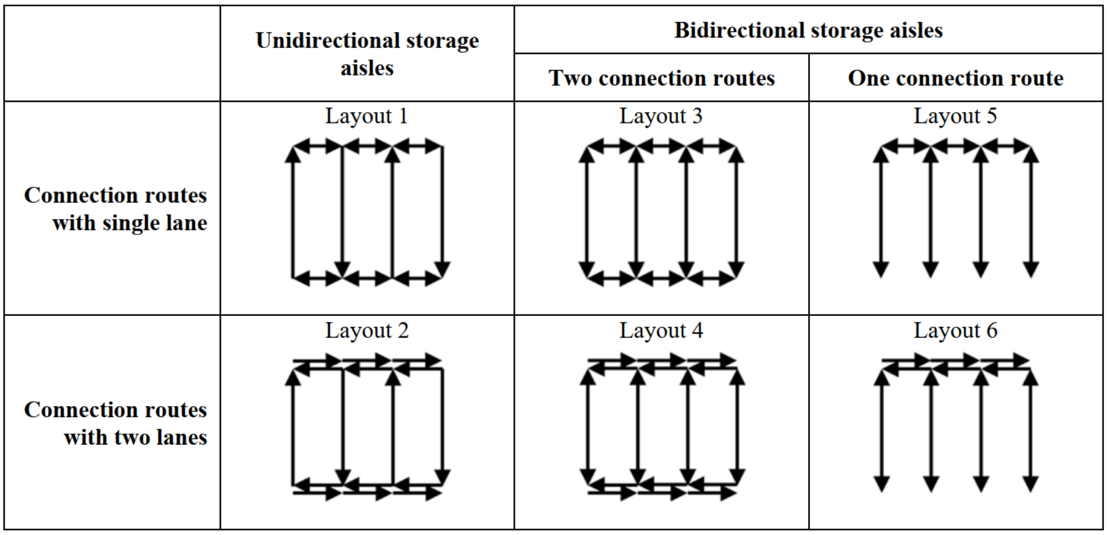


Figure 2.1.5.1. Six Types of Map Layouts [17]

• **<u>Deadlock Avoidance</u>**

The paper proposed that AGVs can avoid deadlocks by reserving a lane section for a single direction. The algorithm proposed by the paper reserve lane sections by disallowing AGVs to enter a lane section if another AGV is driving in the opposite direction. However, AGVs moving in the same direction as the AGV inside the lane section are allowed to enter. The AGVs that are rejected to enter the lane section will have to find another route to reach the destination if there are other possible routes or wait until the lane section is clear. In short, the lane section is reserved for all AGVs travelling in the same direction.

• **<u>Deadlock Detection and Resolution</u>**

Deadlocks are detected when two AGVs from different directions are competing for the same grid in a bidirectional lane section. The paper proposed solving the deadlocks by redirecting one of the AGVs. If other AGVs travel in the same lane section and the same direction as the redirected AGV, they must also be redirected. Figure 2.1.5.2 shows how to detect deadlocks and some examples of ways to resolve deadlocks. The paper mentioned that the solution illustrated is not the only solution but just serves as an example.

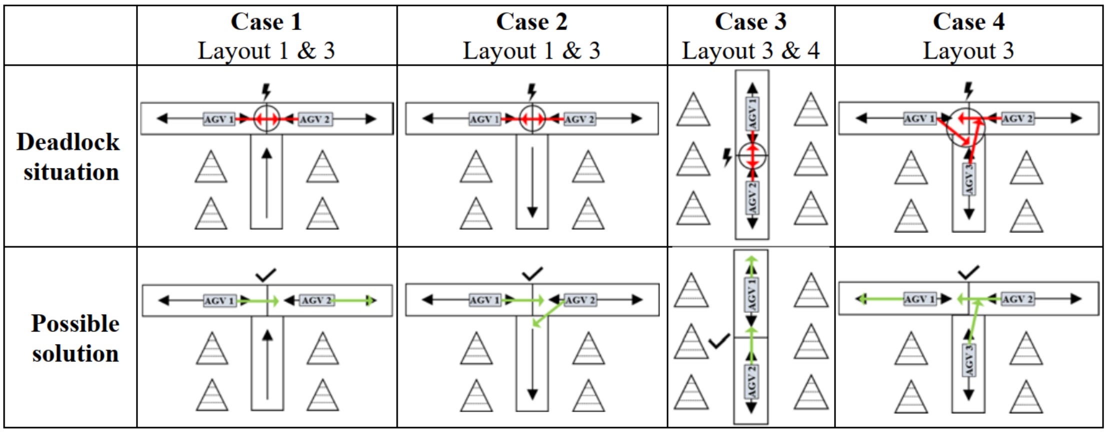


Figure 2.1.5.2. Detection and Resolution of Deadlocks [17]

2.1.5.2 Strengths and Weaknesses

The strength of the solutions provided by this paper is that it has avoided deadlock. However, the weakness of this paper is that their solution could be more efficient. The throughput is hard to be increased when the number of AGVs increases. The main reason is that reserving the whole lane section will cause AGVs to wait or be redirected. This issue will be discussed in detail in chapter 2.2.

2.2 Limitation of Previous Studies

• **<u>Marking as Visited</u>**

In [13], the proposed algorithm does not consider the situation where multiple AGVs work together. However, it is still fine because this is out of their scope. The paper uses Euclidian distance to find the path, which means that the AGV is not navigating the shortest path but only finding the correct path while navigating. The algorithm will mark the visited nodes and not move to the visited nodes again. It only removes the marks after the AGV finishes the task. The combination of these two approaches works fine on the map shown in their paper, but it would only work in a simple map, as illustrated in figure 2.2.1. The AGV in figure 2.2.1 could not get out because the previous grid is marked as visited, while shelves occupy the other three grids. Therefore, the algorithm would cause deadlock even if there is only one AGV.

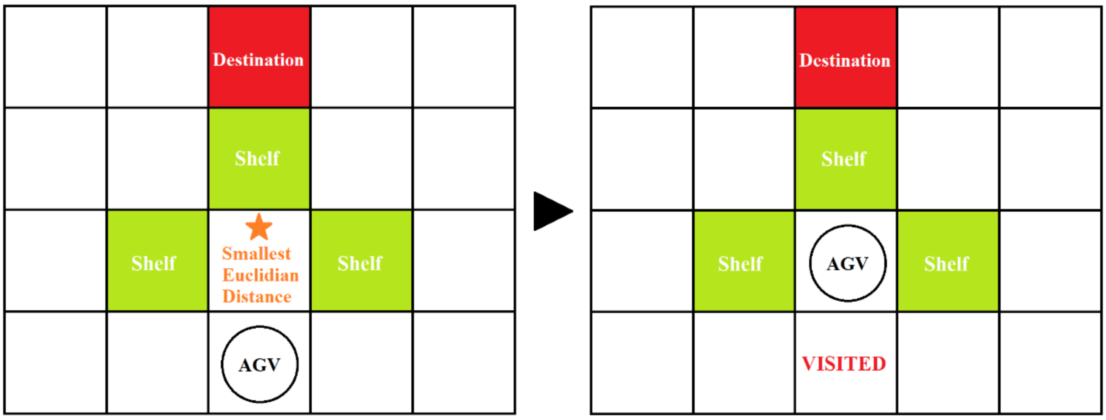


Figure 2.2.1. Deadlock Caused by the Algorithm

• **Reserving Nodes for the Whole Path**

In [16], the paper proposed reserving the nodes for the whole path and releasing the node one by one after the AGV left the node. This approach solves deadlock but greatly decreases the system's throughput, especially when the warehouse area is wide, and hundreds of robots are working inside. A robot may travel a long distance that may block the paths of many other robots. Other robots must wait until the robot releases the node, even if it is empty when other robots need it. Figure 2.2.2 illustrates this issue. The fact is also proven in the graph in figure 2.2.3, taken from their evaluation section. The time taken to finish 90 tasks on a relatively small map stop decreasing when the number of robots is more than 4.

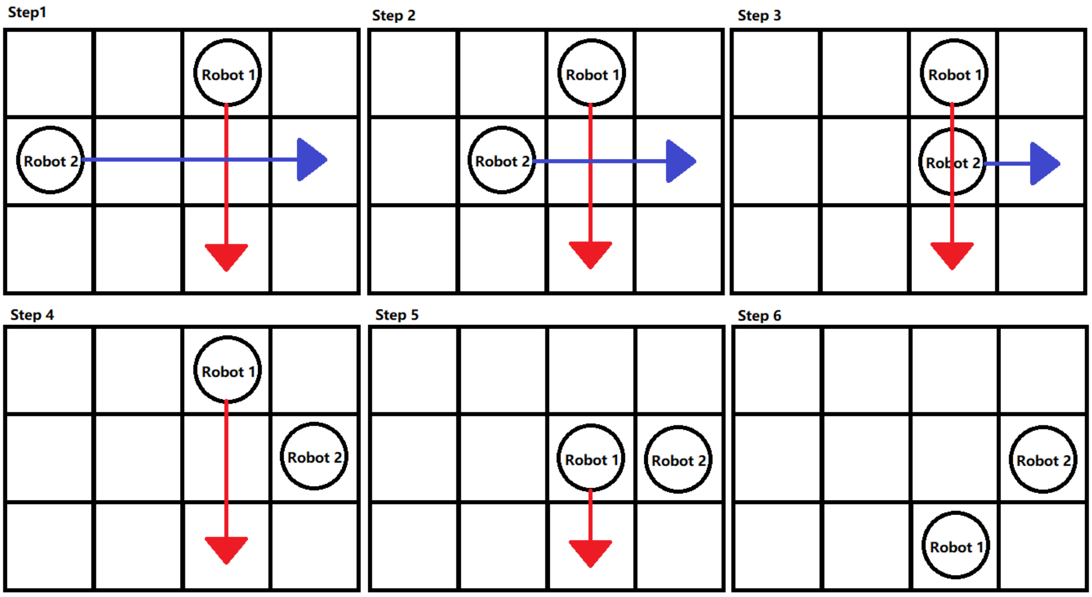


Figure 2.2.2. Inefficiencies of Reserving the Whole Path

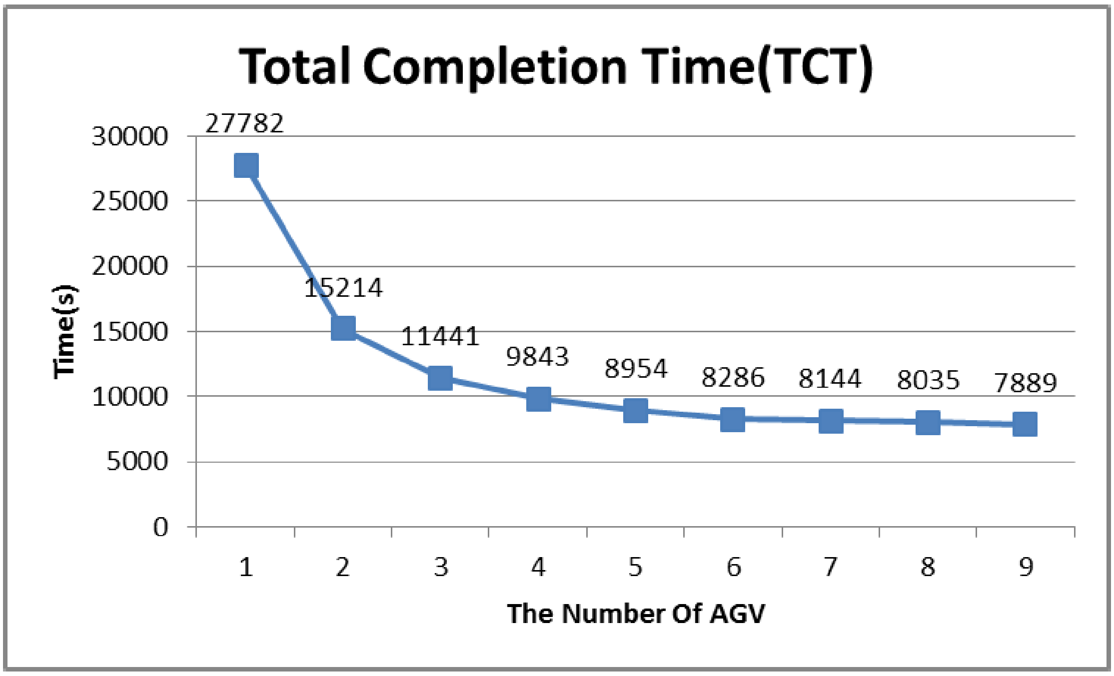
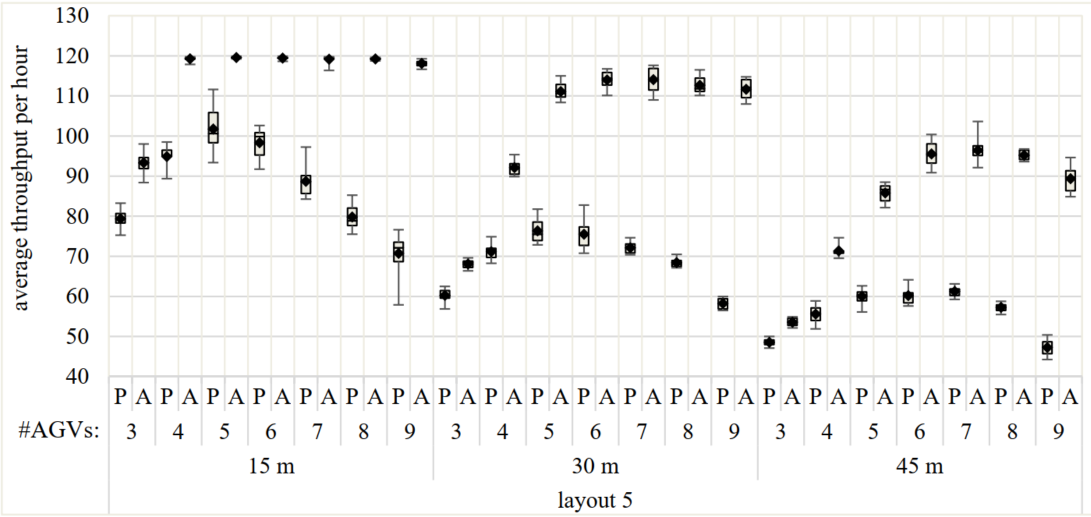


Figure 2.2.3. Time Taken to Finish 90 Tasks VS Number of AGV [16]

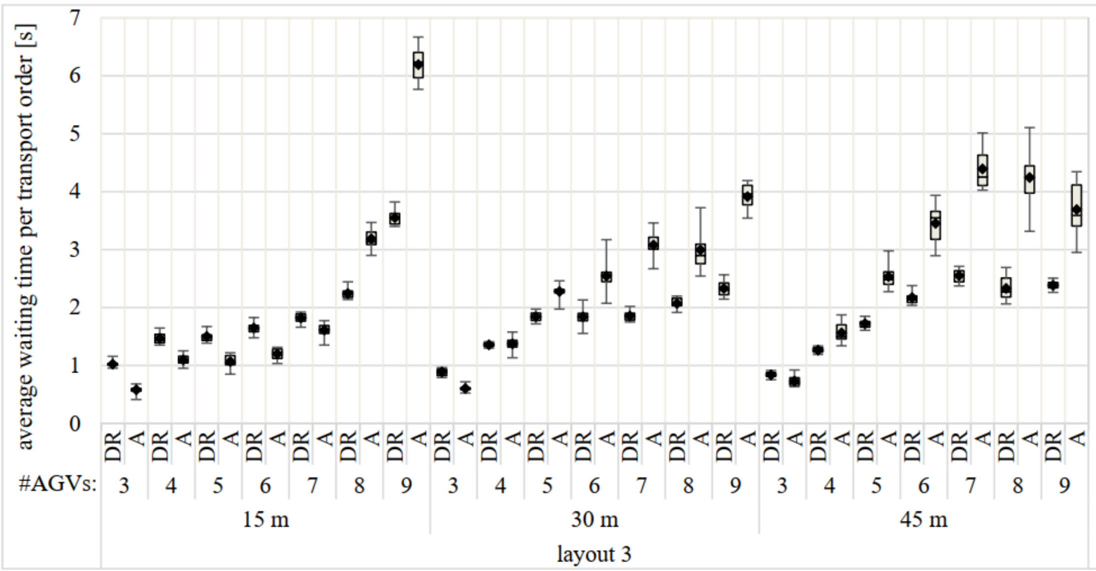
• **A Lane Section for One Direction only**

In [17], the whole concept of the paper is to restrict the AGVs in the same lane section to travel in one direction only. Other AGVs travelling in the opposite direction must wait or be redirected. The major issue of this approach is that it increases the total time to complete the task for AGVs travelling in the opposite direction. Although their proposed solution could handle deadlocks, it reduces the overall throughput and introduces extra waiting and route calculations in the system. Figure 2.2.4 and figure 2.2.5 prove the statement. The map layout used in obtaining the result in figure 2.2.4 is layout 5 in figure 2.1.5.1, while the map layout used in obtaining the result in figure 2.2.5 is layout 3. The system's throughput decreases when the number of AGVs exceeds four.



P = deadlock prevention strategy, A = deadlock avoidance strategy

Figure 2.2.4. Average Throughput Per Hour VS Number of AGVs [17]



DR = deadlock detection and resolution, A = deadlock avoidance

Figure 2.2.5. Average Waiting Time Per Task VS Number of AGVs [17]

• **<u>Use a Simple Grid Layout</u>**

All the papers reviewed used a simple grid layout. The real world's traffic system has special map designs, such as roundabouts, ring roads, and traffic lights to reduce congestion. These map designs are proven to be effective in reducing traffic jams. Since parcel-sorting robots exhibit similar characteristics to vehicles in the real world, these special map designs could also be implemented in the sorting center.

Chapter 3

System Design for Basic Grid Layout

3.1 Assumptions on the Robots

This project follows most of the important assumptions in [14]. The assumptions are as follows:

- The robots can move at a maximum speed at $3ms^{-1}$ [14].
- The robots accelerate and decelerate at $3ms^{-2}$ [14].
- The grid size of all grids is 0.6m x 0.6m [14].
- The battery capacity of the robots can last for 8 hours, and it takes 1 hours for a full charge [14].
- Robots' battery is randomly initialized between 20% to 100%.

3.2 System Design for Simple Grid Layout

3.2.1 System Architecture

The system uses both centralized and decentralized approaches for communication between the robots. The difference between centralized and decentralized approaches is that the centralized approaches will plan the paths of all robots for optimality. In contrast, the robots in a decentralized approach plan their motion as they navigate. The system also applies a two-layer architecture design. The server stays at the top layer, while the robots are at the bottom. The server shares global information, such as the position of pickup points, drop-off points, and charging stations and the location of other robots on the map.

3.2.2 Simple Grid Layout Design

A head-on collision happens when a robot collides with another robot from the opposite direction. A special design for the lanes and intersections is proposed to solve head-on collisions between the robots. The main concept of the proposed design is to restrict the directions that the robots can travel so that robots will never travel in opposing directions at the same lane or intersections. The entire map is built based on top of these concepts.

A lane section in the proposed map layout design will only allow robots from one direction to enter. The direction is fixed, which differs from the paper [17] that proposed changing direction during simulation. With the proposed strategy, we can ensure that there will be no head-on collisions inside the lane section while eliminating the need to re-route the robots when a head-on collision happens.

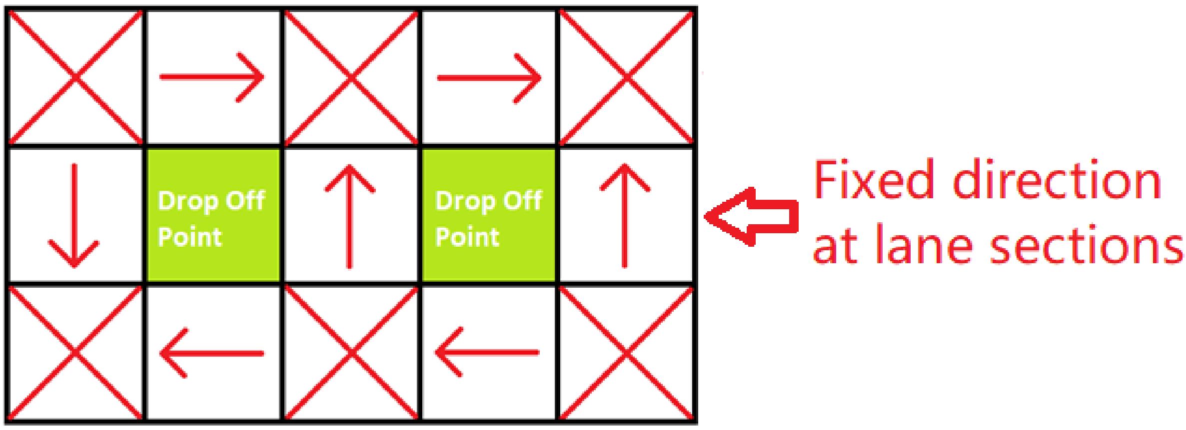


Figure 3.2.2.1. Fixed Direction in Lane Sections

Besides, the traffic entering the intersections should also be restricted so that head-on collisions will not happen at the intersections. 1x1 intersections containing only traffic from two directions will not cause head-on collisions. However, 2x2 intersections containing traffic from two or more directions will have head-on collisions. Figure 3.2.2.2 shows an example of traffic with two, three, and four directions each and how head-on collisions will happen.

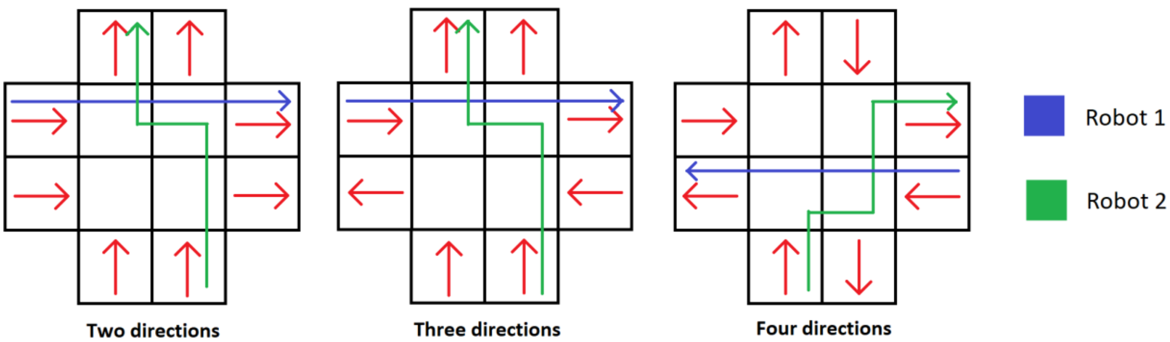


Figure 3.2.2.2. Intersections with Traffics from Multiple Directions

One of the solutions that eliminate head-on collisions in the intersections is to not have double lanes for both horizontal and vertical lanes at the intersections. Double lanes are allowed on only one of the horizontal or vertical lanes. Figure 3.2.2.3 illustrates the concept. Cross collisions will still happen, but that could be solved by requesting one of the colliding robots to wait.

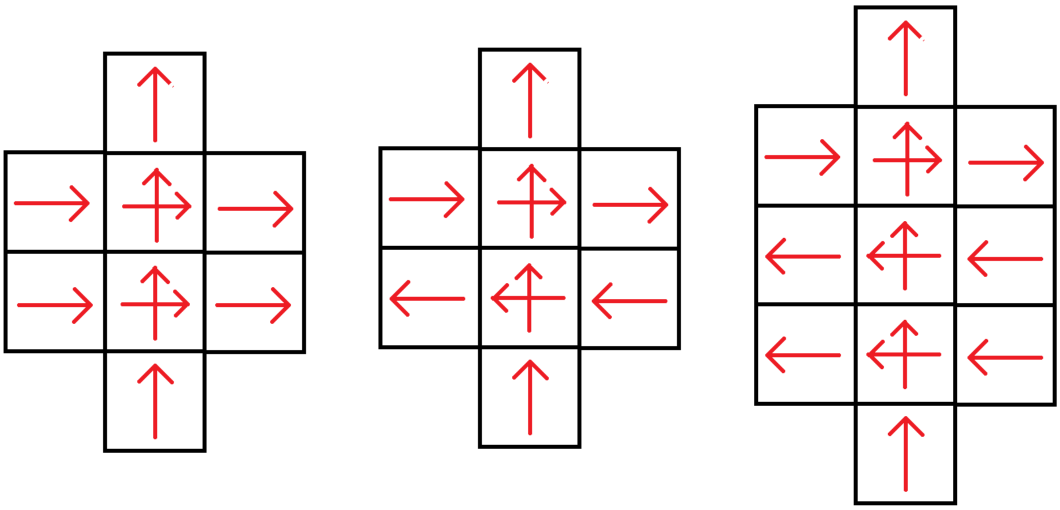


Figure 3.2.2.3. One Lane in Horizontal or Vertical Directions

Besides, head-on collisions in the intersections could also be solved by restricting the directions allowed in each intersection grid to the directions of the adjacent grids. Figure 3.2.2.4 illustrates the solution. Arrows represent the directions allowed in the grids. It eventually forms a roundabout inside the intersection, with traffic coming from four directions. This strategy is preferred because it offers double lanes, which could help share the traffic in high-traffic areas. However, both designs could be implemented together on the same map.

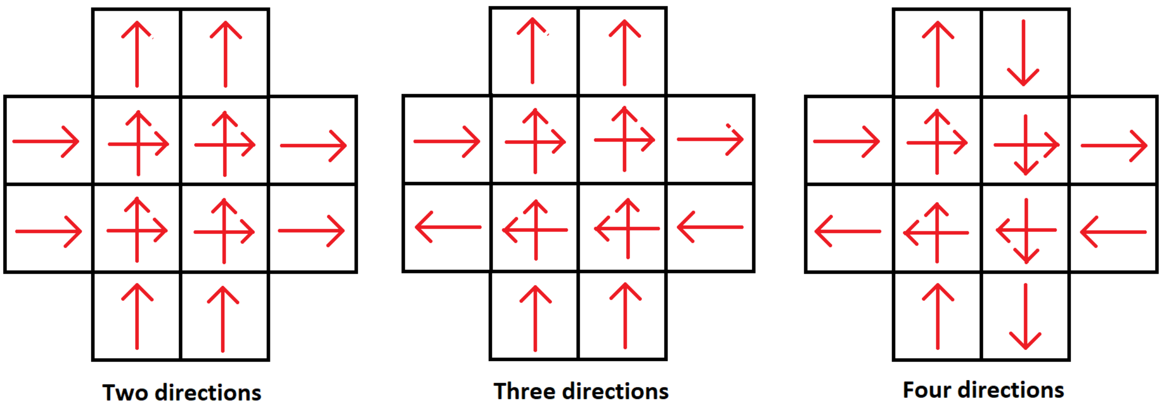


Figure 3.2.2.4. Limiting Directions at Intersection Grids

The final map layout is illustrated in figure 3.2.2.5. There is a total of 672 drop off points, 144 pick up points, and 192 charging stations. The red grids are drop off points, blue square with red triangle grids is pick up points, yellow lightning grids are charging stations, arrow grids indicate the direction allowed in the lane section, and cross grids indicates intersections. The map size is 75 x 75 unit.

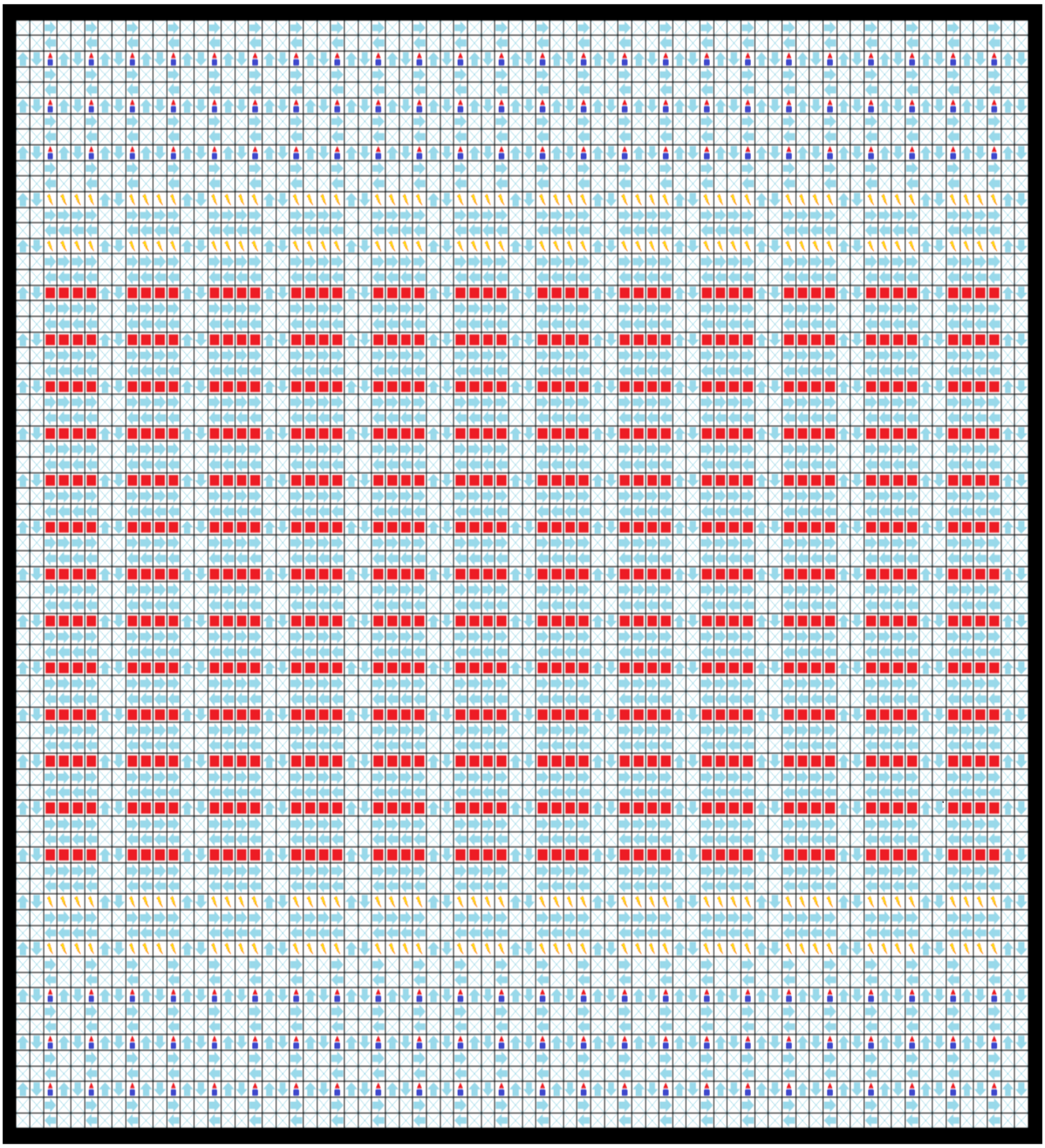


Figure 3.2.2.5. Simple Grid Layout

3.2.3 Deadlock Detection Algorithm and Resolution Strategies

The algorithm proposed here is mainly to detect circular wait deadlocks between the robots. Two types of circular deadlocks exist and require different strategies to handle. The first circular wait deadlock consists of exactly two robots, each waiting for another robot's grid. Another type of circular wait deadlock forms by four or more robots. Figure 3.2.3.1 illustrates the two types of circular wait deadlocks. Note that type one circular wait deadlock (also called head-on collisions in other papers) has already been solved using the proposed map design in chapter 3.2.2.

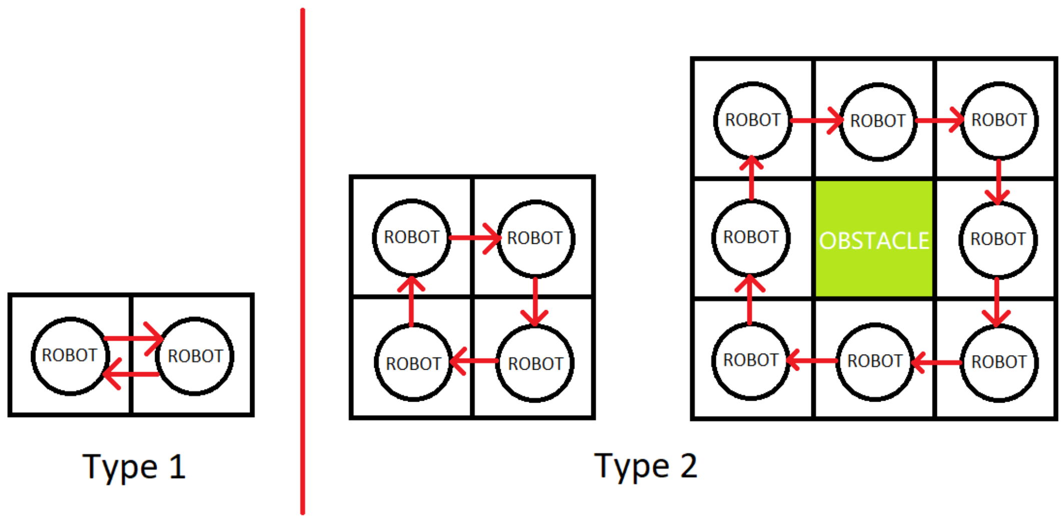


Figure 3.2.3.1. Types of Circular Wait Deadlocks

To solve type 2 circular wait deadlock, when a robot requests to move to a grid but another robot occupies the grid, the system will record the original position of the robot and the direction of the robot that the robot wants to move. Next, the system will check the grid the robot requests to check if the robot occupying the grid is requesting another grid, but the request is denied. The system further checks the requested grid by this robot. This process loops until there is no robot occupying the next requested grid. If the next requested grid eventually reaches back to the original grid, it means that there is a circular wait deadlock. The deadlock type is determined by the number of robots in the circle. Type one involves only two robots, while type two involves 4 or more robots. All robots in the circular wait deadlock should be requested to move to the next grid simultaneously to solve type two circular-wait deadlock. If all the robots in the circle travel at the same speed, they will not collide with each other.

3.2.4 Task-scheduling Algorithm

The task scheduling algorithm used follows the concepts proposed in [14]. When a robot requests a task, the robot will get a pickup point and a drop-off point from the server. The server will always have 5 tasks on hand, and the server will assign the task with pickup station closest to the robots whenever the robots request for a pickup task. Robots can pick up and drop off parcels at the adjacent four grids of the pickup and drop-off points. When a robot finishes its task, it requests a new task. The robots must charge when their battery level is less than 30%. Whenever the battery level drops below the threshold level, they will request charging after finishing the current task. If there are charging stations available, the server will assign the one closest to the robot. If there is no charging station available, the robots will continue to work until there are charging stations available.

3.2.5 Dynamic Grid Cost

The concept of dynamic cost algorithm is also proposed in [14], but the implementation of the algorithm in this project may not follow [14]. In this project, the initial cost of all grids is 3. When a robot finds a route, the cost of all grids in the route will be added by 3. The cost of the grid will be deducted by 2 after the robot has reserved and then released it. After finishing the entire route, the cost of all grids in the route will be further deducted by 1. The reason for using the dynamic cost strategy is to reduce the congestions because the robots will try to avoid navigating to the high traffic areas, thus balancing the traffic on the map.

3.2.6 Path-finding Algorithm for Simple Grid Layout

The path-finding algorithm used in this project is Dijkstra algorithm. Dijkstra algorithm is chosen to adapt the dynamic cost algorithm proposed in 3.2.5. Dijkstra algorithm will always find a route with the lowest cost in the entire route. Since changing direction requires the robots to decelerate and stop, the path-finding algorithm will also penalize routes that require frequent changing direction. The path-finding algorithm will add 10 costs to the routes when changing directions.

3.2.7 Dynamic Grid Reservation

The robots should reserve enough grids for them to decelerate only [14]. The number of grids to reserve changes dynamically according to the current speed of the robots. Since the robots can only move at a maximum speed of $3ms^{-1}$ and decelerate at $3ms^{-2}$, the distance required for the robots to stop while travelling at maximum speed is 1.5m. The dimension of the grid is 0.6m x 0.6m, so the maximum number of grids to reserve is 2.5 units and rounded up to 3 units. The robots will release the grid after they have completely moved out from the grid.

When the robots are not travelling at their maximum speed, they will try to reserve 1 grid for every 0.5 grid travelled. When they reach the maximum speed, they reserve 1 grid for every 1 grid travelled because they cannot accelerate anymore. Figure 3.2.7.1 shows the calculation. Suppose the reservation is not granted, they will start to decelerate immediately and stop at the last grid they reserve. The robots will continue to request the next grid for every 0.5 grid travelled even after the deceleration. If they eventually get the grid, they will accelerate again.

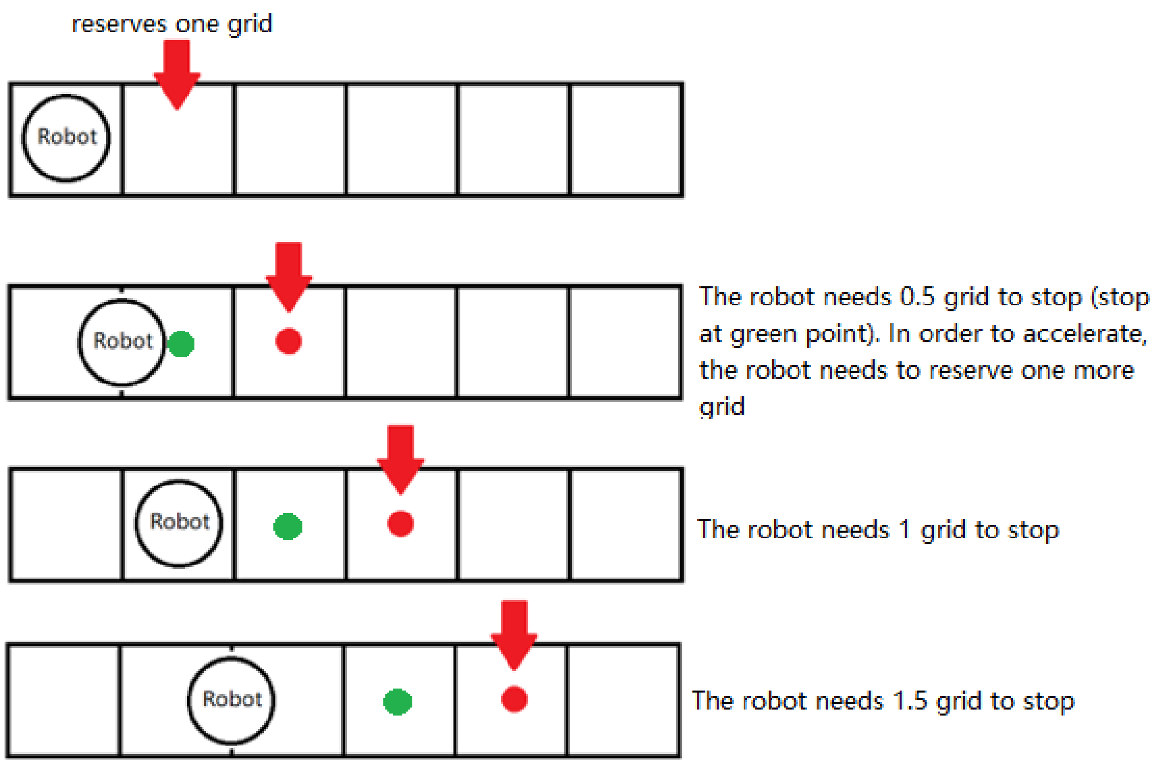


Figure 3.2.7.1. Dynamic Grid Reservation based on Speed.

3.2.8 Mutual Coordination

When robots in a line are waiting to start moving, robots at the back will always have to wait for the robots at the front to move first because of the grid reservation strategy that always reserves sufficient grids so that the robots have enough grids to decelerate. In order to move at maximum speed, the robots will need at least three grids, thus slowly forming a huge gap between the robots while moving in a line, which is inefficient. Mutual coordination strategy is proposed to solve this issue, allowing robots in the line to accelerate mutually. With this strategy, whenever a robot requests a grid that another robot reserves, the server will notify the robot to ask the grid from the robot reserving the grid. The robot reserving the grid will grant the requested grid if it is not the last grid they reserved because the last grid will be where they might need to stop. Since the robots have the same acceleration and maximum speed, robots at the back will never move faster than the robots at the front because they will not get the extra grids they need to accelerate if they want to move faster than robots at the front. This strategy is a type of decentralized approach because communication is established between the robots only. It is very useful for the ring road map layout because all robots in the ring road are moving in one direction, so the mutual coordination strategy can help the robots to accelerate together, ensuring a smoother traffic flow in the ring road.

3.3 System Design for Ring Road Layout

3.3.1 Ring Road Layout Design

Figure 4.1.1 shows the ring road map. The type of grids used are the same as in the simple grid layout. The ring road map has fewer pick-up points, drop-off points and charging points because it needs extra grids to fit the ring road, thus reducing the available space for the inner area. The inner simple grid layout is only 71 x 71 versus 75 x 75 without the ring road.

The map contains an outer ring road and an inner grid layout. The ring road serves as an alternative path for the robots that wish to travel a long distance to reduce congestion at the center of the map. There is only one lane in the ring road, and the ring road is unidirectional. The corners of the ring roads are 90 degrees curves, so the robots can move in curves and maintain their speed instead of decelerating and stopping to change directions at the corners. The radius of the curve should be set within an acceptable range to according to the self-rotation speed of the robots. In the map, the radius of the curves is 1.8 meters. The 90 degrees curves have 2 * pi * 1.8 * 0.25 = 2.83 meters quarter circumference. Since that the robots move at 3 meters per second and need to complete 90 degrees turn in 2.83 meters, the self-rotation speed of the robots should be at least 3 / 2.83 * 90 = 96 degrees per second. However, the robots that enter and exit the ring road would still have to decelerate and stop because they need to change direction.

The inner grids directly connected to the ring roads are meant for the robots to wait before they can enter the ring road, so the robots can only navigate using those grids if they want to enter the ring road. This strategy is used so that the robots waiting to enter the ring road would not block the way of other robots. The left and right sides of the map are charging stations instead of drop-off or pick-up points to reduce the traffic flow near the exits and entrances of the ring road.

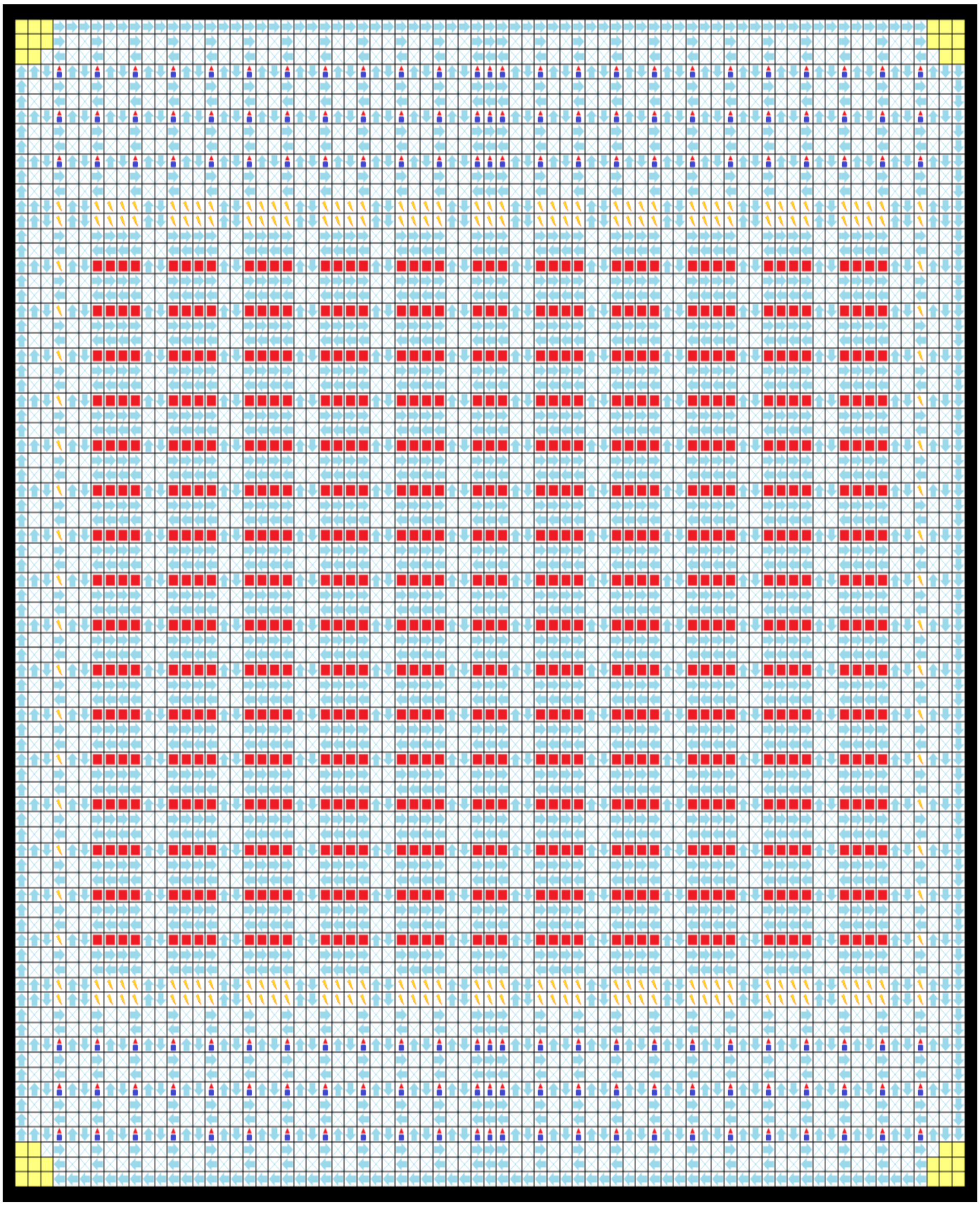


Figure 3.3.1.1. Ring Road Map Design

3.3.2 Path-finding Algorithm for Ring Road Map

When the robots do not use the ring road, they assume the map is a simple grid layout and find the route from source to destination using the Dijkstra algorithm. The grids in the ring road and the inner grids directly connected with the ring road will be the forbidden grids where the robots cannot navigate on these grids. On the other hand, the path-finding algorithm for using the ring road is sub-divided into three simpler path-finding tasks – from source to the entrance of the ring road, from the entrance of the ring road to exit of the ring road and exit of the ring road to the destination. Finding the path from the source to the entrance and from the exit to the destination uses the Dijkstra algorithm while finding the route in the ring road is a depth-first search algorithm since the ring road does not branch to other grids outside the ring road.

3.3.3 Requirements to Enter the Ring Road

The map is divided into one central area and four corner areas. Figure 3.3.3.1 shows the division. The robots can only use the ring road if the area of the start position and the destination are diagonal to each other, and both the starting point and the destination must not be in the center area. The reason to use this strategy is that there is a limited number of grids in the ring road, and thus the ring road should be prioritized for robots that travel a long distance only. It also balances the traffic that enters the ring road and navigates in the center area. Additionally, using the ring road for short-distance travel would add too much overhead compared to just travelling directly to the destination.

Besides, there is also a threshold for the number of robots that can enter the ring road so that the ring road would not be overcrowded with robots. The threshold is set to the number of grids in the ring road, excluding the grids in the curves. There is also a counter that records the number of robots using the ring road. The counter will add by one immediately when the robots find a route that uses the ring road. The counter will then be deducted by one only when the robots that use the ring road finish the entire route. If the counter reaches the threshold, no more robots will be allowed to enter the ring road, and the robots will find a route from the source directly to the destination without entering the ring road.

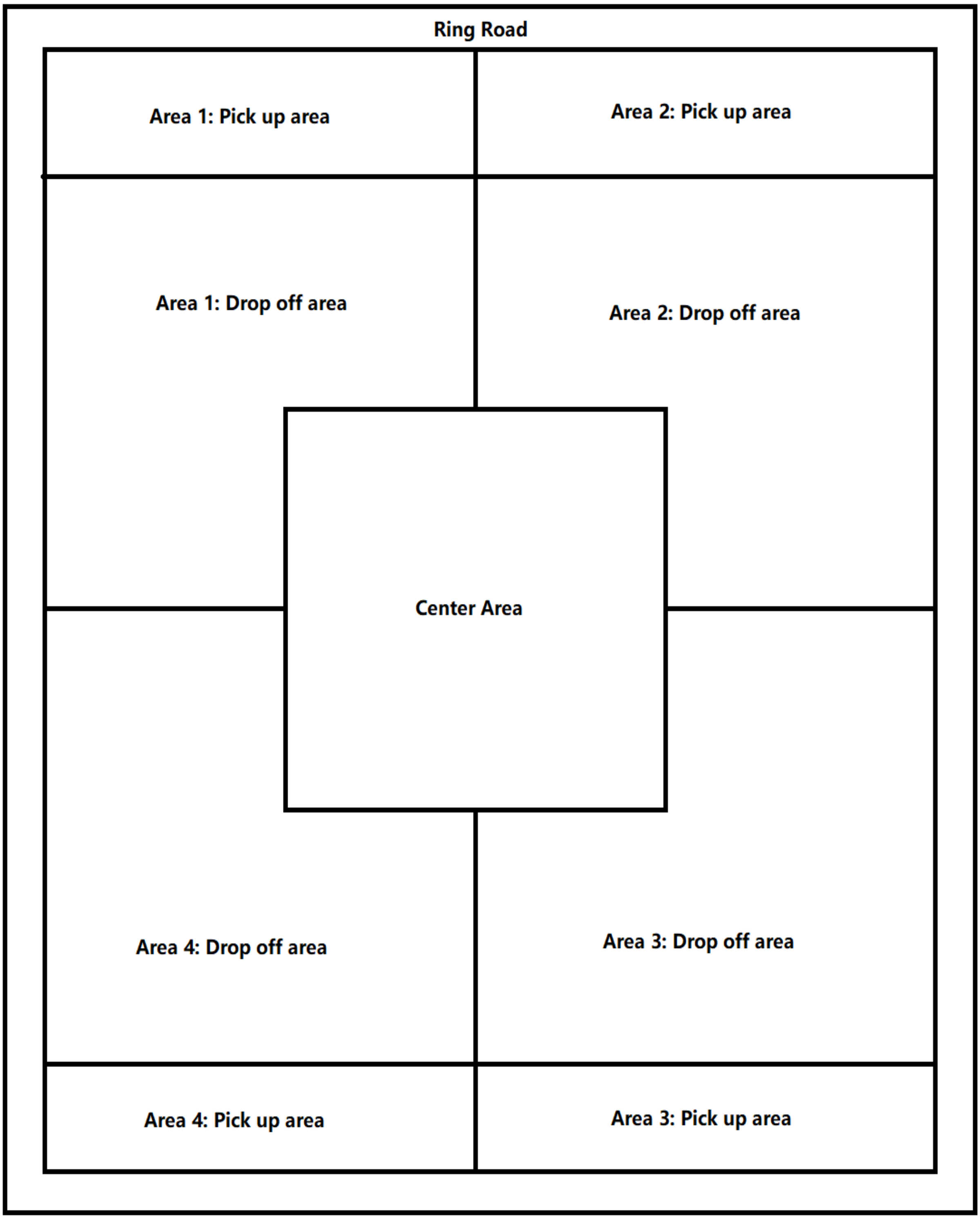


Figure 3.3.3.1. Divisions of the Map

3.3.4 Ring Road Entrances and Exits

Each pick-up or drop-off areas have its own designated exits and entrances. The exits must come first before the entrances, according to the direction of the ring road, to reduce the grids that the robots have to navigate. Figure 3.3.3.1 shows the exits and entrances. The ring road grids with a red circle are exits, while the grids with green circles are entrances. The charging stations separate the pick-up and drop-off areas. The pick-up areas have 21 exits and 21 entrances, while the drop-off areas have 12 exits and 12 entrances. The pick-up areas have a significantly larger number of exits and entrances because the pick-up areas are much smaller than the drop-off areas, thus requiring more exits and entrances to reduce congestion.

Figure 3.3.3.1. Exits and Entrances of an Area

Besides, a counter for each entrance and exit records the number of robots using them. The server will always assign the entrances or exits with the fewest number of robots using it to the robot requesting an exit or entrance. If there is more than one exit or entrance with the same number of robots using it, the server will assign the one closest to the robot's current position. The counter for both the exit and entrances will increase by one immediately when they are assigned to a robot. The counter for the assigned entrance will be deducted by one after the robot enters the ring road. However, the counter for the assigned exit will only be deducted by one after the robot completes the whole route because the robot might be stuck somewhere near the exit even after they have exited the ring road.

3.4 Workflow of Robots

Figure 3.4.1 shows the workflow of the robots. The robots will check if their battery level is greater than 30% whenever they start a new task. If the battery level is lower than 30%, they will request a charging station from the server. If the battery level exceeds 30% or there is no available charging station, the robots will request a parcel sorting task from the server. After the robots get a task, either a charging or a sorting task, the robots will check if the source and destination areas are diagonal to each other, not in the center area, and the threshold for the number of robots in the ring road is not reached. If the conditions are all true, the robot will find a route that uses the ring road and not use the ring road otherwise. While navigating, the robots will reserve one grid for every 0.5 grids they moved or one grid for every grid they moved if their speed is at the maximum. If the server does not grant the reservation, the robots will try to request the grid from the robot at the front. If it is not granted again, the robots will decelerate. If the grid is granted by the server or the robot in front, the robots will accelerate. The robots will check if they have reached the destination for every 0.5 grids they moved. If the robots have reached their destination, they will request a new task from the server. If they have yet to reach the destination, they will check if the last reserved grid is the destination or if the robots need to change direction at the last reserved grid. They will only reserve new grids if the condition is true. If it is not true, they will continue to move until they reach the last reserved grid.

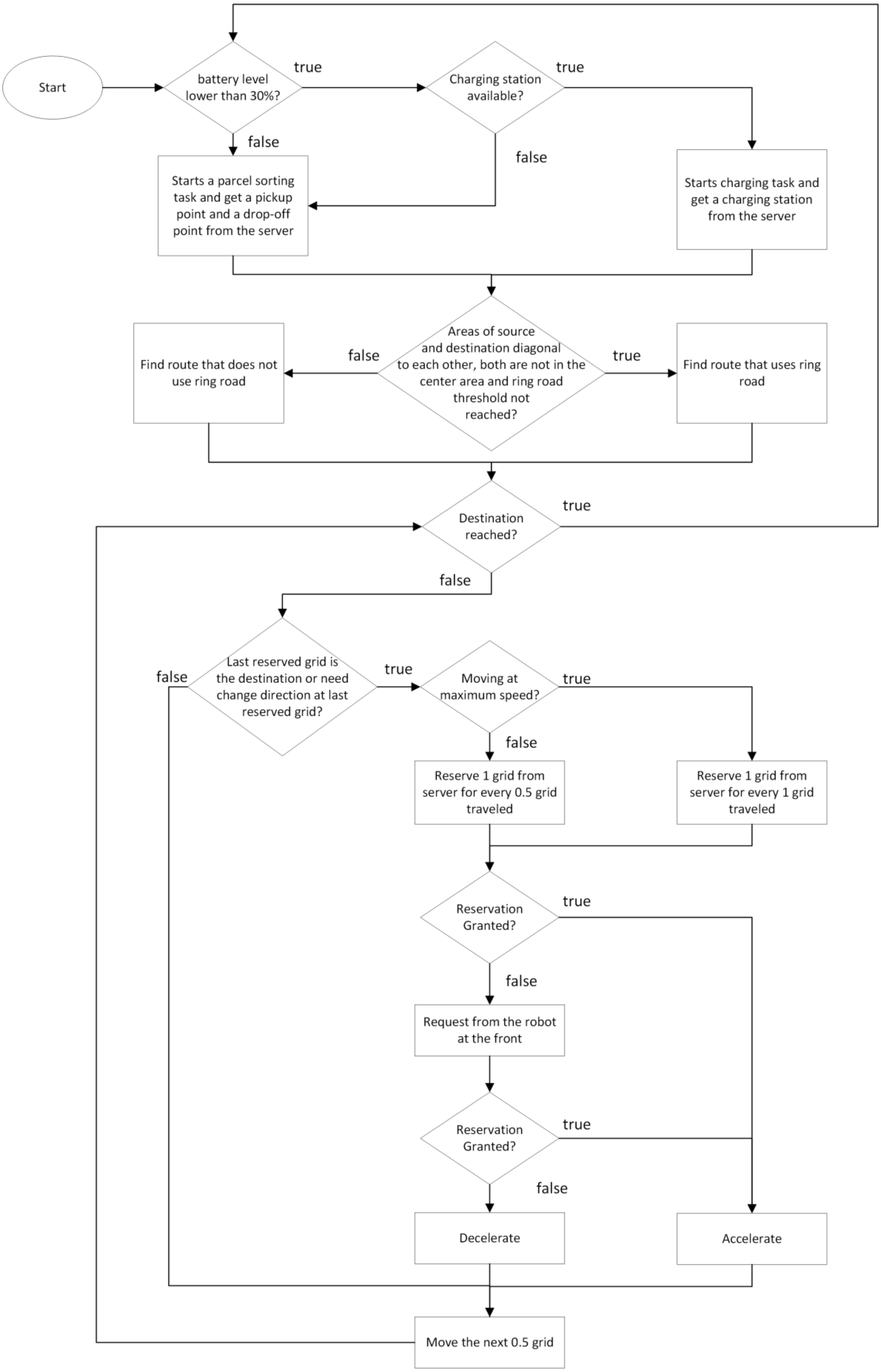


Figure 3.4.1. Workflow of Robots

Chapter 4

Experiment and Simulation

4.1 Tools to Use

The controlling framework and path-finding algorithm is written in C# language using Visual Studio 2022. The visualization is achieved using Unity. Unity is chosen because it could be used to visualize continuous events and show 2-dimensional animation easily. The hardware to develop the controlling framework is an ACER Nitro 5 series laptop with Intel i5-8300H CPU, GTX1050 GPU, 20GB RAM, 512GB SSD, and 1TB HDD.

Table 3.6.1. Specifications of laptop

Description	Specifications
Model	ACER Nitro 5 Series
Processor	Intel Core i5-8300H
Operating System	Windows 10
Graphic	NVIDIA GeForce GTX 1050
Memory	20GB DDR4 RAM
Storage	512GB SSD + 1TB HDD

4.2 Implementation Issues and Challenges

The first issue is that the overhead of the ring road is huge compared to using the simple grid layout. The robots moving in the ring road would reach their destination slower because they must move a longer distance. The robots would also spend additional time waiting for their turn to enter the ring road since the number of robots in the ring road is high and are linked together to move mutually. Besides, since the robots only move in one direction, the robots would experience deceleration or stop multiple times when other robots want to enter or exit the ring road. Therefore, the robots might travel at a lower speed inside the ring road. These issues cause the ring road map to perform significantly worse than the simple grid layout when the number of robots is less. The overhead of the ring road is not worth it when using a lesser number of robots.

The second issue with the ring road map is that it requires a significantly higher number of robots to reach its peak throughput than the simple grid layout. Putting aside cost and efficiency issues, the high number of robots also requires additional charging stations to handle the robots. The pick-up and drop-off points must be sacrificed to put in the additional charging stations. Besides, the map also needs some space to fit in the ring road. The inner grid layout is only 71 x 71 compared to 75 x 75 in the simple grid layout. The inner grid layout of the ring road map is 90% the size of the simple grid layout. This issue further limits the number of pick-up and drop-off points that can fit into the map. A reduced number of pick-up and drop-off points would cause congestion to happen easier because more robots would be going to the same point.

The third issue is that the dynamic cost algorithm would introduce additional overhead into the route of the robots. An example is shown in Figure 4.2.1. Since that robot 1 and robot 2 enter the first intersection at different timestamps, robot 1 and robot 2 can both take the first intersection without causing any issues to each other. However, the dynamic cost algorithm would disallow the second robot to take the first intersection as the additional cost has been added to the grids in the first intersection. The second robot must take the second intersection, making its route longer and bringing no benefits. A solution to this issue might be recording the cost of the grid for every timestamp. However, it is very hard to ensure its accuracy because there will be congestion between the robots that interrupts it.

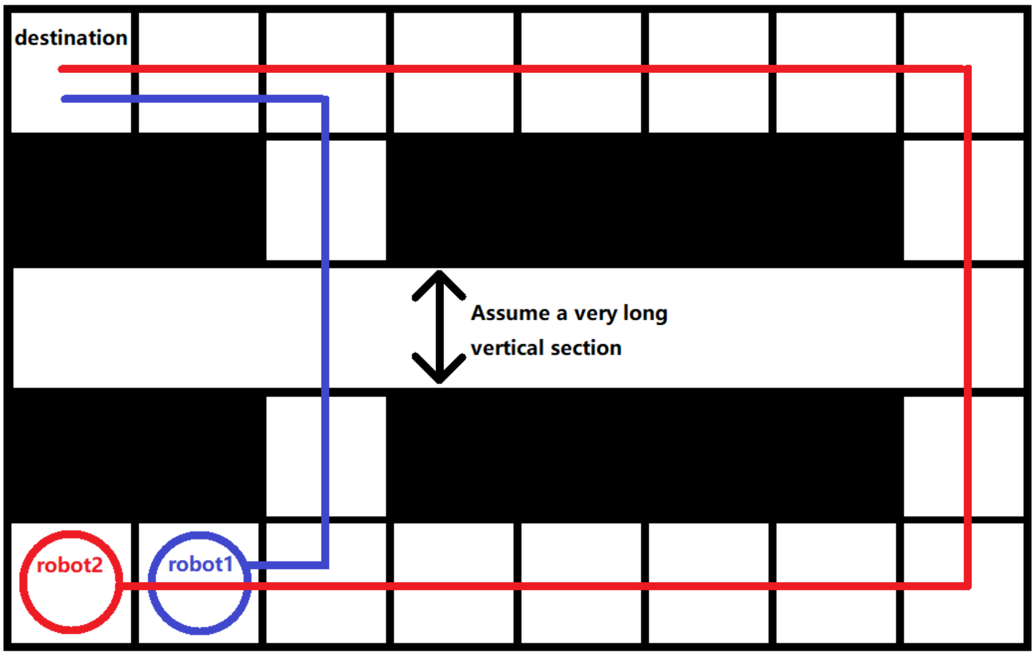


Figure 4.2.1. Dynamic Cost Issue

The last problem is that the mutual coordination strategy would create a starvation problem when a large number of robots are linked together in a line. Figure 4.2.2 shows an example. Note that the grids with green dot are the entrances, and grid directly connected to the ring road is for the robots to wait to enter the ring road. Thus, the robots pointed out in the figure are the robots that are waiting to enter. T robots wishing to use the grid would have to wait for all the robots in the line to leave. This issue is obvious in the ring road because all robots in the ring road are linked together and move in one direction. Other robots that wish to enter the ring road must wait for any of the robots in the ring road to reach their exit and break the link, then only other robots will have a chance to enter the ring road.

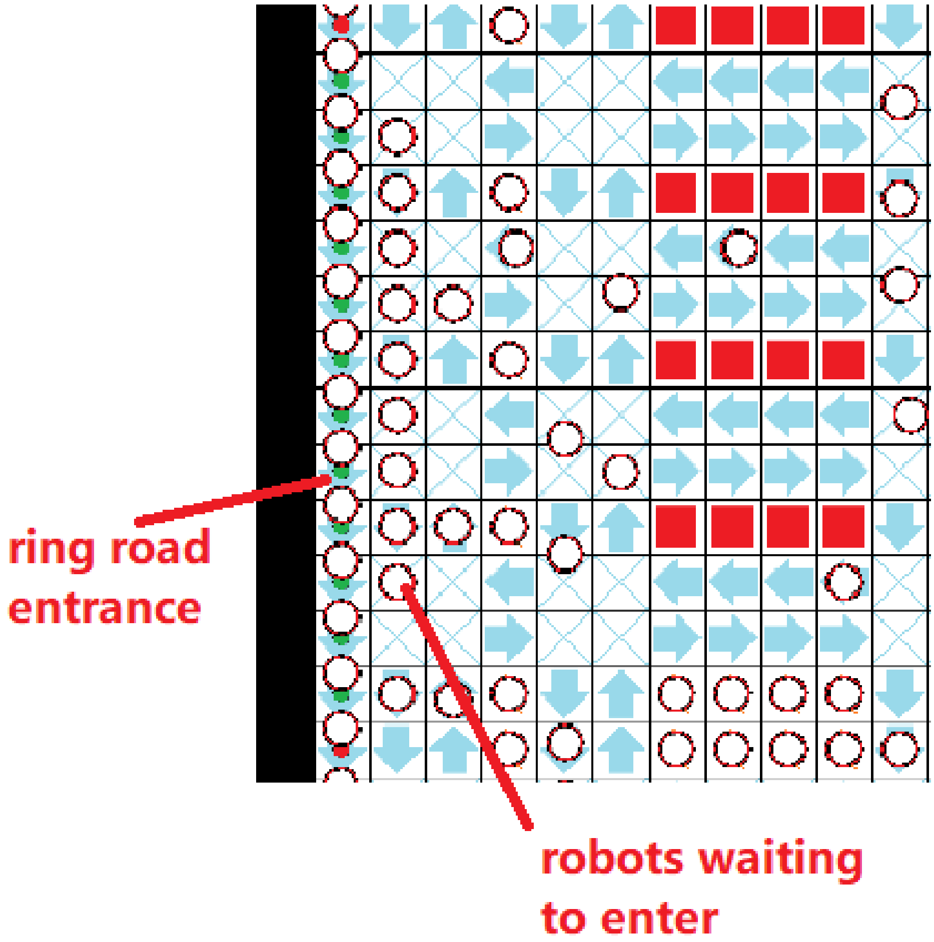


Figure 4.2.2. Starvation when Requesting to Enter the Ring Road

4.3 Concluding Remarks

The ring road has raised the peak throughput but requires more robots to achieve the peak. The ring road should not be used when the number of robots is less because its efficiency is significantly lower compared to a simple grid layout when the number of robots is less. It also limits the number of pick-up and drop-off points that can fit inside the map, so the ring road layout might not be suitable for sorting centers in large countries that need those extra drop-off points. The dynamic cost algorithm introduces additional overhead but is very effective in reducing congestion, so the strategy is still adopted. The mutual coordination would cause starvation, but it can ensure a smoother traffic flow on the ring road since all the robots are moving in one direction. Therefore, this strategy is also adopted.

Chapter 5

System Evaluation and Discussion

5.1 System Performance Definition

The unit used to measure the system's performance is throughput per hour. For parcel sorting, throughput is defined as the number of parcels sorted in a limited amount of time, which, in this case, is one hour.

5.2 Testing Setup and Result

Three different simulations were run under different settings with different number of robots to find their peak throughput. The settings are as below:

- Map without ring road, mutual coordination enabled.
- Map without ring road, mutual coordination disabled.
- Map with ring road, mutual coordination enabled.

All simulations were run for 10 hours to minimize the effect of random events that might result in inaccurate throughput. Figure 5.2.1 shows the throughput of the simulation results in a graph, and table 5.2.1 shows the actual numbers. For the map without a ring road, the simulations for mutual coordination enabled and disabled have similar throughputs before they reach their peak throughput. The simulation with mutual coordination enabled peak at 28801 with 660 robots, while another peak at 28508 with 640 robots. The mutual coordination strategy brings only a slight increase in the throughput. After reaching their peak, their throughput drops at a similar rate, but the one with mutual coordination enabled always has a slightly higher throughput. The map with ring road and mutual coordination enabled has significantly lesser throughput when the number of robots is less. It surpasses the peak of the map without ring road and mutual coordination enabled at 780 robots and continues to rise until it reaches a throughput of 32199 at 900 robots. The peak throughput has increased by (32199 - 28801) / 28801 * 100% = 12% after the ring road is introduced into the map layout. In overall, the ring road map improved the throughput but is less efficient than the map without a ring road. The results are not used to compare with other existing papers because different assumptions were made, and different algorithms were used. It is not fair and not conclusive to say the strategy used in this project is better or vice versa when different assumptions are made.

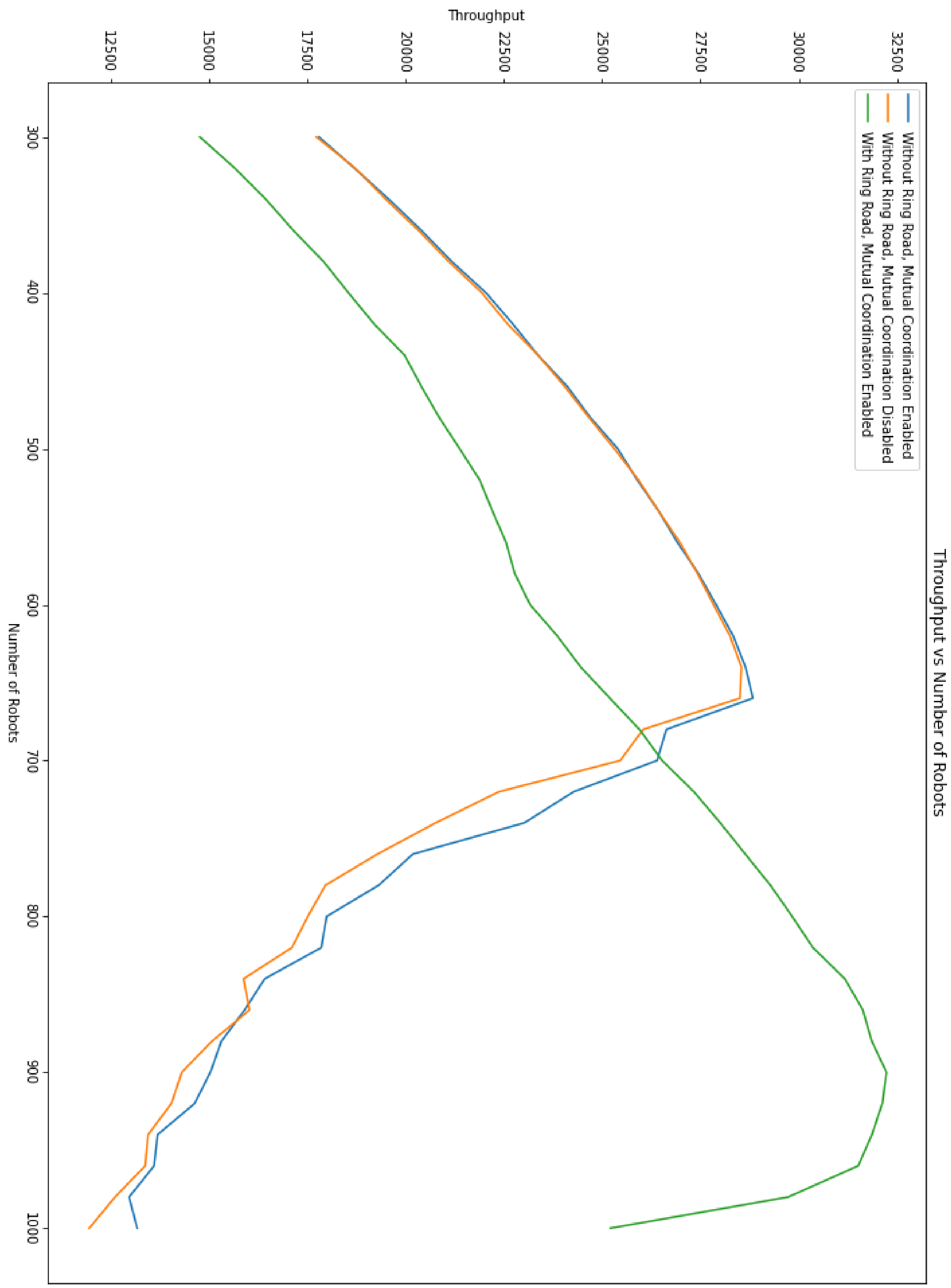


Figure 5.2.1. Graph of Simulation Results under Different Settings

Table 5.2.1. Throughput of Results under Different Settings

without ring road + mutual coordination disabled		without ring road + mutual coordination enabled		with ring road + mutual coordination enabled	
robots	throughput	robots	throughput	robots	throughput
300	17692	300	17760	300	14730
320	18681	320	18662	320	15617
340	19461	340	19539	340	16409
360	20292	360	20368	360	17109
380	21064	380	21140	380	17878
400	21906	400	22011	400	18511
420	22558	420	22693	420	19164
440	23335	440	23332	440	19939
460	24009	460	24077	460	20364
480	24630	480	24665	480	20821
500	25274	500	35362	500	21339
520	25889	520	25858	520	21849
540	26401	540	26403	540	22175
560	26942	560	26878	560	22518
580	27387	580	27412	580	22741
600	27799	600	27864	600	23131
620	28214	620	28297	620	23820
640	28508	640	28612	640	24419
660	28471	660	28801	660	25160
680	26010	680	26601	680	25913
700	25425	700	26362	700	26491
720	22342	720	24242	720	27298
740	20728	740	12984	740	27971
760	19249	760	20151	760	28606
780	17925	780	19279	780	29247
800	17476	800	17956	800	29799
820	17072	820	17820	820	30328
840	15842	840	16380	840	31138
860	15995	860	15869	860	31598
880	15038	880	15279	880	31825
900	14269	900	14997	900	32199
920	14006	920	14596	920	32093
940	13413	940	13655	940	31828
960	13339	960	13564	960	31481
980	12571	980	12930	980	29695
1000	11915	1000	13136	1000	25178

5.3 Objective Evaluation

The objectives of this project has been achieved, where the controlling framework and path finding algorithm have been developed, a ring road map layout has been proposed, and simulations as well as visualizations have been developed. The simulations were run for 10 hours, which greatly reduce the impact of random events in the simulations to increase the credibility of the results. The results obtained prove that the ring road can increase the peak throughput, but the layout only works when the number of robots are huge.

Chapter 6
Conclusion and Recommendation

6.1 Conclusion

To conclude, this project built a complete controlling framework and path-finding algorithms for the navigation of multiple robots in the sorting centers, while ensuring high throughput. This project also proposed a map layout that implements the ring road concept. The simulation is written in C#, and the visualization is achieved using Unity. The proposed map layout and the mutual coordination strategy, combined with the strategies adopted from other research papers, were proved to raise the peak throughput of the sorting centers. On a fixed 75 x 75 map size, the map with ring road has a throughput of 32199 while the map without ring road has only 28801. The throughput has increased by 12% using the ring road map. However, the proposed map layout is less efficient than the simple grid layout and will only be beneficial when the number of robots is high. The ring road map also requires more robots to reach the peak throughput. The ring road map peak at 900 robots, while the map without a ring rod peak at 660 robots.

6.2 Novelties of Work

The novelty of this project includes the implementation of the ring road concept into the map layout of sorting centers and a mutual coordination strategy that allows the robots in a line to accelerate and decelerate mutually. The existing research papers have yet to put any effort into examining the effect of map layout on the navigation of robots, and this project is the first to change the map layout. This project has implemented the ring road map layout, and the idea is that the robots that wish to travel a longer distance can take an alternative path so that they do not congest with other robots navigating in the high-traffic center areas. The mutual coordination strategy also allows robots in the ring road to accelerate and decelerate together, allowing robots to navigate faster in the ring road. The combination of the ring road map layout and the mutual acceleration strategy has achieved a higher peak throughput compared to the simple grid layout.

6.3 Future work

The simulation results show that the efficiency of the ring road has room to improve, especially when the number of robots is low. More rules and conditions can be set to control the usage of the ring road so that the ring road would not drag the throughput too much when the number of robots is low. The peak throughput can be improved as well. There are only two rules to meet to use the ring road, in which the number of robots using the ring road cannot exceed the number of grids in the ring road, and the robots are not allowed to use the ring road when the starting point or the destination is in the center area. When the number of robots is large, the number of robots in the ring road will always hit the threshold, and many robots still wish to enter the ring road. Perhaps the size of the center area could be changed dynamically according to the number of robots, or other rules can be set so that priority is given to those robots that need the ring road more.

The mutual coordination algorithm could also be modified so that the number of robots that can be linked together would not be too large. Adding this constraint would allow other robots a greater chance of competing for the grids. There should be a balance for the number of robots in the link to not cause severe starvation. The starvation problem would cause the robots to reach their destination slower and block the path of other robots, especially if the robots starve at the intersections. The throughput is expected to be improved if the balance can be found, as there will be less starvation.

REFERENCES

[1] Shopify, "Global Ecommerce Sales (2020–2025)," shopify.com. https://www.shopify.com/blog/global-ecommerce-sales (accessed July. 16, 2022).

[2] PWC, "Logistics in Malaysia Market overview and M&A trends," pwc.com. https://www.pwc.com/my/en/assets/blog/pwc-my-deals-strategy-logistics-in-malaysia.pdf (accessed July. 16, 2022).

[3] Intel, "What are Autonomous Mobile Robots," intel.com. https://www.intel.com/content/www/us/en/robotics/autonomous-mobile-robots/overview.html (accessed July. 16, 2022).

[4] Amware Fulfillment, "How to Reduce Warehouse Labour Costs with Smarter Order Picking," amwarelogistics.com. https://www.amwarelogistics.com/blog/reducing-warehouse-labor-costs-smarter-order-picking (accessed July. 16, 2022).

[5] Peruza, "TOP 4 Challenges Warehouses Face in Parcel Sorting," peruza.com. https://peruza.com/blog/parcel-sorting/top-4-challenges-warehouses-face-in-parcel-sorting/ (accessed July. 17, 2022).

[6] M. Balqish-Salleh, "Warehousing: No better time to automate," The Edge Malaysia, para. 10, May 9, 2022. [Online]. Available: https://www.theedgemarkets.com/article/warehousing-no-better-time-automate. [Accessed: Jul. 17, 2022].

[7] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," European Journal of Operational Research, vol. 294, no. 2. Elsevier B.V., pp. 405–426, Oct. 2021. Accessed Jul. 17, 2022. doi: 10.1016/j.ejor.2021.01.019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221721000217

[8] J. Myers, "The Paradox of In-Demand Jobs and Global Talent Shortage," weforum.org. https://www.weforum.org/agenda/2022/05/most-in-demand-professions-list-2022/ (accessed July. 18, 2022).

[9] A. Stark, "Want to Reduce Warehouse Costs? Think Warehouse Labour Management," kanelogistics.com. https://www.kanelogistics.com/blog/reduce-costs-warehouse-labor-management (accessed July. 18, 2022).

[10] Malaysian Communication and Multimedia Commission, "Postal & Courier, Facts & Figures 4Q 2021," MCMC, Cyberjaya, SGR, MY, 2021. Accessed on: Jul., 18, 2022. [Online]. Available: https://www.mcmc.gov.my/skmmgovmy/media/General/P-C-Q4_220412_BI_PDF.pdf

[11] Malaysian Communication and Multimedia Commission, "Consumer Satisfaction for Courier 2019," MCMC, Cyberjaya, SGR, MY, 2019. Accessed on: Jul., 18, 2022. [Online]. Available: https://www.mcmc.gov.my/skmmgovmy/media/General/pdf/Consumer-Satisfaction-Survey-Courier-2019.pdf

[12] Parcel Perform & iPrice Group, “Consumers are still Not Happy with their E-commerce Delivery Experience, a New Survey by Parcel Perform and iPrice Group reveals” iprice.my. https://iprice.my/trends/insights/consumers-are-still-not-happy-with-their-e-commerce-delivery-experience-a-new-survey-by-parcel-perform-and-iprice-group-reveals/

[13] N. V. Kumar and C. S. Kumar, “Development of collision free path planning algorithm for warehouse mobile robot,” in Procedia Computer Science, 2018, vol. 133, pp. 456–463. doi: 10.1016/j.procs.2018.07.056.

[14] C. H. Ch’ng, S. Y. Liew, C. S. Wong, and B. Y. Ooi, “An Efficient Multi-AMR Control Framework for Parcel Sorting Centers,” Mar. 2020. doi: 10.1109/SAS48726.2020.9220043.

[15] Z. Zhang, Q. Guo, J. Chen, and P. Yuan, “Collision-Free Route Planning for Multiple AGVs in an Automated Warehouse Based on Collision Classification,” IEEE Access, vol. 6, pp. 26022–26035, Mar. 2018, doi: 10.1109/ACCESS.2018.2819199.

[16] Y. Guo, H. Huang, F. Gao, and X. Qiao, “Improving the Performance of Multi-AGV Systems with a Dynamic Unlock Algorithm,” in Proceedings - 10th International Conference on Intelligent Computation Technology and Automation, ICICTA 2017, Oct. 2017, vol. 2017-October, pp. 39–43. doi: 10.1109/ICICTA.2017.16.

[17] M. Muller, J. H. Ulrich, T. Reggelin, S. Lang, and L. S. Reyes-Rubiano, “Comparison of Deadlock Handling Strategies for Different Warehouse Layouts with an AGVS,” in Proceedings - Winter Simulation Conference, Dec. 2020, vol. 2020-December, pp. 1300–1311. doi: 10.1109/WSC48552.2020.9383883.

APPENDIX

1. The C# codes to run on visual studio and unity player:
   https://github.com/biao135/Multi-Robot-Path-Planning-Algorithm-for-Automated-Parcel-Sorting-Centre-with-Ring-Road-Layout

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 2**
Student Name & ID: Jong Qian Biao 1901846	
Supervisor: Dr Liew Soung Yue	
Project Title: Multi-Robot Path-Planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout	

1. WORK DONE

Refresh the work of last semester.

2. WORK TO BE DONE

Acceleration of robots, Mutual coordination between the robots.

3. PROBLEMS ENCOUNTERED

Robots would never stop at the desired position.
Map is too small, robots are too few.

4. SELF EVALUATION OF THE PROGRESS

On track.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 4**
Student Name & ID: Jong Qian Biao 1901846	
Supervisor: Dr Liew Soung Yue	
Project Title: Multi-Robot Path-Planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout	

1. WORK DONE Acceleration of robots, mutual coordination of robots.
2. WORK TO BE DONE Traffic light system, change map layout, expand map size, increase number of robots.
3. PROBLEMS ENCOUNTERED Traffic light system does not work if the number of grids for queueing is too few.
4. SELF EVALUATION OF THE PROGRESS On track.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 6**
Student Name & ID: Jong Qian Biao 1901846	
Supervisor: Dr Liew Soung Yue	
Project Title: Multi-Robot Path-Planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout	

1. WORK DONE Investigated traffic light system and found that it does not work, finalized map layout.
2. WORK TO BE DONE Increase number of robots, expand map.
3. PROBLEMS ENCOUNTERED Unity runs slower and slower when number of robots increases.
4. SELF EVALUATION OF THE PROGRESS Need to rush a bit.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 8**
Student Name & ID: Jong Qian Biao 1901846	
Supervisor: Dr Liew Soung Yue	
Project Title: Multi-Robot Path-Planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout	

1. WORK DONE
Increase number of robots, expand map size. Fixed the issue where unity runs slower when number of robots increases.
2. WORK TO BE DONE
Find strategy that smooths the robots when moving in and out the ring road.
3. PROBLEMS ENCOUNTERED
Robots might collide with each other when moving in curves. Deadlock happens in curves.
4. SELF EVALUATION OF THE PROGRESS
Need to rush a bit.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 10**
Student Name & ID: Jong Qian Biao 1901846	
Supervisor: Dr Liew Soung Yue	
Project Title: Multi-Robot Path-Planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout	

1. WORK DONE
Solved collisions and deadlock in curve. Reduced the cost of ring road.
2. WORK TO BE DONE
Reduce the cost of ring road. Modify the code to be able to run outside of unity. Generate simulation results.
3. PROBLEMS ENCOUNTERED
Simulation runs too slow in unity.
4. SELF EVALUATION OF THE PROGRESS
Need to rush a bit.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 12**
Student Name & ID: Jong Qian Biao 1901846	
Supervisor: Dr Liew Soung Yue	
Project Title: Multi-Robot Path-Planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout	

1. WORK DONE Moved the code to run outside of unity. Simulation result generated.
2. WORK TO BE DONE Write report.
3. PROBLEMS ENCOUNTERED No problem.
4. SELF EVALUATION OF THE PROGRESS On track.

Supervisor's signature

Student's signature

POSTER

Multi-robot Path-planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout

Student name: Jong Qian Biao Supervisor: Dr Liew Soung Yue

Abstract

Recent approaches to parcel sorting systems use a simple grid layout for their road system, which have less effect in resolving traffic jam issue at the core area of the sorting centers. This project implements an outer ring-road concept in addition to the inner grid layout because robots that travel a longer distance can travel in an alternative path to avoid congesting with other robots in high traffic center area. This project also evaluates the recent strategies and proposes new strategies for avoiding collisions and resolving deadlocks while ensuring high throughput for the parcel sorting system.

Figure 1. Ring road example

Objectives

1. To design a map layout that implements the ring-road concept.
2. To develop a controlling framework that could avoid collisions and solve deadlocks between robots.
3. To develop a path-finding algorithm that is computationally inexpensive to the extent that makes real-time operations possible.
4. To develop continuous event simulations for the parcel sorting system with C# in Unity game engine.

Problem Statements

1. Parcel sorting job is boring and exhausting to human.
2. Inefficiencies and human errors in manual parcel sorting.
3. Avoiding collisions and solving deadlocks between the robots while ensuring high throughput in automated parcel sorting system.

Approaches

1. Grid reservation strategy to avoid collisions.
2. Dynamic grid cost strategy to reduce congestion.
3. Unidirectional lanes and intersections to avoid type 1 deadlocks.
4. Detects and breaks type 2 (circular-wait) deadlocks.
5. Ring road map layout to provide alternative path for robots that travel a longer distance.
6. Mutual acceleration strategy so that the robots in a line can accelerate and decelerate at the same time.

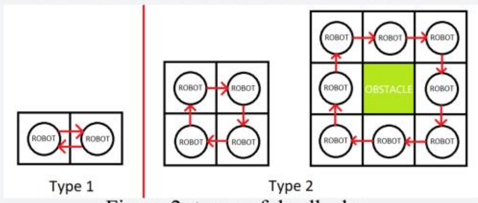


Figure 2: types of deadlocks

Results

The map size is fixed at 75 x 75 grids and all simulations were run for 10 hours to minimize inaccurate throughput due to random events. For the map without a ring road, enabling or disabling mutual coordination have similar throughputs until they reach their peaks. With mutual coordination, the peak throughput is 28801 with 660 robots while the another is 28508 with 640 robots. The map with ring road and mutual coordination enabled has significantly lower throughput when the number of robots is less. It surpasses the peak of the map without ring road and mutual coordination enabled at 780 robots and reaching 32199 at 900 robots. The peak throughput has increased by 12% after the ring road is implemented.

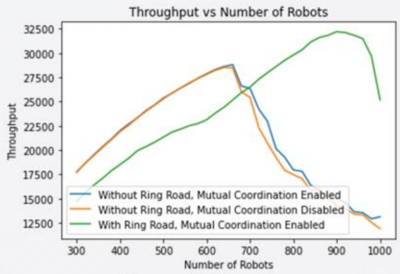


Figure 3. Results

Conclusion

The mutual coordination strategy and the ring road map layout proposed have increased the peak throughput. However, the ring road map layout is less efficient when the number of robots is less. Future work include adding new rules for prioritizing ring road for robots that need the ring road more and reduce starvation problem created by the mutual coordination strategy.

PLAGIARISM CHECK RESULT

Turnitin Originality Report

Multi-Robot Path-Planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout by Qian Biao Jong

From FYP2 (Final Year Project 2023)

Similarity Index	Similarity by Source	
5%	Internet Sources:	2%
	Publications:	4%
	Student Papers:	1%

Processed on 24-Apr-2023 23:14 +08
ID: 2074084814
Word Count: 9860

sources:

1 — 1% match (Chee-Henn Ch'ng, Soung-Yue Liew, Chee-Siang Wong, Boon-Yaik Ooi. "An Efficient Multi-AMR Control Framework for Parcel Sorting Centers", 2020 IEEE Sensors Applications Symposium (SAS), 2020)
Chee-Henn Ch'ng, Soung-Yue Liew, Chee-Siang Wong, Boon-Yaik Ooi. "An Efficient Multi-AMR Control Framework for Parcel Sorting Centers", 2020 IEEE Sensors Applications Symposium (SAS), 2020

2 — 1% match (Zheng Zhang, Qing Guo, Juan Chen, Peijiang Yuan. "Collision-Free Route Planning for Multiple AGVs in Automated Warehouse Based on Collision Classification", IEEE Access, 2018)
Zheng Zhang, Qing Guo, Juan Chen, Peijiang Yuan. "Collision-Free Route Planning for Multiple AGVs in Automated Warehouse Based on Collision Classification", IEEE Access, 2018

3 — <1% match (student papers from 26-Aug-2021)
Submitted to Universiti Tunku Abdul Rahman on 2021-08-26

4 — <1% match (student papers from 20-Apr-2022)
Submitted to Universiti Tunku Abdul Rahman on 2022-04-20

5 — <1% match (student papers from 15-Apr-2022)
Submitted to Universiti Tunku Abdul Rahman on 2022-04-15

6 — <1% match (Internet from 23-Jun-2022)
https://academic.hep.com.cn/fem/article/2022/2095-7513/30877

7 — <1% match (Lin Zhang, Yingjie Zhang, Yangfan Li. "Path Planning for Indoor Mobile Robot Based on Deep Learning", Optik, 2020)
Lin Zhang, Yingjie Zhang, Yangfan Li. "Path Planning for Indoor Mobile Robot Based on Deep Learning", Optik, 2020

8 — <1% match (Marcel Muller, Jan Hendrik Ulrich, Tobias Reggelin, Sebastian Lang, Lorena S. Reyes-Rubiano. "Comparison of Deadlock Handling Strategies for Different Warehouse Layouts with an AGVS", 2020 Winter Simulation Conference (WSC), 2020)
Marcel Muller, Jan Hendrik Ulrich, Tobias Reggelin, Sebastian Lang, Lorena S. Reyes-Rubiano. "Comparison of Deadlock Handling Strategies for Different Warehouse Layouts with an AGVS", 2020 Winter Simulation Conference (WSC), 2020

9 — <1% match ()
Seladji, Ilham. "Automatic Pain Assessment Through Facial Expressions", RIT Scholar Works, 2020

10 — <1% match (Internet from 13-Dec-2022)
https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=12205&context=theses

11 — <1% match (Internet from 05-Feb-2023)
https://www.mdpi.com/2071-1050/13/10/5644/xml

12 — <1% match (Pengfei Lin, Jiancheng Weng, Siyong Ma, Shuo Xu. "Data-Driven Operation Characteristics Analysis of Traditional Taxis and Transportation Network Company Cars", CICTP 2017, 2018)

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Jong Qian Biao
ID Number(s)	19ACB01846
Programme / Course	Bachelor of Computer Science
Title of Final Year Project	Multi-Robot Path-Planning Algorithm for Automated Parcel Sorting Centre with Ring Road Layout

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 5 % **Similarity by source** Internet Sources: 2 % Publications: 4 % Student Papers: 1 %	Within the required range.
Number of individual sources listed of more than 3% similarity: 0	Within the required range.
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	Signature of Co-Supervisor
Name: Liew Soung Yue	Name:
Date: 28/4/2023	Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	19ACB01846
Student Name	Jong Qian Biao
Supervisor Name	Dr Liew Soung Yue

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____
(Signature of Student)
Date: 24/4/2023