University of BRISTOL

Department of Electrical and Electronic Engineering

MSc in Image and Video Communications and Signal Processing

Estimating the Performance of Robotic Mobile Fulfillment System (RMFS) Considering the Order Priorities

Yutao Xiong

iz22144

September 2023

DECLARATION AND DISCLAIMER

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Taught Postgraduate Programmes and that it has not been submitted for any other academic award.

Except where indicated by specific reference in the text, this work is my own work. Work done in collaboration with, or with the assistance of others, is indicated as such. I have identified all material in this dissertation which is not my own work through appropriate referencing and acknowledgement. Where I have quoted from the work of others, I have included the source in the references/bibliography.

Any views expressed in the dissertation are those of the author.

The author confirms that the printed copy and the electronic version of this thesis are identical.

Signed: Yutao Xiong

Date: 29/8/2023

Abstract

Many logistics warehouses utilise Robotic Mobile Fulfillment Systems (RMFS) due to their efficiency, speed, and adaptability. Although different priority orders are quite common within the logistics industry, there has been little research investigating the impact of order priorities on system performance using an analytical model.

This project outlines the ordering procedure by constructing a Semi-Open Queuing Network (SOQN) and modifies and optimises the conventional Approximation Mean Value Analysis (AMVA) algorithm to produce a priority-based AMVA algorithm. The newly proposed algorithm is used to estimate the steady-state system performance.

Taking the order priority through numerical simulation tests (using Anylogic) to validate the results obtained from QN, our approach shows that the analytical model and the newly proposed algorithm can precisely predict the performance of RMFS. When compared to the simulation model, our analytical model's runtime is more than a factor of 100 faster and the performance estimations have an accuracy of at least 90% correct.

Contents

Chapter 1 Introduction

1.1 Background

In recent years, the rapid growth of e-commerce and online shopping has led to a significant increase in demand for efficient order fulfillment systems. These systems play a crucial role in ensuring that customer orders are processed, packed, and shipped promptly, meeting customer expectations and enhancing brand loyalty [1]. To meet these demands, companies have turned to automation solutions, such as Robotic Mobile Fulfillment Systems (RMFS), to streamline their order fulfillment processes.

RMFS was first introduced in a scholarly publication [2], and its patent rights were subsequently acquired by Kiva Systems, which is now a subsidiary of Amazon Robotics [3]. The robots depicted in Figure 1 are capable of directly lifting and moving racks throughout a warehouse. It has been determined that the implementation of robotics and automation in the domain of RMFS significantly reduces the time and effort required for manual picking and packing, resulting in increased levels of efficiency and productivity. In addition, RMFS features a high degree of flexibility and scalability, allowing for effortless adaptation to changing business requirements.

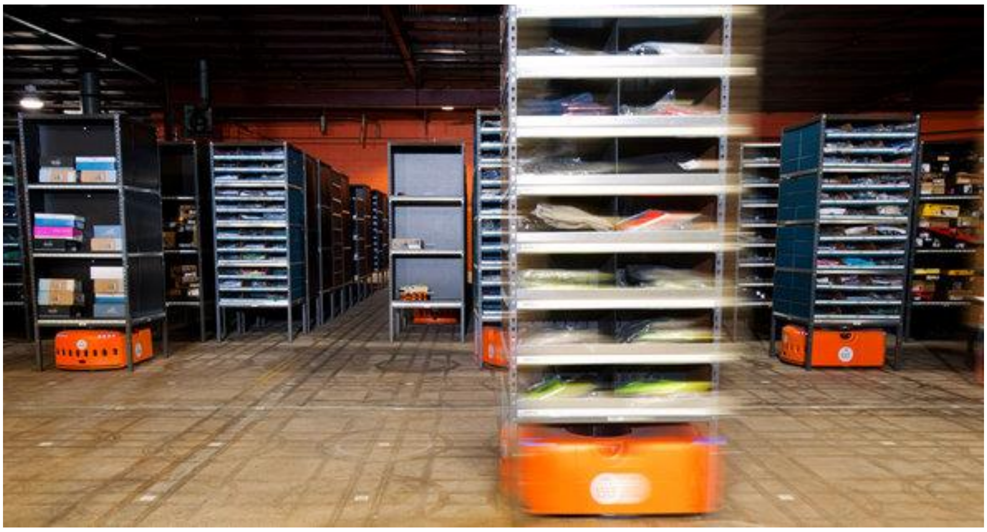


Figure 1. The robots used in RMFS[1].

Order prioritisation is a fundamental strategy in supply chain management, catering to the diverse expectations of modern customers. This practice strategically allocates resources and streamlines workflows to ensure timely order fulfillment, considering factors like customer urgency, order value, and service agreements. Notably, the Prime membership model by Amazon exemplifies how order prioritisation enhances customer loyalty, operational efficiency, and market competitiveness in today's fast-paced commerce landscape.

1.2 Aim and Objectives

The problem of this project is to estimate the performance of such robotic system by calculating the workstation queuing length, the order cycle time (from the time an order enters the system until it exits the system), and the robot and workstation utilisation. Building a simulation model is one solution, but it is time-consuming and extremely difficult to adapt to the changes. The alternative method, which is extensively employed these years, is to analyse using a queuing network. In comparison to the simulation model, the duration of this type of model can be reduced by hundreds of times. However, the accuracy will inevitably de-

[1] https://archive.nytimes.com/dealbook.nytimes.com/2012/03/19/amazon-com-buys-kiva-systems-for-775-million/

crease, so it is a challenge to increase the accuracy of the analytical model. The aim of this project is to estimate the performance of RMFS using SOQN and an appropriate algorithm considering the order priorities.

The aim of this project can be broken down into several objectives:

- Enable automated warehouse layout generation (via the entry of a few parameters).
- Construct a queueing network based on RMFS operation process considering order priority.
- Implement a path planning algorithm to calculate the robot movement time.
- Modify the traditional AMVA algorithm to adapt to situations caused by different priorities in order to increase the accuracy.
- Multiple sets of experiments are conducted to validate the accuracy of the analytical model and the newly proposed algorithm using a simulation model.

1.3 Innovation

The phenomenon of “prioritisation” can be seen everywhere in real life, as evidenced by Amazon's Prime membership mechanism. There have been a number of academics who have studied how to estimate the performance of RMFS using many sorts of queuing network. Nevertheless, there has been a lack of research examining the impacts arising from the prioritising of orders. To the best of my knowledge, this is the first thesis that study the performance of RMFS considering the order priorities. This study presents the construction of a SOQN that incorporates priority considerations and adapts the conventional AMVA algorithm to address issues such as queue-jumping or preemption resulting from varying priorities.

1.4 Outline

The remainder of this thesis is as follows:

- Chapter 2: This chapter focuses on some of the literature that has been used in studying the performance of RMFS; there is also some related literature on priority queuing networks.
- Chapter 3: This chapter discusses the warehouse layout, the queuing network model, and the calculation of travel times as they pertain to the modelling of this project.
- Chapter 4: This chapter highlights on the algorithm's design, including the problem description and algorithm implementation.
- Chapter 5: This chapter describes some simulation experiments to demonstrate the accuracy of the proposed methodology.
- Chapter 6: This chapter provides a summary of the paper's major contributions and innovations, as well as a discussion of its limitations and improvement opportunities.

Chapter 2 Literature Review

2.1 Performance of RMFS

The queuing network model is an appropriate tool for assessing and improving the efficacy of robotic systems because it has the ability to methodically display the many processing phases involved in order fulfilment. Queuing network models are used by researchers to assess how different optimisation tactics, such as robot assignment and zone allocation, affect the system's overall effectiveness.

To study RMFS throughput using Markov chain models, Roy et al. [4] create closed queueing network models. These models include single-class queueing network models and multi-class queueing network models for order picking and replenishment procedures. Despite the authors' thorough examination and construction of several CQNs to assess the effectiveness of RMFS under diverse circumstances, the CQNs are fundamentally limited. The results' accuracy is hampered by the CQNs' inflated order volume and very high predicted throughput [5].

To evaluate the effectiveness of RMFS and establish the ideal number and speed of robots for each scenario, Yuan et al. [6] build two OQNs. The many robot subtypes, such as pooled and dedicated robots, are also discussed in the study. The study estimates the ideal number and speed of robots, takes into account system congestion, and simulates the analytical models to verify them. Despite the OQN model's efficiency and simplicity, it is essential to recognise that resource availability cannot always be guaranteed in practical applications. This constraint may influence the precision of system performance analysis.

A more realistic portrayal of scenarios with constrained resources and external demand inputs is offered by the semi-open queueing network (SOQN) model. In Figure 2, the generic SOQN model is displayed. Using the SOQN models, the overall effectiveness of RMFS and the accompanying robot utilisation in various warehouse designs are assessed by Lamballais et al. [7]. Due to the paucity of prior research on robot employment, this study is unique. The authors create four different models for analysis, considering single-line orders, multi-line orders, and zone division or not. The performance of the RMFS system is examined by Zou et al. [8] utilising SOQN while taking into account a variety of assignment rules. They also provide a guideline for assignments depending on the processing speed of each workstation. They develop a neighbourhood search technique to iteratively find the best assignment rule as well. To measure the performance of RMFS in the setting of the dynamic order arrival rates that define the present e-commerce industry, Duan et al. [9] use Markov-modulated Poisson Processes (MMPP) and SOQN. Through experimentation, they are able to show the advantages of time-varying estimate. By analysing the performance of RMFS under conditions of dynamic order arrival, this work makes a unique contribution. To solve the situation of anomalous order matching, which may have been brought about by earlier models failing to take the shelf condition into account, Kuang [10] suggests a "secondary picking" technique using a SOQN model. This study then develops a matching queuing network model based on the "secondary picking" method and suggests a "shelf designation" technique to lower the robot's journey distance during order fulfilment, consequently reducing the order turn-over time.

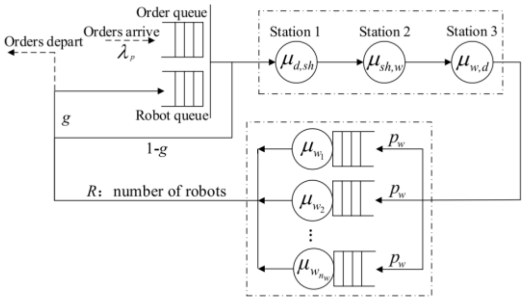


Figure 2. The SOQN model in [9]

2.2 Solutions of SOQN

For CQNs and SOQNs, the Matrix-geometric method (MGM) and AMVA are the two most common solutions.

The MGM [11] involves representing the system's state transitions as a matrix and using properties of matrices to analyse the steady-state behaviour of the system. MGM is especially useful when dealing with networks where customer movements between different service centres can be modelled using matrices.

The AMVA algorithm [12] is another technique employed to estimate the performance metrics of queueing networks, offering a simplified approach to complex systems like the SOQN. AMVA involves approximating the queueing network as a set of interconnected service centres, each representing a distinct component. By assuming that each service centre behaves like a single queue, AMVA allows the application of classical queueing theory techniques to estimate performance metrics.

2.3 Priority Queuing Network

Prioritisation is ubiquitous in everyday life and can result in a variety of situations, including queue jumping and preemption. Several academics have conducted research on priority queuing networks.

Bolch et al. [13] introduce prioritisation to queuing networks and studied both the case of a single station and the combined case of multiple stations. The stations are separated into two distinct categories: PR (preemptive) and HOL (non-preemptive). They propose an extension of the MVA (PRIOMVA) algorithm for open and closed orders, but the proposed steps are too complicated and inapplicable to SOQN.

Raymond et al. [14] present an efficient and accurate method for computing the average performance measures of priority queuing networks, called the MVA Priority Approximation. The method is based on the Mean Value Analysis (MVA) technique and is designed to handle networks with multiple priority classes and preemptive priority disciplines. Based on this paper, Eager et al. [15] improve it by substituting the exact MVA algorithm with the approximation MVA algorithm, which is able to reduce the computational complexity with roughly the same degree of precision. But their approach doesn't take into account external order arrivals, so it only applies to CQNs. The specifics will be explained in the following chapters.

Qi et al. [16] estimate the performance of smart compact automated parking system (CAPS) considering the service priorities. They build a priority queuing network to analyse the process and develop approximation methods to solve it. The priority queuing network model is shown in Figure 3.

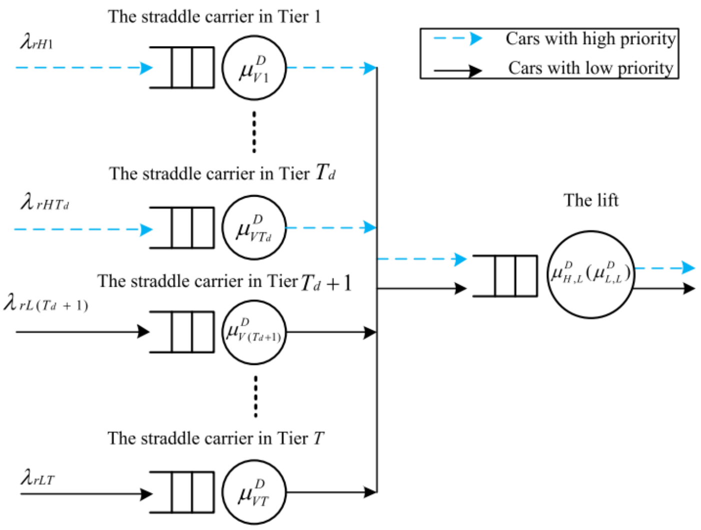


Figure 3. The priority queuing network for a CAPS [16]

Chapter 3 Model Description

3.1 Problem Description

3.1.1 Warehouse Layout

Figure 4 depicts an exemplary layout of the RMFS warehouse in a top view with workstations on three sides: the movable shelves are stored in a rectangular area with dimensions W by L. The workstations are evenly distributed around the shelves (typically, no workstations are located on the side near the door for space adjustment or restocking of the shelves' contents). The storage area's dark grey squares indicate that the shelves are at the corresponding storage point, while the shelving area's light grey squares indicate that the shelves are not at the corresponding storage point but are on their way to complete the order selection. The storage area is divided into 5×2 blocks by intersecting aisles and cross aisles because Lamballais et al. [7] determined that system throughput efficiency is greatest at this ratio, and this thesis adheres to their conclusion. In the unloaded state, the robot can pass through the bottom of the shelf to reach the target shelf, whereas in the loaded state, it can only advance along the transverse longitudinal aisle to reach the target location.

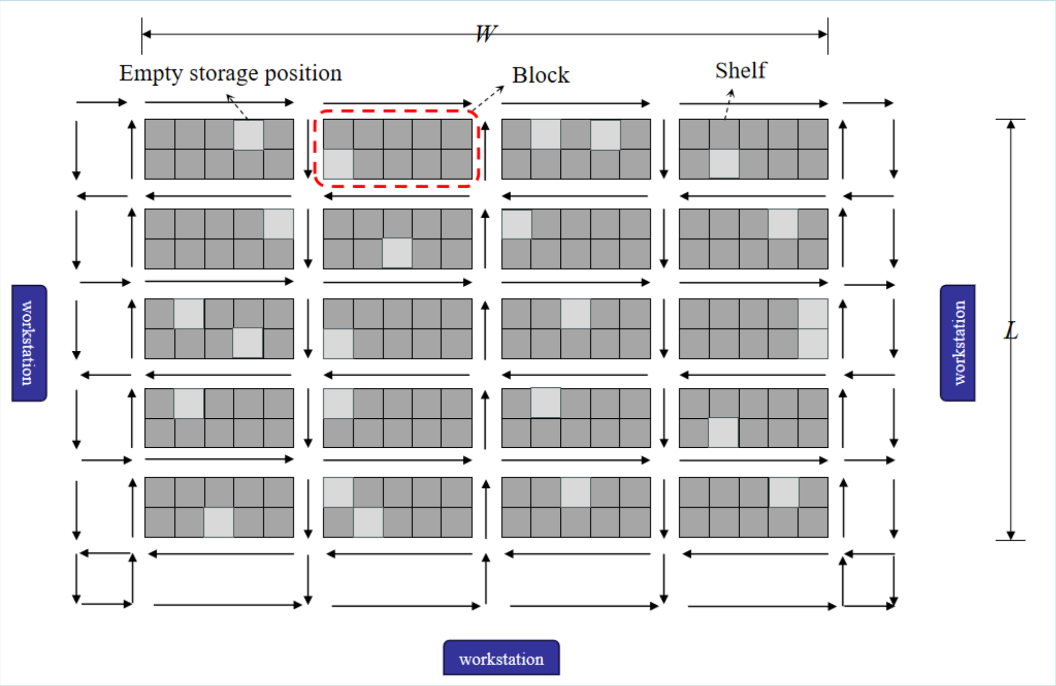


Figure 4. The top view of the warehouse

It should be noted that, unlike the research on the scheduling problem during system operation, the research in this paper focuses on the evaluation of the overall efficiency of the 'parts-to-picker' warehouse system prior to operation. Therefore, it does not involve real-time scheduling of vehicles, and it is not possible to simulate anomalies such as deadlocks and collisions in the real-time scheduling process. In order to avoid deadlocks or blockages at intersections during the common scheduling process of multiple robots, this thesis stipulates that when designing the warehouse routes, aisles and cross aisles, and routes leading to workstations outside the shelf storage area are unidirectional.

3.1.2 Assumptions

To evaluate the overall performance of RMFS and develop the corresponding queuing network model, the following assumptions are made in this thesis:

1) Orders are triggered randomly, and each shelf has the same probability of being selected.
2) Aisles are all unidirectional (except for the routes next to the workstations).
3) Waiting delays at intersections are negligible.
4) Damage and power loss during robot operation are not considered.
5) The robot maintains a constant speed while moving.
6) The replenishment process required by the depletion of goods in the shelf is not considered.
7) The picking time of a workstation for each pod follows a uniform distribution U [a, b].
8) Order arrivals obey a Poisson distribution.
9) The robot moves the shelf from the workstation to the storage area after the picker removes the goods from it. The robot employs the point of service completion (POSC) dwell point policy when storing the shelf at an open storage position rather than waiting at a specified dwell point.
10) Specific priority orders can only be handled by a specific type of robots to avoid competition for resources and to control order spikes.
11) Different types (priorities) of robots share the same workstation, so some situations like queue-jumping or preemption may occur.

3.1.3 Symbol Description

The symbols in this thesis are illustrated in Table 1.

Table 1. Symbol Description.

Symbol	*Meaning*
N_s	The number of shelves.
w	The width of each grid in the warehouse.
λ^s	The order arrival rate of the class s order.
t_l, t_d	Time for the robot to lift and lower the shelves.
v	The speed of each robot.
W	The width of the storage area.
L	The length of the storage area.
S	The total number of priority levels.
M	The total number of stations excluding synchronization station
c_m	The number of servers at station *m.*
n_{wl}, n_{wb}, n_{wr}	The number of workstations located on the left/on the bottom/on the right
$\vec{N}$	The full population vector.
$\vec{n}$	The current population vector
n	The number of robots in population vector $\vec{n}$.
σ_w^2	The variance of workstation service time.
ES_m	The first moment of the service time of station *m.*
ES_m^2	The second moment of the service time of station *m.*
$ES_{rem,m}$	The expected time remaining until the first departure at station *m.*
$L_m^s(\vec{n})$	The expected class *s* robot queue length including robots in service at station *m* when the system contains $\vec{n}$ robots.
$L_s(\vec{n})$	The expected robot queue length at synchronization station *s* when the system contains $\vec{n}$ robots.
$\tilde{L}_m^s(\vec{n})$	The expected class *s* robot queue length excluding robots in service at station *m* when the system contains $\vec{n}$ robots.

Continued Table 1

L_{ext}^{s}	The external queue length of class *s* order.
$\lambda_{mr}^{s}(\vec{n})$	The arrival rate of class *r* robots when a class *s* robot is waiting at station *m* when the system contains $\vec{n}$ robots.
$Q_m(\vec{n})$	The probability that all servers are busy at station *m* when the system contains $\vec{n}$ robots.
$p_m(i\,\|\,\vec{n})$	The probability that there are *i* robots at station *m* when the system contains $\vec{n}$ robots.
$TH^{s}(\vec{n})$	The throughput of the order of priority *s* when the system contains $\vec{n}$ robots.
$P^{s}(\vec{n})$	The probability that the orders being served in the station when the order of priority *s* arrives all have a priority higher than or equal to it when the system contains $\vec{n}$ robots.
$ET_m^{s}(\vec{n})$	The expected residence time for an order (priority *s*) at station *m* when there are n robots in the system.
$ET_s(\vec{n})$	The time required to complete the current order (priority *s*) at synchronization station *s* when the system contains $\vec{n}$ robots.
ρ^{s}	The utilisation of class s robots.
t_{oc}^{s}	The class s order cycle time.
ρ_{ws}	The utilisation of workstations.
v_m	The visit ratio of station *m.*

3.2 SOQN Model

The objective of "parts-to-picker" warehouse scheduling is to complete order picking using warehouse resources (robots, racks, workstations, and workers). In this process, the robots in the warehouse are fixed resources, and whether the orders in the order pool can be executed depends on whether there are available robots, and the robots in the process of completing the corresponding tasks are in one-to-one correspondence with the tasks, i.e., the number of robots operating in the warehouse corresponds to the number of tasks being served. As a system with resource constraints (limited robots) and external order arrivals, it is not possible to construct either an OQN (because of resource constraints on internal services, orders must wait for free robots before they can be executed) or a CQN, and thus the construction of the SOQN shown in Figure 5 provides a more accurate description of the operation of the system.

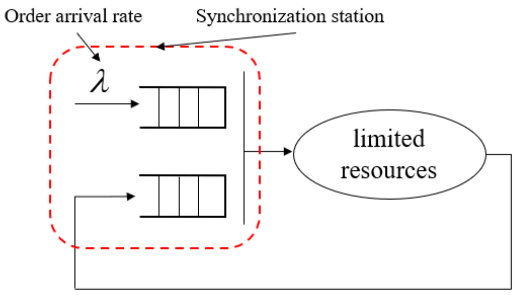


Figure 5. A SOQN model

3.2.1 Model Construction

Using prior knowledge regarding queuing networks, a SOQN model can be constructed for this undertaking, as depicted in Figure 6.

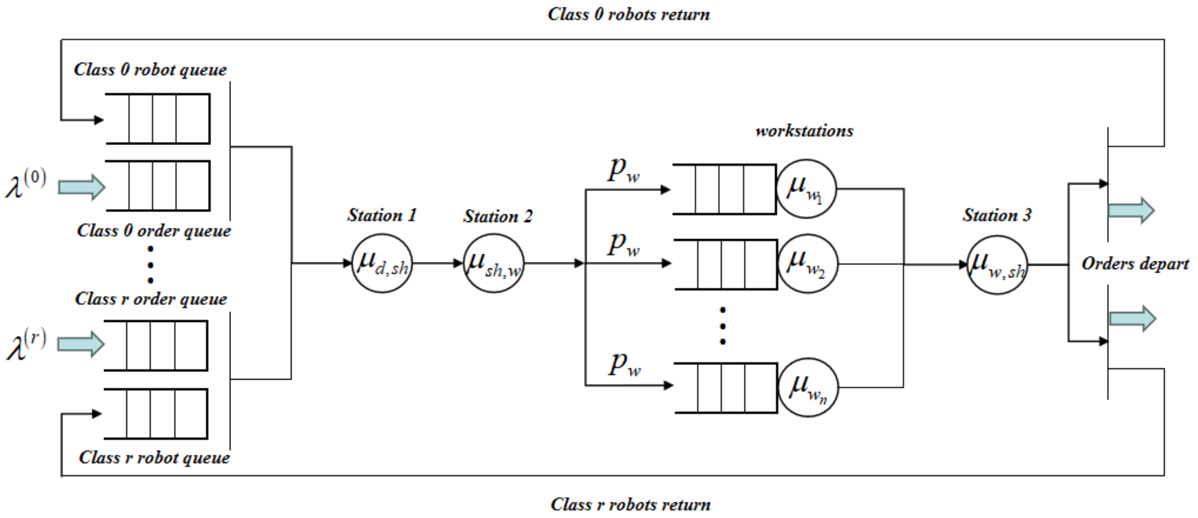


Figure 6. The SOQN model of this project

Model specific explanations will be elaborated in the next subsection.

3.2.2 Order Fulfillment Process

In conjunction with Figure 6, the queuing network basically describes the entire process of an order from the time it enters the system to the time it leaves the system, as follows:

When a type of order arrives in the system at rate λ, it enters the external queue and waits until there is an available robot (with the same type) in the external robot queue to match it. Then they entered the system together. The process in the system relies on the three basic robot movements shown in Figure 7.

An idle robot moves from its dwell place to the target shelf corresponding to the order as soon as it is bound to the corresponding order in the first step. In this process the robot is in an unloaded state, so it can pass under the shelves. This is *Move1*.

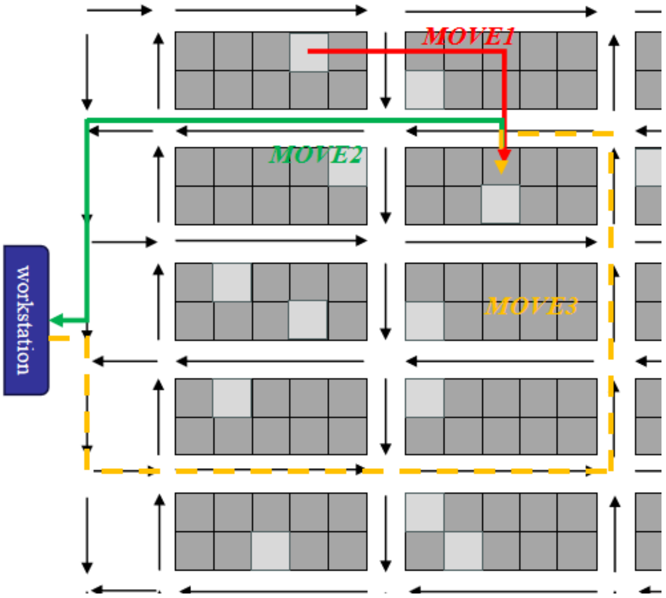


Figure 7. Roadmap for robot movement.

The robot reaches underneath the designated shelf and lifts it up and delivers it to a specific workstation in the direction of the aisle with probability p_w. This is *Move2*.

Due to the fact that multiple robots share the same workstation, the queue in front of the workstation is susceptible to queue-jumping/preemption. After the selecting operation at the workstation has been completed, the robot transports shelves back to the original storage point. Since the robot is loaded for this operation, it must travel along the aisles. This is *Move3*.

After returning the target shelf to its storage location, the robot positions it on the floor and the current order exits the system. The robot remains beneath the current shelf, returns to an inactive state (i.e., it is returned to the external robot queue in the SOQN), and awaits the next order.

Move1, *Move2* and *Move3* are represented as infinite-server (IS) stations (*Station1*, *Station2* and *Station3* respectively) because robots do not need to queue to begin traveling. The network is modelled as SOQN to calculate the waiting time for an order to be matched with a robot.

3.3 Travel Times

3.3.1 Unloaded

The only unloaded motion modelled in this paper is the idle robot's movement from the dwell place to the target shelf after receiving an order (*Move1*). Unloaded traveling robots can pass through the bottom of the shelves, so the distance travelled by an unloaded robot is the Manhattan distance between the robot's dwell place d_r and the target shelf d_s. Consequently, the distance between the robot and the objective shelf is:

$$d_{d_r,d_s} = \left|x_{d_r} - x_{d_s}\right| + \left|y_{d_r} - y_{d_s}\right| \tag{1}$$

In this paper, we employ POSC as the robot's storage strategy, which means that the robot temporarily stores at the location corresponding to the target shelf in the preceding picking process. Then, the first-order moments ES_{d_r,d_s} and second-order moments $ES^2_{d_r,d_s}$ of the robot's time to reach the destination shelf are calculated as follows:

$$ES_{d_r,d_s} = \frac{\sum_{i\in S}\sum_{j\in S} t_{d_r,d_s}}{N_s^2} \tag{2}$$

$$ES^2_{d_r,d_s} = \frac{\sum_{i\in S}\sum_{j\in S} t^2_{d_r,d_s}}{N_s^2} \tag{3}$$

where S is the set of shelves. The travel time from the robot dwell place to the target shelf is expressed as:

$$t_{d_r,d_s} = \frac{\left|x_{d_{s_i}} - x_{d_{s_j}}\right| + \left|y_{d_{s_i}} - y_{d_{s_j}}\right|}{v} \tag{4}$$

3.3.2 Loaded

When loaded, the robots cannot pass through the floor of the shelves and can only move along the aisles (*Move2* and *Move3*). This thesis specifies that in a one-way warehouse, as depicted in Figure 4, the top-to-bottom aisle is alternately transformed from right to left, and the left-to-right cross aisle is alternately transformed from bottom to top. Given the opposite process of sending to and returning from the workstation, the path from the aisle to the workstation is, of course, designed to be bidirectional in this thesis.

Lamballais et al. [7], Kuang et al. [10] and a number of scholars have studied the distance travelled and formal parameters for various cases, which cannot be detailed in this thesis due to page constraints. In this project, the A* algorithm [17] with direction is directly used to solve the formal parameters for the loaded case (see Appendix A).

There are two noteworthy aspects of the modified A* algorithm: 1. if the robot moves to a grid labelled with a direction, it will investigate in that direction rather than all around it (seen as the *findNeighbors* function in Appendix A). 2. If the robot moves until it is level with the one of the workstations, it will stop and cease searching.

Using this algorithm, the distance from the shelf storage point d_s to the workstation d_w can be found. The first-order moments and second-order moments of the traveling time from the target shelf to the workstation can be expressed as follows, respectively:

$$ES_{d_s,d_w} = \frac{\sum_{i\in S}\sum_{j\in W} t_{d_s,d_w}}{N_S \times N_W} \tag{5}$$

$$ES^2_{d_s,d_w} = \frac{\sum_{i\in S}\sum_{j\in W} t^2_{d_s,d_w}}{N_S \times N_W} \tag{6}$$

$$t_{d_s,d_w} = \frac{d_{d_s,d_w}}{v} \tag{7}$$

The previous part discusses moving a loaded robot from a target shelf to a workstation (*Move2*). However, there's a reverse process in the order fulfillment - the robot completes order picking and returns the shelf from the workstation to the storage point (*Move3*). Although routes are unidirectional in the warehouse, the distances for these reverse processes can be assumed equal, resulting in similar time distribution from workstation to shelves:

$$ES_{d_s,d_w} = ES_{d_w,d_s} \tag{8}$$

$$ES^2_{d_s,d_w} = ES^2_{d_w,d_s} \tag{9}$$

Chapter 4 Algorithm Design

4.1 Prerequisite Description

As indicated in Chapter 3, this project's queuing network is modelled to account for multiple job (‘order’ in the warehouse) classes and dedicated resource constraints (exemplified by a single type of resource serving a single type of job). There are service stations designated 1-*M* and job categories numbered 1-*S* on the interior. $\vec{N}=\left(N^{(1)},\ldots N^{(S)}\right)$ represents the quantity of tokens (‘robots’ in the warehouse). Each type of task has its own synchronization station. A job of category *s* arrives at its workstation at rate λ^{s}, and when it departs, its token is returned to the synchronization station. The service rate of each service station is independent of the class of visited jobs, and its first-order and second-order moments are ES_m and ES_m^2, respectively. Assuming that n_I^s denotes the jobs of category *s* in the system, there will be the following restriction, i.e., the number of jobs in each category is less than or equal to the number of tokens [12]:

$$n_I^{(s)} \le N^{(s)} \quad s=1,\ldots S \tag{10}$$

4.2 Algorithm Implementation (AMVA-Pri)

This thesis modifies the conventional AMVA algorithm and proposes the priority-based AMVA algorithm (AMVA-Pri for short), which can be applied to SOQNs containing prioritised queues.

4.2.1 Overall Procedure

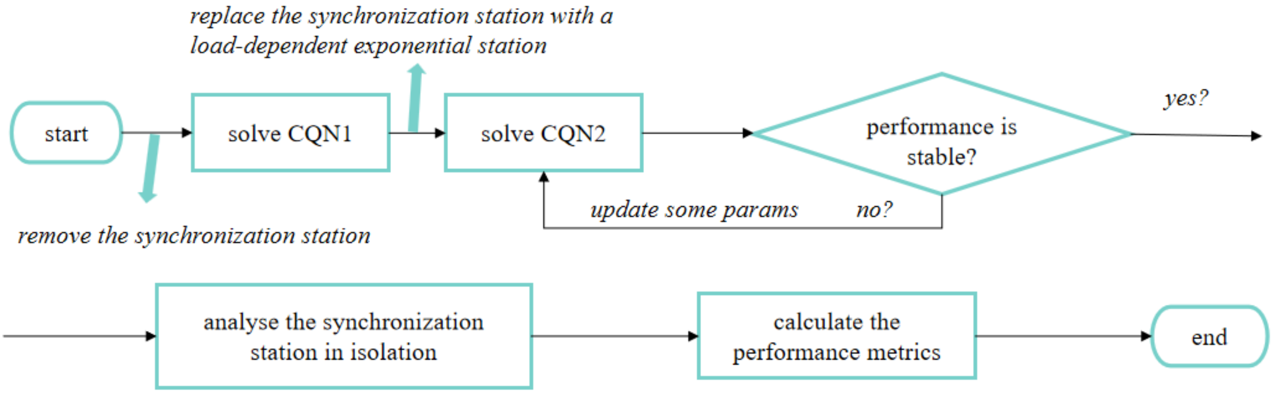

replace the synchronization station with a load-dependent exponential station
start
solve CQN1
solve CQN2
performance is stable?
yes?
update some params
no?
remove the synchronization station
analyse the synchronization station in isolation
calculate the performance metrics
end

Figure 8. The flowchart of AMVA-Pri algorithm

The procedure of the AMVA-Pri algorithm is shown in Figure 8. Several steps are required to solve the SOQN:

❖ Step 1: A CQN named CQN1 is constructed by removing the synchronization stations in SOQN. This CQN is analysed with the AMVA-Pri algorithm (see Appendix B, will be explained later). This step yields TH_{CQN1}^{s}, the throughput of the CQN1 for each class of order.

❖ Step 2: A CQN named CQN2 is created by replacing each synchronization station in the SOQN with a load-dependent exponential station. Assuming that the service rate $\mu_{M+s}\left(r\right)$ of the synchronization station labelled *M*+*s* is λ^{s} when $r>1$, the data obtained by combining CQN1 is also used with the AMVA-Pri algorithm to obtain several performance metrics like utilisation of robots, order cycle time and etc.

- ❖ Step 3: Determine if the performance parameters derived from CQN2 are stable (by comparing TH^{s}_{CQN2} and λ^{s}), if they are not, adjust the load-dependent service rate and repeat step 2 until they are stable.
- ❖ Step 4: The synchronization stations are analysed in isolation to calculate L^{s}_{ext}, the external queue length of class s orders (see Appendix C).
- ❖ Step 5: Combining step 4 and solving the stabilized CQN2 with the AMVA-Pri algorithm yields ρ^{s}, t^{s}_{oc} and ρ_{ws} (see Appendix D).

Two points need to be highlighted here:

1. In the Step 2 of converting synchronization stations to service stations with a negative exponential distribution, let $\mu_{M+s}\left(\vec{n}\right)$ represent the service rate of the *s*th synchronization station when the system has $\vec{n}$ robots. Considering the equilibrium equations of the synchronization station in all circumstances, its service time parameter and order arrival parameter satisfy the following equation:

$$\lambda^{s} = p_{M+s}\left(1\right)\times\mu_{M+s}\left(1\right)+\left(1-p_{M+s}\left(0\right)-p_{M+s}\left(1\right)\right)\times\lambda^{s} \tag{11}$$

where $p_{M+s}\left(r\right)$ is the probability that the synchronization station contains *r* orders. Norton's theorem [18] states that the number of tokens at a synchronization station in a CQN can be expressed as a finite birth/death process with a downcrossing rates equal to the service rate $\mu_{M+s}\left(r\right)$ and an upcrossing rate equal to the internal network's throughput, with the synchronization station being removed when it contains (*N*-*r*) tokens. When the synchronization station has only one token, the service rate will satisfy the following equation:

$$\mu_{M+s}\left(1\right)=\frac{\lambda^{s}}{q^{s}} \tag{12}$$

where q^{s} is given by

$$q^{s}=1-\frac{\lambda^{s}}{TH^{s}_{C_{M+s}}\left(\vec{N}\right)} \tag{13}$$

2. In Step 3, multiple synchronization stations are involved, and the service rates of the various synchronization stations are interdependent, necessitating an iterative phase which is not required by the standard AMVA algorithm [12].

4.2.2 Specific Steps and Details

The majority of the steps are similar to the conventional AMVA algorithm, but a few aspects need to be emphasized.

1. Since multiple orders are evaluated, the algorithm classifies the queue lengths ($L^{s}_{m}\left(\vec{n}\right)$, $\tilde{L}^{s}_{m}\left(\vec{n}\right)$, L^{s}_{ext}) that appear.

2. When calculating the *ET*, which represents the residence time of the robot at each station, it is essential to apply distinct calculation methods based on station categories. For instance, for stations that operate without priority considerations (referred to as IS stations), standard calculations will suffice. In the case of HOL stations, where queue jumping is the only relevant factor, the adjustment in time due to part-based timing needs to be considered. Meanwhile, for PR stations, both queue jumping and preemptive phenomena come into play. Thus, it becomes imperative to factor in corresponding waiting or preemptive times.

a) For HOL stations:

According to Equation (17), it is evident that the residence time of a robot belonging to category *s* in front of the workstation, can be divided into four distinct components. These components include: the remaining time of the robot that is currently being serviced by the workstation upon its arrival in the queue, the residence time of a robot with a priority greater than or equal to *s* that has just entered the queue (which is determined by its length and multiplied by the service time), the residence time caused by a robot with a priority greater than *s* that is inserting itself into the queue while it is already in the queuing process, and finally, the length of the service time required for this particular robot.

The most significant variable in Equation (17) is $\lambda_{mr}^{s}\left(\vec{n}\right)$ which represents the arrival rate of class *r* robots that 'interrupt the residence time' of a class *s* robot [15]. This variable's exact value is hard to calculate, but it can be approximated using the 'Chandy-Lakshmi' (CL) approximation [19] (in conjunction with the linear interpolation approximation), as shown in Equation (21).

b) For PR stations:

Since PR means preemptible, in calculating the impact caused by intermediate arrivals of robots with priority greater than *s*, compared to HOL, it is necessary to account for the time when the robot with priority *s* is being served; at the same time, in the first part of the calculation, i.e. when the robot with priority *s* has just arrived, the remaining service time of the robots in the service station changes, and the waiting time will be added only if the priority of the robot with priority of the robot being served is greater than or equal to *s*. The probability of this is depicted by Equation (20).

3. When analysing synchronization stations separately (see Appendix C), Markov chains [13] are required. In the context of concurrent processing, let $\left(n_0, r_0\right)$ represent the quantity of pending orders and the number of robots awaiting synchronization at the station. According to this mechanism, it can be inferred that either the external number of pending orders or the number of machines is equal to 0. When $r_0 = 0$, the state transition process is shown in Figure 9.

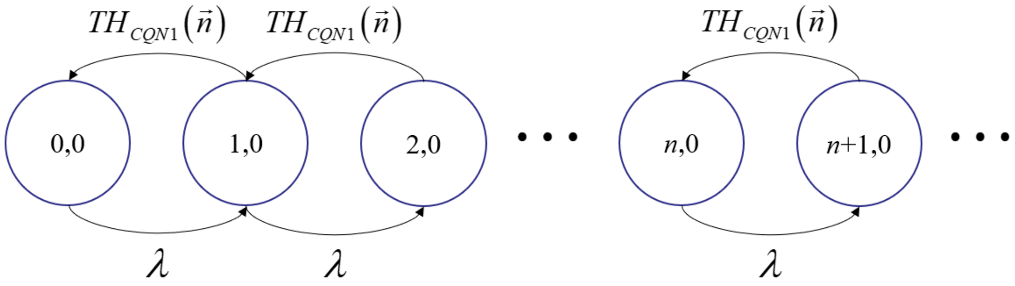

$TH_{CQN1}\left(\vec{n}\right)$
$TH_{CQN1}\left(\vec{n}\right)$
$TH_{CQN1}\left(\vec{n}\right)$
0,0
1,0
2,0
• • •
n,0
n+1,0
• • •
λ
λ
λ

Figure 9. State transition process at the synchronization station

where $TH_{CQN1}\left(\vec{n}\right)$ denotes the throughput, i.e., processing speed, of the internal network when the current population vector is $\vec{n}$. The Markov chain is a birth-and-death process, and its solution yields the external queue length in the stable state which is shown by Equation (32).

Chapter 5 Results and Comparison

5.1 Warehouse Design and Fixed Parameters

In order to validate the analytical model presented in Chapter 3 and the algorithm proposed in Chapter 4, this chapter builds a RMFS simulation system using Anylogic (a simulation software based on Java[2]).

The main design ideas are described in more detail in subsection 3.1.1 and will not be repeated here. In this simulation experiment, the number of workstations is fixed at 3 ($\left[n_{wl}, n_{wb}, n_{wr}\right] = \left[1,1,1\right]$) and distributed in the center of left, bottom and right. Both the aisle width and the cross aisle width are equal to the shelf width *w*. Based on prior research, the other fixed parameters are shown in Table 2.

Table 2. The fixed parameters

w(m)	t_l (s)	t_d (s)	v(m/s)	σ_w^2	S	c_m
1	1	1	1.3	100/12	2	1

5.2 Simulation Validation

5.2.1 Small-scale Warehouse

Assuming that the number of storage blocks distributed horizontally and vertically is 3 by 2, as shown in Figure 10.

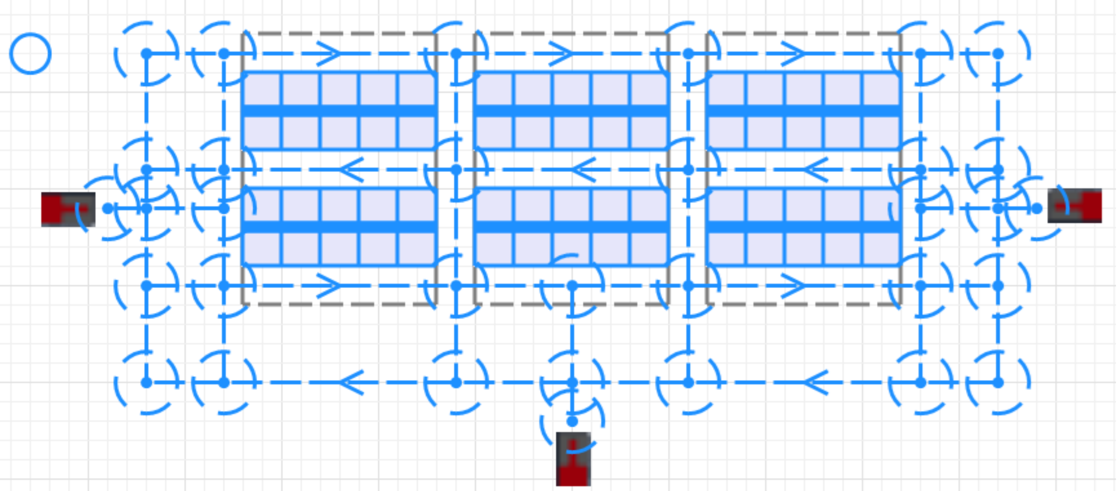


Figure 10. Simulation model of a warehouse (small scale)

In this layout, the results of analytical and simulation models for each scenario are contrasted by varying the arrival rates of high and low priority orders, the number of high and low priority robots, and the average service time at the workstation. The figures in the tables are the average of one hundred simulations in which the network was simulated for one week. The width of the 95 percent confidence intervals is typically less than 1 percent and at most a few percent of the number itself. Three performance metrics are evaluated, including per robot utilisation, per order cycle time and workstation utilisation.

Table 3 and Table 4 show the comparison results for each of the eight cases (the former for the PR workstations and the latter for the HOL workstations). Figure 11-16 illustrate the bar graph results for three of these scenarios for visual comparison.

[2] https://www.anylogic.com/

R_l / R_h means the number of the low/high-priority robots. a_l / a_h means the order arrival rate of the low/high-priority orders. ES_{ws} represents the mean service time of the workstations.

Table 3. Simulation verification of a small-scale warehouse (workstations are PR)

PR						Analytical model					Simulation model				
Case	R_l	R_h	a_l(/h)	a_h	ES_{ws}	ρ_{r_l}(%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws}(%)	ρ_{r_l}(%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws}(%)
1	4	8	90	180	30	94.15	55.59	722.36	95.58	75.01	95.32	51.05	901.94	90.08	75.01
2	5	10	112	224	25	90.72	52.16	391.94	87.64	77.78	91.59	47.37	432.11	81.81	77.82
3	3	6	80	160	30	91.16	60.27	483.78	91.96	66.74	91.72	57.15	472.31	82.19	66.67
4	4	8	120	240	20	84.12	57.26	203.42	73.41	66.72	81.92	54.12	195.49	67.18	66.68
5	4	8	60	120	20	28.80	25.15	72.80	61.24	33.34	28.93	25.32	69.22	59.37	33.42
6	4	8	100	200	20	60.36	45.29	106.44	67.27	55.58	59.22	44.14	99.42	62.71	55.56
7	4	8	70	140	30	54.85	38.53	132.90	81.96	58.34	55.72	36.85	132.64	77.09	58.41
8	6	12	120	240	20	58.45	38.86	117.21	71.07	66.67	58.23	36.14	118.39	66.51	66.84

Table 4. Simulation verification of a small-scale warehouse (workstations are HOL)

HOL						Analytical model					Simulation model				
Case	R_l	R_h	a_l(/h)	a_h	ES_{ws}	ρ_{r_l}(%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws}(%)	ρ_{r_l}(%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws}(%)
1	4	8	90	180	30	77.00	56.36	198.92	97.19	75.05	82.32	55.05	312.24	108.14	74.97
2	5	10	112	224	25	75.16	53.19	170.84	89.52	77.79	78.75	50.82	192.82	84.12	77.83
3	3	6	80	160	30	75.03	61.28	170.23	94.07	66.86	77.89	61.19	194.04	88.29	67.12
4	4	8	120	240	20	71.71	58.94	122.62	75.89	66.76	71.92	56.96	125.50	70.08	67.02
5	4	8	60	120	20	26.43	25.56	66.49	62.27	33.34	26.10	25.24	63.14	60.72	33.41
6	4	8	100	200	20	52.44	46.46	87.53	69.52	55.61	51.83	45.19	82.24	65.68	55.47
7	4	8	70	140	30	45.87	39.10	106.80	83.25	58.36	47.48	39.12	107.16	81.04	58.08
8	6	12	120	240	20	49.79	39.95	96.76	73.11	66.67	50.46	38.24	98.35	69.63	66.75

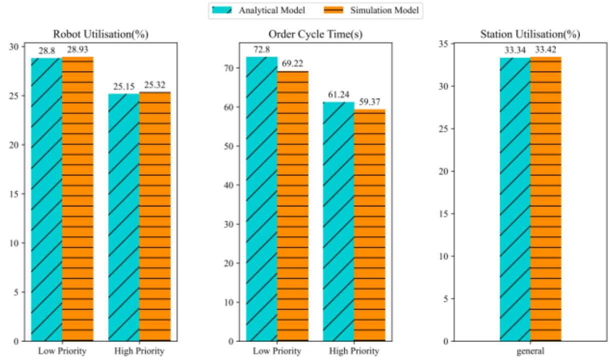


Figure 11. Case 5 and the workstations are PR

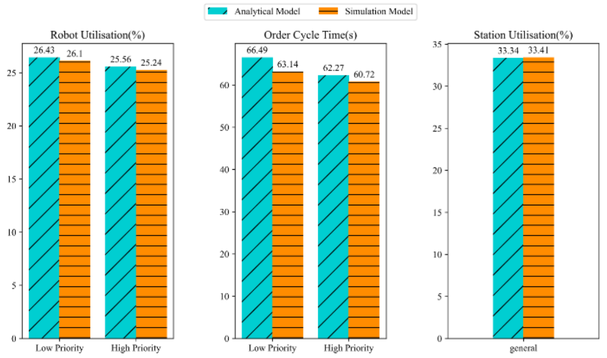


Figure 12. Case 5 and the workstations are HOL

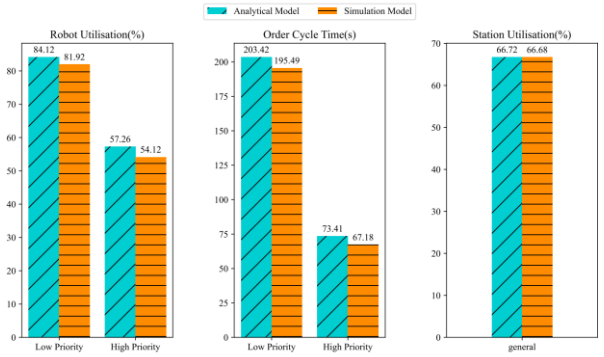


Figure 13. Case 4 and the workstations are PR

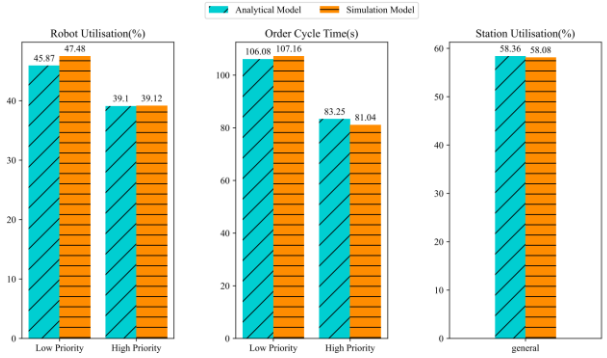


Figure 14. Case 4 and the workstations are HOL

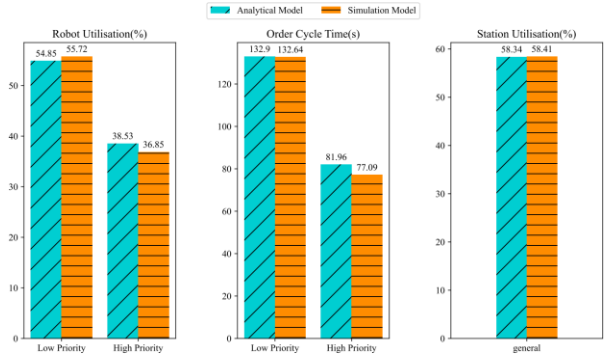


Figure 15. Case 7 and the workstations are PR

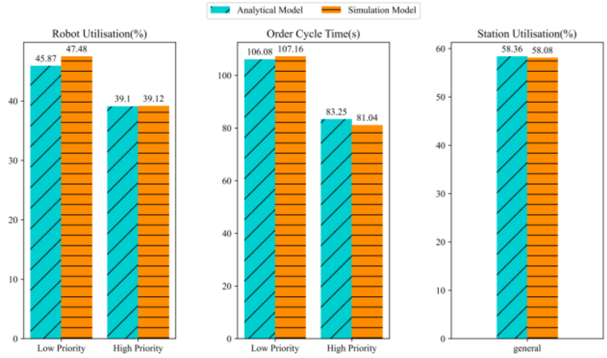


Figure 16. Case 7 and the workstations are HOL

As shown in the preceding tables and figures:

1) Regardless of whether the workstations are PR or HOL, there is little difference between the results produced by the analytical and simulation models, most of which are over 90% accurate, and some of which can reach over 95%. The error consists primarily of the difference between the approximate and actual value. The algorithms invariably contain approximation operations that lead to errors in the length of the queue and waiting time at the station, and consequently to bias in the order cycle time; and this error is amplified as the robot utilisation rate increases. However, these errors are still within acceptable limits.

2) Different categories of workstations have different effects on the performance of the system. Comparing the results for different categories of workstations in the same situation reveals that when the workstations are HOL (i.e., high priority robots are unable to seize low priority robots), the order cycle time for high priority is longer than in the PR mode, leading to an increase in the utilisation of high priority robots, and vice versa for low priority. This is reasonable because, if the workstation is HOL, it means that high priority robots cannot preempt low priority robots that are being serviced, which can result in longer wait periods for them and a commensurate increase in the utilisation.

5.2.2 Large-scale Warehouse

This part increases the warehouse size from 3 by 2 to 9 by 6 to account for scenarios like large-scale warehouses, as depicted in Figure 17.

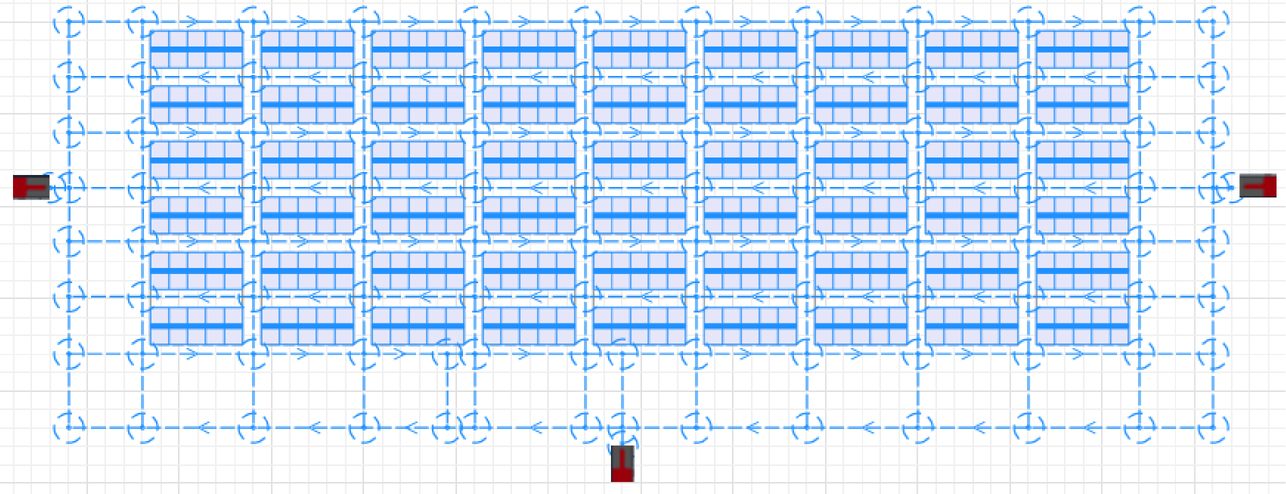


Figure 17. Simulation model of a warehouse (large scale)

Similar to the preceding subsection, the results of the analytical and simulation models are contrasted for various cases. Three performance metrics are evaluated.

Table 5 and Table 6 show the comparison results for each of the eight cases (the former for the PR workstations and the latter for the HOL workstations). Figure 18-23 illustrate the bar graph results for three of these scenarios for visual comparison.

Table 5. Simulation verification of a small-scale warehouse (workstations are PR)

PR						*Analytical model*					*Simulation model*				
Case	*R_l*	*R_h*	*a_l*(/h)	*a_h*	ES_{ws}	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
1	4	8	90	180	25	91.53	71.85	484.03	130.19	62.52	93.41	70.95	471.54	123.26	62.61
2	4	8	80	160	30	93.59	69.68	720.64	135.69	66.69	94.47	67.29	821.34	131.61	66.72
3	4	8	80	160	20	66.46	58.85	154.74	111.72	44.46	66.39	58.32	142.34	107.86	44.51
4	4	8	87	174	20	73.08	63.47	173.17	112.66	48.35	73.82	63.55	164.77	109.66	48.42
5	4	8	55	110	20	41.93	39.03	119.84	104.19	30.56	42.78	39.54	115.06	103.78	30.76
6	5	10	90	180	25	75.09	58.22	205.67	121.24	62.53	76.09	57.38	206.89	116.68	62.39
7	5	10	70	140	20	44.67	40.29	123.17	105.01	38.90	45.24	40.65	120.02	104.47	39.03
8	6	12	80	160	30	64.77	46.89	201.73	129.03	66.67	66.44	45.52	209.87	124.25	66.74

Table 6. Simulation verification of a small-scale warehouse (workstations are HOL)

HOL						*Analytical model*					*Simulation model*				
Case	*R_l*	*R_h*	*a_l*(/h)	*a_h*	ES_{ws}	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
1	4	8	90	180	25	81.17	72.90	231.06	133.11	62.62	82.92	73.72	222.26	129.81	62.62
2	4	8	80	160	30	81.33	70.50	263.35	142.89	66.79	82.72	69.81	267.46	137.71	66.75
3	4	8	80	160	20	61.89	59.66	137.33	113.54	44.50	62.12	59.31	127.98	109.75	44.60
4	4	8	87	174	20	67.52	64.44	146.83	114.81	48.41	68.15	65.07	137.55	112.57	48.51
5	4	8	55	110	20	39.98	39.39	113.52	105.20	30.57	40.45	40.06	109.94	105.11	30.61
6	5	10	90	180	25	66.50	59.00	162.62	123.07	62.57	67.42	59.52	163.18	121.06	62.71
7	5	10	70	140	20	41.96	40.74	114.87	106.24	38.90	42.35	41.34	111.16	106.44	38.83
8	6	12	80	160	30	56.13	47.38	166.92	130.46	66.68	58.65	47.57	175.24	129.26	66.68

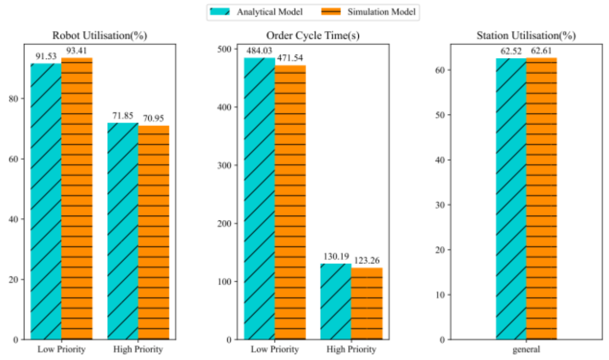


Figure 18. Case 1 and the workstations are PR

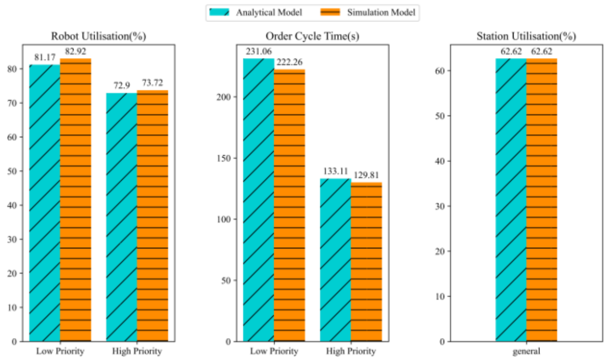


Figure 19. Case 1 and the workstations are HOL

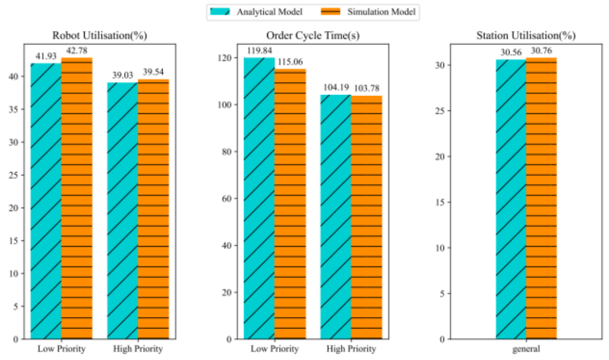


Figure 20. Case 5 and the workstations are PR

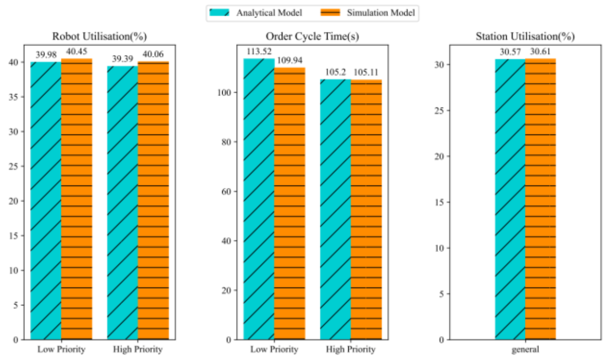


Figure 21. Case 5 and the workstations are HOL

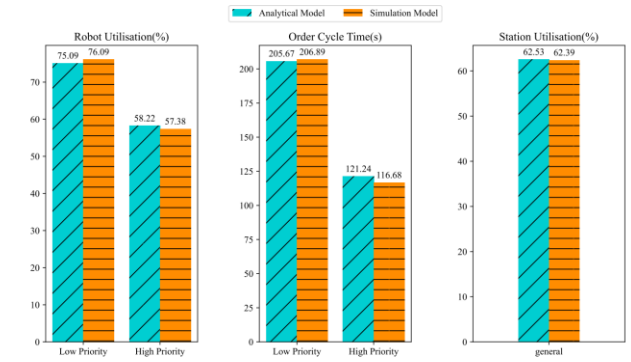


Figure 22. Case 6 and the workstations are PR

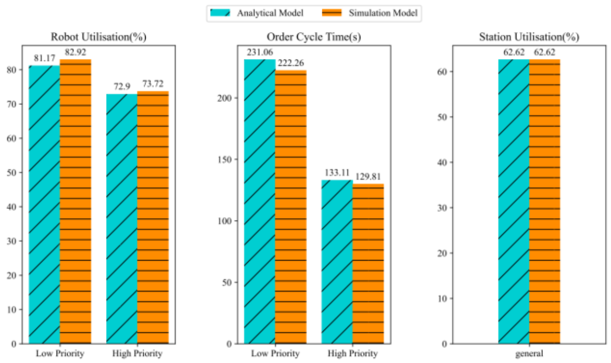


Figure 23. Case 6 and the workstations are HOL

The results of large-scale warehouse experiments are comparable to those of small-scale ones, and the rate of accuracy is generally greater than 90%, with some exceeding 95%. It makes sense that the overall estimation performance of large-scale warehouses is superior to that of small-scale warehouses, given that the order cycle time in large-scale warehouses consists primarily of robot travel times, which have relatively small estimation errors compared with the waiting times in the queues because approximations are hardly used when calculating the travel times, so they are closer to the real values.

5.2.3 Impact of the number of robots

It is intriguing to study the impact of the number of robots, and the designer can control the throughput of the entire system or the throughput of a particular priority level by adjusting the number of robots or the ratio of high priority robots to low priority robots.

This section of the experiment consists of two sub-experiments, the first of which varies the total number of robots (while maintaining the ratio between high and low priority robots), and the second of which varies the ratio between high and low priority robots (while maintaining the number of robots overall). To validate the results, a small-scale warehouse (5.2.1) is chosen for this experiment.

A. Varying the total number of robots

This part of the experiment consists of two cases, and some of the experimental parameters of the first one are shown in Table 7. The results for the first experiments are shown in Table 8 and Table 9.

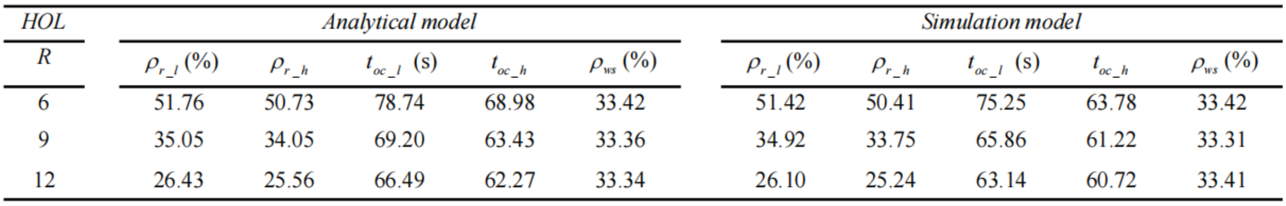
Table 7. Part of parameters for Experiment 1

a_h	*a_l*	ES_{ws}	*Ratio*(*r_h*/*r_l*)
120	60	20	2

Table 8. Experiment 1 (workstations are PR)

PR	*Analytical model*					*Simulation model*				
R	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
6	56.36	49.74	89.44	67.25	33.36	55.77	48.92	85.35	62.67	33.33
9	38.19	33.48	76.34	62.30	33.34	37.77	32.96	72.36	60.12	33.23
12	28.80	25.15	72.80	61.24	33.34	28.93	25.32	69.22	59.37	33.42

Table 9. Experiment 2 (workstations are HOL)



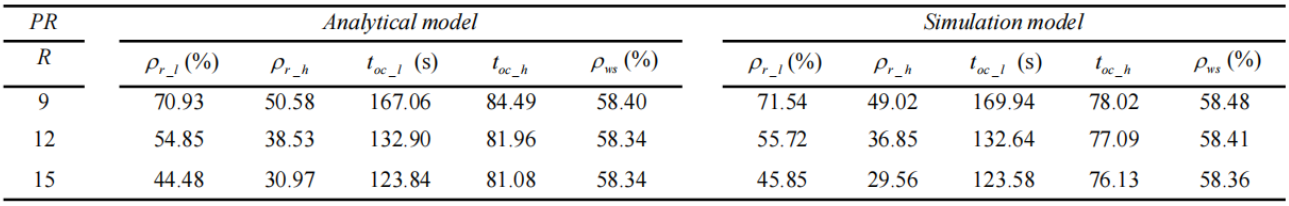
HOL	*Analytical model*					*Simulation model*				
R	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
6	51.76	50.73	78.74	68.98	33.42	51.42	50.41	75.25	63.78	33.42
9	35.05	34.05	69.20	63.43	33.36	34.92	33.75	65.86	61.22	33.31
12	26.43	25.56	66.49	62.27	33.34	26.10	25.24	63.14	60.72	33.41

Some of the experimental parameters of the second one are shown in Table 10. The results for the second experiments are shown in Table 11 and Table 12.

Table 10. Part of parameters for Experiment 2

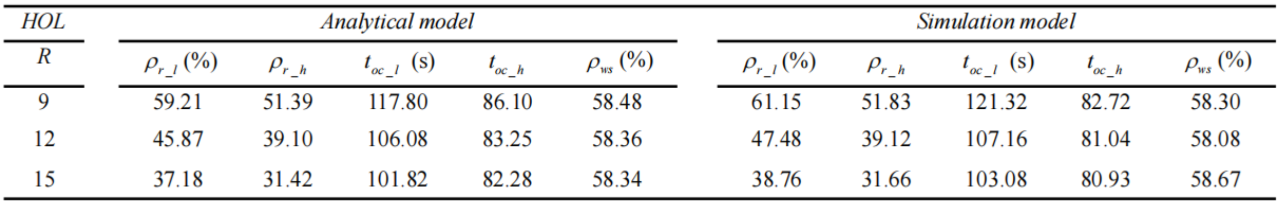
a_h	*a_l*	ES_{ws}	*Ratio*(*r_h*/*r_l*)
140	70	30	2

Table 11. Experiment 2 (workstations are PR)



PR	*Analytical model*					*Simulation model*				
R	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
9	70.93	50.58	167.06	84.49	58.40	71.54	49.02	169.94	78.02	58.48
12	54.85	38.53	132.90	81.96	58.34	55.72	36.85	132.64	77.09	58.41
15	44.48	30.97	123.84	81.08	58.34	45.85	29.56	123.58	76.13	58.36

Table 12. Experiment 2 (workstations are HOL)



HOL	*Analytical model*					*Simulation model*				
R	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
9	59.21	51.39	117.80	86.10	58.48	61.15	51.83	121.32	82.72	58.30
12	45.87	39.10	106.08	83.25	58.36	47.48	39.12	107.16	81.04	58.08
15	37.18	31.42	101.82	82.28	58.34	38.76	31.66	103.08	80.93	58.67

The results indicate that the overall estimation accuracy is close to 95%; it can be seen that as the number of robots increases, the utilisation rate and order cycle time of the robots decreases, but the rate of decline is decreasing. The reason for this result is that as the number of robots increases, the processing capacity of the system will increase, resulting in a shorter external queue length and a correspondingly shorter order cycle time. However, the order arrival rate is relatively low, and the increase in the number of robots per unit of time is greater than the number of orders that are hoarded, so robot utilisation will be reduced. Lastly, the increase in the number of robots will lead to the increase in queuing phenomenon, so the shortening of the order cycle time will occur at a decreasing rate. In practise, businesses must arrive at a relatively optimal solution based on a combination of cost and other factors.

B. Varying the ratio between high and low priority robots

This part of the experiment consists of two cases as well, and some of the experimental parameters of the first one are shown in Table 13. The results for the first experiments are shown in Table 14 and Table 15.

Table 13. Part of parameters for Experiment 1

a_h	*a_l*	ES_{ws}	*R*
120	60	20	12

Table 14. Experiment 1 (workstations are PR)

PR		*Analytical model*					*Simulation model*				
R_l	*R_h*	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
4	8	28.80	25.15	72.80	61.24	33.34	28.93	25.32	69.22	59.37	33.42
5	7	23.07	28.74	71.29	61.53	33.34	23.31	29.22	68.70	59.69	33.27
6	6	19.23	33.51	70.61	62.17	33.36	19.12	32.94	68.56	60.48	33.27

Table 15. Experiment 1 (workstations are HOL)

HOL		*Analytical model*					*Simulation model*				
R_l	*R_h*	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
4	8	26.43	25.56	66.49	62.27	33.34	26.10	25.24	63.14	60.72	33.41
5	7	21.17	29.21	65.30	62.60	33.35	21.29	28.98	62.92	60.78	33.24
6	6	17.65	34.10	64.78	63.39	33.38	17.54	33.88	62.71	60.88	33.68

Some of the experimental parameters of the second one are shown in Table 16. The results for the second experiments are shown in Table 17 and Table 18.

Table 16. Part of parameters for Experiment 2

a_h	*a_l*	ES_{ws}	*R*
240	120	20	18

Table 17. Experiment 2 (workstations are PR)

PR		*Analytical model*					*Simulation model*				
R_l	*R_h*	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
6	12	58.45	38.86	117.21	71.07	66.67	58.23	36.14	118.39	66.51	66.84
7	11	50.34	42.32	111.88	71.29	66.68	50.96	39.24	113.36	67.51	66.79
8	10	45.00	46.45	109.44	71.74	66.71	45.09	43.76	110.88	68.58	66.38

Table 18. Experiment 2 (workstations are HOL)

PR		*Analytical model*					*Simulation model*				
R_l	*R_h*	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)	ρ_{r_l} (%)	ρ_{r_h}	t_{oc_l} (s)	t_{oc_h}	ρ_{ws} (%)
6	12	49.79	39.95	96.76	73.11	66.67	50.46	38.24	98.35	68.13	66.75
7	11	42.85	43.52	93.99	73.38	66.69	41.65	43.65	94.97	68.77	66.80
8	10	37.54	47.80	92.55	73.95	66.74	38.59	45.75	93.92	69.71	66.59

The results indicate that the estimated performance is comparable to that of previous experiments, with an accuracy exceeding 95%.

It is evident that as the proportion of low-priority robots increases, there is a reduction in both the utilisation of low-priority robots and the cycle time for low-priority orders. Conversely, the utilisation of high-priority robots and the cycle time for high-priority orders increase. This observation is logical as the increase in low-priority robots leads to a greater number of low-priority orders being processed promptly. However, the maximum processing speed of the system's internal operations surpasses the rate at which external orders arrive, resulting in a decrease in the utilisation rate of low-priority robots.

Chapter 6 Conclusions and Future work

6.1 Conclusions

Considering the impact of priority factors such as realistic membership mechanisms, this paper brings order priorities into a Robotic Mobile Fulfilment System (RMFS) for the first time. The performance of this system is estimated by using a Semi-Open Queuing Network (SOQN) model and a newly proposed priority-based Approximation Mean Value Analysis (AMVA-Pri) algorithm. Finally, the resulting performance metrics (robot utilisation/order cycle time/workstation utilisation) are compared through simulation experiments to verify the accuracy of the proposed algorithm.

The results indicate that both the analytical model and the newly proposed AMVA-Pri algorithm can effectively predict the performance of RMFS in the presence of order prioritisation, regardless of the size of the warehouse. This estimation assists the clients in selecting a warehouse quickly and aids the warehouse manager in achieving subsequent optimisation.

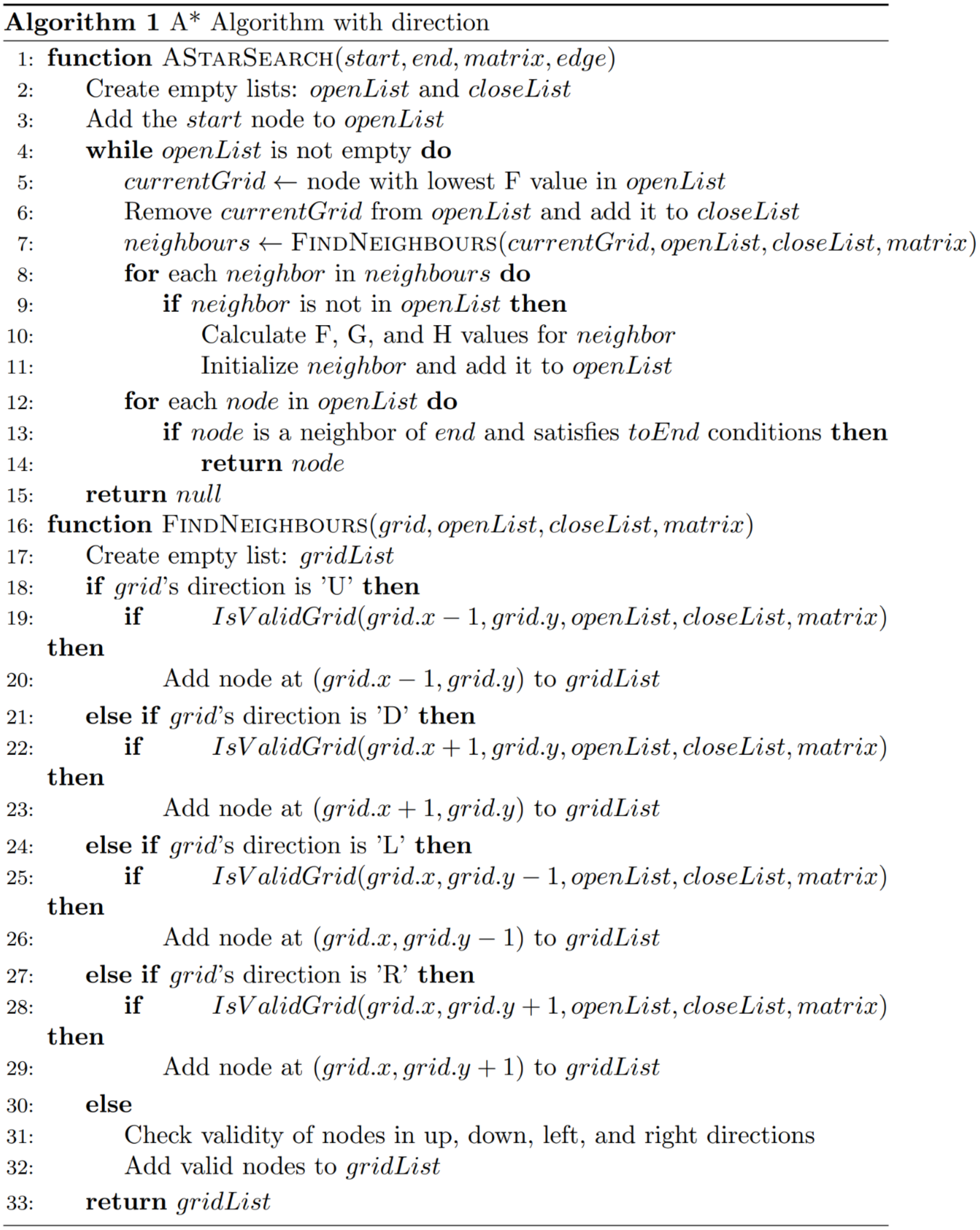
6.2 Future work

This endeavour has many limitations and room for improvement despite the reasonable experimental results.

1. The algorithm may benefit from more optimisation, particularly in scenarios where the robot utilisation is high. In such instances, the accuracy of performance estimation may be compromised due to the system's reliance on numerous approximations. In subsequent analyses, it may be advantageous to turn the approximations into precise numerical values.
2. Some businesses may choose to use only one type of robots to handle different categories of orders for convenience. This is not difficult, and only requires appropriate modifications to the AMVA algorithm based on shared tokens.
3. It would be more realistic to consider that robots with different priorities move at different speeds, e.g. those with a higher priority move faster.
4. To enhance the complexity, it is possible to incorporate a charging process (each robot will run out of power) and a congestion control process (aisles are not unidirectional).
5. Consider realising the project architecture and deploying it online to facilitate the client's and enterprise's ability to meet their requirements.
6. The configuration of the warehouse is not constrained to rectangular shapes but can be expanded to include alternative geometries such as hexagons or fishbone patterns [20], among others.

Appendix

Appendix A. A* algorithm with direction (pseudocode)



Algorithm 1 A* Algorithm with direction

```
function AStarSearch($start, end, matrix, edge$)
    Create empty lists: $openList$ and $closeList$
    Add the $start$ node to $openList$
    while $openList$ is not empty do
        $currentGrid \leftarrow$ node with lowest F value in $openList$
        Remove $currentGrid$ from $openList$ and add it to $closeList$
        $neighbours \leftarrow$ FindNeighbours($currentGrid, openList, closeList, matrix$)
        for each $neighbor$ in $neighbours$ do
            if $neighbor$ is not in $openList$ then
                Calculate F, G, and H values for $neighbor$
                Initialize $neighbor$ and add it to $openList$
        for each $node$ in $openList$ do
            if $node$ is a neighbor of $end$ and satisfies $toEnd$ conditions then
                return $node$
    return $null$
function FindNeighbours($grid, openList, closeList, matrix$)
    Create empty list: $gridList$
    if $grid$'s direction is 'U' then
        if $IsValidGrid(grid.x - 1, grid.y, openList, closeList, matrix)$ then
            Add node at $(grid.x - 1, grid.y)$ to $gridList$
    else if $grid$'s direction is 'D' then
        if $IsValidGrid(grid.x + 1, grid.y, openList, closeList, matrix)$ then
            Add node at $(grid.x + 1, grid.y)$ to $gridList$
    else if $grid$'s direction is 'L' then
        if $IsValidGrid(grid.x, grid.y - 1, openList, closeList, matrix)$ then
            Add node at $(grid.x, grid.y - 1)$ to $gridList$
    else if $grid$'s direction is 'R' then
        if $IsValidGrid(grid.x, grid.y + 1, openList, closeList, matrix)$ then
            Add node at $(grid.x, grid.y + 1)$ to $gridList$
    else
        Check validity of nodes in up, down, left, and right directions
        Add valid nodes to $gridList$
    return $gridList$
```

Appendix B. AMVA-Pri Algorithm Process

Step 1. Initialisation: Set $p_m\left(0|\vec{0}\right)=1$, $Q_m\left(\vec{0}\right)=0$,for $m=1,\dots,M$ and for $s=1,\dots S$ $\tilde{L}_m^s\left(\vec{0}\right)=0$,

$p_{M+s}\left(0|\vec{0}\right)=1$, $Q_{M+s}\left(\vec{0}\right)=0$, $L_w^s\left(\vec{0}\right)=0$ $w=1,\dots,M$ $L_w\left(\vec{0}\right)=0$ $w=M+1,\dots,M+S$

Step 2. Preprocessing: for $m=1,\ldots,M$

$$ES_{rem,m}=\frac{c_m-1}{c_m+1}\frac{ES_m}{c_m}+\frac{2}{c_m+1}\frac{1}{c_m}\frac{ES_m^2}{2ES_m} \tag{14}$$

Step 3. Iteration: for $n=1,\ldots,N$, for all states $\vec{n}$ for which $\sum_{s=1}^{S} n^s = n \text{ and } n^s \le N^s$

a) For $s=S,S-1,\ldots 1$ and $m=1,\ldots,M$

for ***NON-Priority***:

$$ET_m^s\left(\vec{n}\right)=v_m\left[Q_m\left(\vec{n}-\vec{e}_s\right)ES_{rem,m}+\tilde{L}_m^s\left(\vec{n}-\vec{e}_s\right)\frac{ES_m}{c_m}+ES_m\right] \tag{15}$$

for ***HOL***:

$$ET_m^s\left(\vec{n}\right)=v_m\left\{Q_m\left(\vec{n}-\vec{e}_s\right)ES_{rem,m}+\sum_{r\in H_s\cup E_s}\tilde{L}_m^r\left(\vec{n}-\vec{e}_s\right)\frac{ES_m}{c_m}+\sum_{r\in H_s}\lambda_{mr}^s\left(\vec{n}\right)\left[ET_m^s\left(\vec{n}\right)-v_m ES_m\right]\frac{ES_m}{c_m}+ES_m\right\}$$

(16)

$$ET_m^s\left(\vec{n}\right)=\frac{v_m\left[Q_m\left(\vec{n}-\vec{e}_s\right)ES_{rem,m}+\sum_{r\in H_s\cup E_s}\tilde{L}_m^r\left(\vec{n}-\vec{e}_s\right)\frac{ES_m}{c_m}+ES_m-\sum_{r\in H_s}v_m\lambda_{mr}^s\left(\vec{n}\right)\frac{ES_m^2}{c_m}\right]}{1-\sum_{r\in H_s}v_m\lambda_{mr}^s\left(\vec{n}\right)\frac{ES_m}{c_m}} \tag{17}$$

for ***PR***:

$$ET_m^s\left(\vec{n}\right)=v_m\left[Q_m\left(\vec{n}-\vec{e}_s\right)ES_{rem,m}P^s\left(\vec{n}-\vec{e}_s\right)+\sum_{r\in H_s\cup E_s}\tilde{L}_m^r\left(\vec{n}-\vec{e}_s\right)\frac{ES_m}{c_m}+\sum_{r\in H_s}\lambda_{mr}^s\left(\vec{n}\right)ET_m^s\left(\vec{n}\right)\frac{ES_m}{c_m}+ES_m\right]$$

(18)

$$ET_m^s\left(\vec{n}\right)=\frac{v_m\left[Q_m\left(\vec{n}-\vec{e}_s\right)ES_{rem,m}P^s\left(\vec{n}-\vec{e}_s\right)+\sum_{r\in H_s\cup E_s}\tilde{L}_m^r\left(\vec{n}-\vec{e}_s\right)\frac{ES_m}{c_m}+ES_m\right]}{1-\sum_{r\in H_s}v_m\lambda_{mr}^s\left(\vec{n}\right)\frac{ES_m}{c_m}} \tag{19}$$

where

$$P^s\left(\vec{n}-\vec{e}_s\right)\approx\left(\frac{\sum_{r\in H_s\cup E_s}TH^r\left(\vec{n}-\vec{e}_s\right)}{\sum_{r\in H_s\cup E_s\cup L_s}TH^r\left(\vec{n}-\vec{e}_s\right)}\right)^{c_m} \tag{20}$$

$$\lambda_{mr}^s\left(\vec{n}\right)\approx TH^r\left(\vec{n}-L_m^r\left(\vec{n}\right)\vec{e}_r\right)\approx TH^r\left(\vec{n}\right)-L_m^r\left(\vec{n}\right)\left(TH^r\left(\vec{n}\right)-TH^r\left(\vec{n}-\vec{e}_r\right)\right) \tag{21}$$

$$ET_{M+s}\left(\vec{n}\right)=\left[L_{M+s}\left(\vec{n}-\vec{e}_s\right)+Q_{M+s}\left(\vec{n}-\vec{e}_s\right)+p_{M+s}\left(0\,|\,\vec{n}-\vec{e}_s\right)q^s\right]\frac{1}{\lambda^s} \tag{22}$$

$$TH^s\left(\vec{n}\right)=\frac{n^s}{\sum_{m=1}^{M}ET_m^s\left(\vec{n}\right)+ET_{M+s}\left(\vec{n}\right)} \tag{23}$$

$$L_m^s\left(\vec{n}\right)=TH^s\left(\vec{n}\right)ET_m^s\left(\vec{n}\right) \tag{24}$$

$$L_{M+s}\left(\vec{n}\right)=TH^s\left(\vec{n}\right)ET_{M+s}\left(\vec{n}\right) \tag{25}$$

b) for $m=1,\ldots,M$ and for $l=1,\ldots,\min\left(c_m-1,n\right)$

$$p_m\left(l\,|\,\vec{n}\right)=\frac{ES_m}{l}\sum_{s=1}^{S}v_m TH^s\left(\vec{n}\right)p_m\left(l-1\,|\,\vec{n}-\vec{e}_s\right) \tag{26}$$

c) for $m=1,\ldots,M$, if $n<c_m$, $Q_m\left(\vec{n}\right)=0$, otherwise,

$$Q_m(\vec{n}) = \frac{ES_m}{c_m}\sum_{s=1}^{S} v_m TH^s(\vec{n})\left[Q_m(\vec{n}-\vec{e}_s) + p_m(c_m - 1 | \vec{n}-\vec{e}_s)\right] \qquad (27)$$

and for $s = 1,\ldots,S$

$$Q_{M+s}(\vec{n}) = \frac{TH^s(\vec{n})}{\lambda^s}\left[Q_{M+s}(\vec{n}-\vec{e}_s) + p_{M+s}(0 | \vec{n}-\vec{e}_s)q^s\right] \qquad (28)$$

d) for $m = 1,\ldots,M$

$$p_m(0 | \vec{n}) = 1 - \sum_{l=1}^{\min(c_m - 1, n)} p_m(l | \vec{n}) - Q_m(\vec{n}) \qquad (29)$$

and for $s = 1,\ldots,S$

$$p_{M+s}(0 | \vec{n}) = 1 - Q_{M+s}(\vec{n}) \qquad (30)$$

e) for $m = 1,\ldots,M$, if $n < c_m$, $\tilde{L}_m^s(\vec{n}) = 0$, otherwise, for $s = 1,\ldots,S$

$$\tilde{L}_m^s(\vec{n}) = \frac{ES_m}{c_m} v_m TH^s(\vec{n})\left[\tilde{L}_m^s(\vec{n}-\vec{e}_s) + Q_m(\vec{n}-\vec{e}_s)\right] \qquad (31)$$

Appendix C. Calculation of external queuing length

$$L_{ext}^s = p^s(0,0)\frac{\rho^s}{\left(1-\rho^s\right)^2} \quad \rho^s = \frac{\lambda^s}{TH_{CQN1}^s\left(\vec{N}\right)} \qquad (32)$$

$$p^s(0,0) = \prod_{n=1}^{N}\frac{\lambda^s}{TH_{CQN1}^s(\vec{n})}p^s(N,0) \qquad (33)$$

$$p^s(N,0) = \left[\sum_{n_t=\vec{0}}^{\vec{N}}\prod_{n=1}^{\vec{N}-n_t}\frac{\lambda^s}{TH_{CQN1}^s(\vec{n})} + \prod_{n=1}^{N}\frac{\lambda^s}{TH_{CQN1}^s(\vec{n})}\frac{\rho^s}{1-\rho^s}\right]^{-1} \qquad (34)$$

Appendix D. Calculation of some performance metrics

$$\rho^s = 1 - \frac{L_{M+s}}{N_s} \qquad (35)$$

$$t_{oc}^s = \frac{L_{ext}^s + \sum_{m \in M} L_m^s}{\lambda^s} \qquad (36)$$

$$\rho_{ws} = \sum\left(1 - p_{ws}(0 | R)\right)v_{ws} \qquad (37)$$

Reference

[1]. Shah, Bhavin, and Vivek Khanzode. "A comprehensive review of warehouse operational issues." International Journal of Logistics Systems and Management 26.3 (2017): 346-378.

[2]. Jünemann, Reinhardt. Materialfluß und Logistik: Systemtechnische Grundlagen mit Praxisbeispielen. Springer-Verlag, 2019.

[3]. Wurman, Peter R., Raffaello D'Andrea, and Mick Mountz. "Coordinating hundreds of cooperative, autonomous vehicles in warehouses." AI magazine 29.1 (2008): 9-9.

[4]. Roy, Debjit, et al. "Robot-storage zone assignment strategies in mobile fulfillment systems." Transportation Research Part E: Logistics and Transportation Review 122 (2019): 119-142.

[5]. Azadeh, Kaveh, René De Koster, and Debjit Roy. "Robotized and automated warehouse systems: Review and recent developments." Transportation Science 53.4 (2019): 917-945.

[6]. Yuan, Zhe, and Yeming Yale Gong. "Bot-in-time delivery for robotic mobile fulfillment systems." IEEE Transactions on Engineering Management 64.1 (2017): 83-93.

[7]. Lamballais, Tim, Debjit Roy, and M. B. M. De Koster. "Estimating performance in a robotic mobile fulfillment system." European Journal of Operational Research 256.3 (2017): 976-990.

[8]. Zou, Bipan, et al. "Assignment rules in robotic mobile fulfilment systems for online retailers." International Journal of Production Research 55.20 (2017): 6175-6192.

[9]. Duan, Guofang, et al. "Performance evaluation for robotic mobile fulfillment systems with time-varying arrivals." Computers & Industrial Engineering 158 (2021): 107365.

[10]. Kuang. "Evaluating performance in a Robotic Mobile Fulfillment System based on Semi-Open Queuing Network. " MS thesis. Tsinghua University, 2019.

[11]. Jia, Jing, and Sunderesh S. Heragu. "Solving semi-open queuing networks." Operations research 57.2 (2009): 391-401.

[12]. Buitenhek, Ronald, Geert‐Jan van Houtum, and Henk Zijm. "AMVA‐based solution procedures for open queueing networks with population constraints." Annals of Operations Research 93.1-4 (2000): 15-40.

[13]. Bolch, Gunter, et al. Queueing networks and Markov chains: modeling and performance evaluation with computer science applications. John Wiley & Sons, 2006.

[14]. Bryant, Raymond M., et al. "The MVA priority approximation." ACM Transactions on Computer Systems (TOCS) 2.4 (1984): 335-359.

[15]. Eager, Derek L., and John N. Lipscomb. "The AMVA priority approximation." Performance Evaluation 8.3 (1988): 173-193.

[16]. Chen, Qi, et al. "Performance evaluation of compact automated parking systems with mobile application and customer service priority." International Journal of Production Research 59.10 (2021): 2926-2959.

[17]. Erke, Shang, et al. "An improved A-Star based path planning algorithm for autonomous land vehicles." International Journal of Advanced Robotic Systems 17.5 (2020): 1729881420962263.

[18]. Chandy, K. Mani, Ulrich Herzog, and Lin Woo. "Parametric analysis of queuing networks." IBM Journal of Research and Development 19.1 (1975): 36-42.

[19]. Chandy, K. M., and M. S. Lakshmi. "An approximation technique for queueing networks with preemptive priority queues." Department of Computer Sciences, The University of Texas Austin, TX Working Paper. 1984.

[20]. Wang, Yanyan, et al. "Storage assignment optimization for fishbone robotic mobile fulfillment systems." Complex & Intelligent Systems 8.6 (2022): 4587-4602.