OPTIMIZATION OF PRODUCT PLACEMENT AND PICKUP IN AUTOMATED WAREHOUSES

by

Abeer Abdelhadi

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science in Industrial Engineering

School of Industrial Engineering

West Lafayette, Indiana

August 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Prof. Seokcheon Lee, Chair

Department of Industrial Engineering

Prof. Hua Cai

Department of Industrial Engineering

Dr. Patrick Brunese

Department of Industrial Engineering

Approved by:

Dr. Abhijit Deshmukh

Head of the Graduate Program

ACKNOWLEDGMENTS

Firstly, I would like to express my thanks to my patient and supportive advisor, Professor Seokcheon Lee, who has supported me throughout this journey. His guidance has been invaluable. I also wish to thank my committee members, Professor Hua Cai and Doctor Patrick Brunese, for their continuous support.

Further, I wish to thank Abubakr Alabbasi for his insightful suggestions, and for constantly motivating me to work harder. Without him, this thesis would not have been possible. I also wish to thank all the DC Lab members, including Rahul Sucharitha, Zekun Liu, Sungbum Jun, Cansu Agrali, Hoyoung Jeong, Francisco Munoz, Ibrahim Alturki, Juan Camilo Paz Sr and Yonggab Kim, whose hard work inspired me to excel. Each of them never hesitated to help me and constantly provided me with useful comments.

I would also like to thank my family for believing in me and for constantly being there for me through the ups and downs of this journey. I am truly grateful for all what you have done for me and for all the sacrifices you have made. Last but not least, I would like to thank all of my friends for their continuous support. Thank you for listening to me and offering me advice, your encouragement truly helped me excel and push my limits.

TABLE OF CONTENTS

LIST OF TABLES 6

LIST OF FIGURES 7

ABSTRACT 8

CHAPTER 1. INTRODUCTION 9

1.1 Motivation 9

1.2 Our Contribution 10

1.3 Thesis Organization 10

CHAPTER 2. REVIEW OF LITERATURE 11

CHAPTER 3. SYSTEM MODEL AND ASSUMPTIONS 13

3.1 Warehouse Description and Assumptions 13

3.2 Model Description 16

3.3 Retrieval Time 17

CHAPTER 4. PROPOSED PLACEMENT AND SCHEDULING 19

4.1 Probabilistic Scheduling 19

4.2 Products Placement 20

4.3 Latency Tail Probability (LTP) and Mean Latency Characterization 21

CHAPTER 5. OPTIMIZATION FOR MEAN LATENCY AND LTP 24

5.1 Optimization Problem for Mean Latency and LTP Trade-off 24

5.2 Proposed Algorithm 26

5.2.1 Product Assignment Optimization 27

5.2.2 Scheduling Optimization 28

5.2.3 Auxiliary Variables Optimization 29

5.3 Proposed Algorithm Convergence 30

CHAPTER 6. EXTENDED ANALYSIS FOR SCHEDULING 31

CHAPTER 7. NUMERICAL RESULTS 34

7.1 Evaluation of Placement Algorithm 36

7.2 Convergence of Algorithm 38

7.3 Evaluation of Scheduling Algorithm 40

CHAPTER 8. CONCLUSIONS AND FURTURE WORK 44

REFERENCES 45

LIST OF TABLES

3.1 Notations 13
7.1 AGV Speeds 34

LIST OF FIGURES

3.1 A schematic illustrating the warehouse structure, composed of X rows, Y columns and Z shelves. The distance from the loading zone (depot) to the rows is r, while the spacing between the centers of two columns is m. 14

4.1 An illustration of our proposed scheduling policy. Upon arrival of product p, the dispatcher chooses an AGV v with probability $q_{v,p}$. 19

7.1 Effect of arrival rate on average weighted latency 36

7.2 Effect of arrival rate on weighted LTP 36

7.3 Visual representation of the placement of products in the warehouse. A black cell is corresponding to a ”zero” coordinate whereas a white cell is corresponding to a ”one” coordinate. For instance, for $p = 1$, the corresponding location is $(5,2,2)$ for the uniform assignment. However, this product $p = 1$ is stored at location $(1,1,1)$ for the Greedy policy, and in location $(2,5,1)$ for our optimized assignment policy. . . . 37

7.4 Demonstrating convergence of proposed algorithm by plotting δ versus weighted LTP 38

7.5 Demonstrating convergence of proposed algorithm by plotting iteration number versus weighted LTP 39

7.6 Arrival rates versus weighted average latency. Solid lines represent scheduling with the proposed optimal placement of products whereas dotted lines represent scheduling with a uniform placement 40

7.7 δ versus weighted LTP 41

7.8 Weighted average latency for different number of AGVs 42

7.9 Weighted average latency for different arrival rates and classes 43

ABSTRACT

Smart warehouses have become more popular in these days, with Automated Guided Vehicles (AGVs) being used for order pickups. They also allow efficient cost management with optimized storage and retrieval. Moreover, optimization of resources in these warehouses is essential to ensure maximum efficiency.

In this thesis, we consider a three-dimensional smart warehouse system equipped with heterogeneous AGVs (i.e., having different speeds). We propose scheduling and placement policies that jointly consider all the different design parameters including the scheduling decision probabilities and storage assignment locations. In order to provide differentiated service levels, we propose a prioritized probabilistic scheduling and placement policy to minimize a weighted sum of mean latency and latency tail probability (LTP). Towards this goal, we first derive closed-form expressions for the mean latency and LTP. Then, we formulate an optimization problem to jointly optimize a weighted sum of both the mean latency and LTP. The optimization problem is solved efficiently over the scheduling and decision variables. For a given placement of the products, scheduling decisions of customers' orders are solved optimally and derived in closed forms. Evaluation results demonstrate a significant improvement of our policy (up to 32%) as compared to the state of other algorithms, such as the Least Work Left policy and Join the Shortest Queue policy, and other competitive baselines.

CHAPTER 1. INTRODUCTION

1.1 Motivation

Warehouses are a crucial part of modern supply chains. They greatly influence the success or failure of businesses [1]. While many companies attempted direct shipping to customers, there were plenty of circumstances where it was not applicable, due to failure to reduce supplier lead times to the levels desired by customers in a cost effective way. Thus these customers need to be served from an inventory rather than to order [2]. Warehouses are also crucial from a cost perspective, representing about 22% of the logistics costs in the USA and about 25% in Europe [3].

Further, the demand of customer orders has witnessed a tremendous growth in the last decade. With this increasing demand, price-based differentiated services and timely delivery of products are important. Further, order picking (i.e., time required to serve an incoming request of a customer) accounts for up to 55% of overall operating costs for a warehouse [4]. This increases the pressure on companies and enterprises (e.g., Amazon and Walmart) to establish a more efficient and flexible order picking system in the competitive market to gain more revenues and achieve customer satisfaction. Smart warehouse automation represents an efficient and competitive solution for suppliers and providers. Further, one of the performance measures investigated in this thesis is the Latency Tail Probability (LTP) of customer orders of products, which is defined as the probability that the latency is greater than a certain threshold. We note that the LTP has been shown to affect the customers' experience more than the mean, and that motivates us to consider the tail metric.

1.2 Our Contribution

In this thesis, we propose a novel, yet efficient, policy for optimizing the placement of the products and scheduling the customers' orders at the Automated Guided Vehicles (AGVs). In contrast to many queue-based scheduling techniques such as Join the Shortest Queue (JSQ), Power-of-d (Pow(d)), and Least Work Left (LWL), where the instantaneous queue length is continuously tracked, our scheduling policy is independent of the instantaneous queue level and employs only the average queue length in its decision, and thus is less complex. Further, our policy can differentiate and provide different priority levels for the customers' orders so that customers in higher classes can be prioritized over others. We aim to jointly optimize the storage assignment, minimize the average processing time of customer's orders in the smart 3D warehouse automation (i.e., mean latency), and reduce the latency tail probability (LTP) for the orders of the customers. To the best of our knowledge, this work is the first to consider a joint optimization of product placement, mean latency and LTP for customer orders for a 3-dimensional warehouse system.

1.3 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 gives the related work and states the key differences between our work and the most related work in the literature. Chapter 3 presents the system model for the problem and the key assumptions of the considered smart warehouse system. Chapter 4 finds an upper bound on tail latency through probabilistic scheduling and derives the mean latency expressions. Chapter 5 formulates the joint optimization problem for the mean latency and LTP and presents our proposed algorithm. Chapter 6 shows extended analysis for scheduling. Chapter 7 presents our numerical results, and chapters 8 concludes the thesis with the conclusions and references.

CHAPTER 2. REVIEW OF LITERATURE

There are three categories of research topics related to our work: placement optimization, product scheduling techniques and minimizing mean latency and latency tail probability in smart warehouse systems.

Placement Optimization: Recently, placement optimization has received great attention to study the performance of the rapid deployment of smart warehouse systems. Some of these placement policies include class-based storage where higher priority products are placed closer to the loading zone (e.g. [5], [6]). Another policy is the greedy approach, also known as the full-turnover storage (e.g. [7], [8]), where products with the higher arrival rates are stored closest to the loading zone. Furthermore, the product affinity-based storage tackles the pairwise relationships between the products [9]. However, the above approaches are heuristics and consider only one feature when optimizing the products placement. In our work, we propose an optimal placement that jointly considers more than one aspect for product assignments, e.g., rate of product requests and product classification.

Scheduling Techniques for Products: Different approaches have been proposed to schedule products (or tasks in general) on different AGVs (or workers/servers). Some examples of these approaches are V-choose-2, or Power-of-d (Pow(d)): d servers are randomly selected and then the request is sent to the shortest AGV queue (e.g. [10]). Similar approaches like Join the Shortest Queue (JSQ) [11] and Least-Work-Left (LWL) [12] have also been proposed in the literature. However, these policies do not distinguish between the products and need to continuously track the queue length of each individual AGV which increases the complexity of AGV selection. Further, different from previous methods, our policy gives more priority to the products with higher weights (e.g., higher arrival rates) in order to optimize the overall performance.

Latency in Warehouse Systems: Various modeling and scheduling techniques have been proposed for quantifying tail and mean latency in warehouse systems considering different settings. Such techniques include fuzzy collaborative intelligence-based algorithm (e.g. [13]) which was motivated by the collaborative control theory. Also, an approach that considers the storage assignment and the travel distance was investigated [14]. Moreover, k-means batching was tackled [15]. Further, an approach that considers object-oriented dynamic modeling where both production planning and inventory replenishment systems are modeled was studied [16]. Some papers tackled the tail latency for demands with high uncertainty [17]. However, in most of these previous works, retrieval times for product pickup are assumed to be either identical or deterministic, which might not be the case in reality. Moreover, joint optimization of both latency tail probability and mean latency is not considered. Our framework considers both metrics when scheduling and placing products to further improve the smart warehouse systems.

CHAPTER 3. SYSTEM MODEL AND ASSUMPTIONS

3.1 Warehouse Description and Assumptions

Below are the basic notations used for our model along with their interpretations. All variables listed are input variables, except for the last three. Those three variables are the decision variables, which we aim to optimize later.

Table 3.1. *Notations*

Notation	Name
p (Cardinality:P)	Product types index
v (Cardinality:V)	AGV index
x (Cardinality:X)	Rows index
y (Cardinality:Y)	Columns index
z (Cardinality:Z)	Shelves index
r	Fixed distance from loading zone to rows
m	Fixed distance between two adjacent columns
n	Fixed distance between two adjacent shelves
λ_p	Arrival rate of product of type p
θ	Tuning parameter/trade-off factor for mean and tail latency
ω_p	Weight of order of product p
ρ_v	Utilization/load intensity of AGV v
$\alpha_{v,p}$	Fixed/minimum time to retrieve product p via AGV v
μ_v	Service rate at an AGV v
$S_{v,p}$	Retrieval time for product p at an AGV v
W_v	Waiting time at the queue of AGV v
$L_{v,p}$	Latency for product p if assigned to AGV v
S_v	Retrieval time at an AGV v
ID_p	ID of product p
$q_{v,p}$	Probability of product p being assigned to an AGV v (decision variable)
$\mathscr{S}_p$	Placement of Products (decision variable)
t_v	Auxiliary variable of the MGF (decision variable)

We consider a 3D automated warehouse system composed of shelves, rows, and columns as depicted in Figure 1. Each row and column coordinates of each shelf can only store one type of product. Further, any product p of a certain type can only be stored in a given location determined by the (x_p, y_p, z_p) coordinates. In our model, we consider differentiated classes or levels of service for customer orders. These differentiated classes are assumed to be price-based and depend on the agreement between the customer and the service provider(s), e.g., Amazon or Walmart. The arrival of product orders is assumed to follow a Poisson distribution where product p is requested with rate λ_p. The choice of Poisson process is common and widely used to model random requests of online retailers (e.g. [18]).

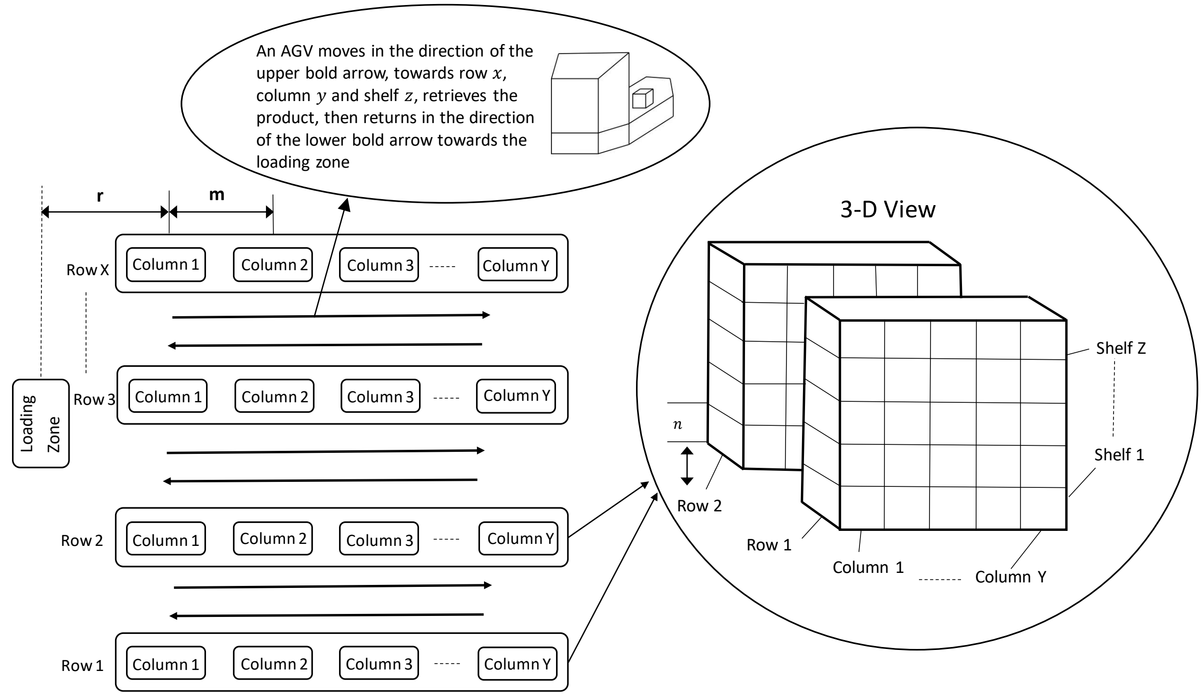
An AGV moves in the direction of the upper bold arrow, towards row x, column y and shelf z, retrieves the product, then returns in the direction of the lower bold arrow towards the loading zone
r
m
Row X
Column 1
Column 2
Column 3
Column Y
Row 3
Row 2
Row 1
Loading Zone
3-D View
Shelf Z
Shelf 1
n
Row 2
Row 1
Column 1
Column Y

Figure 3.1. A schematic illustrating the warehouse structure, composed of X rows, Y columns and Z shelves. The distance from the loading zone (depot) to the rows is *r*, while the spacing between the centers of two columns is *m*.

As depicted in Figure 3.1, the warehouse system modelled has a total number of X rows, Y columns and Z shelves. The bold arrows represent the route of travel of the AGVs. Further, *r* is the fixed distance from the loading zone to the first column whereas *m* is the distance between each two columns. Such layout is widely used among major retail stores, e.g. Walmart and Target, and have been used in previous papers that tackled warehouse systems (e.g. [19], [20], [21]).

While this thesis assumes similar structures of warehouse systems, our framework and analysis remain applicable to a wide range of other systems, such as those in [22] and [23]. As part of smart warehouse automation system, pick-and-pack operations are performed using robotics, i.e., Automated Guided Vehicles (AGVs). Hence, upon an arrival of a product request, an AGV is assigned to serve this request. In Chapter 4, we explain our proposed scheduling policy and show how this policy is optimized to reduce the pickup and retrieval times of the incoming online orders. Each AGV has a different horizontal speed, C_v for AGV v, whereas the speed of the AGV arm is assumed to be a constant (denoted by f) for all v. We aim to optimize the following decisions:

- The storage assignment of the products on the rows, columns and shelves
- The dispatching of the incoming product orders to the AGVs

Without loss of generality, we further consider the following assumptions:

- The horizontal and vertical distances between the rows and shelves are constant
- The distance from the loading zone to the nearest row (seen in Fig 3.1 as r) is constant
- A strict-order picking policy is considered, where each AGV can handle only one order of a customer at a time
- There is always an availability of the inventory and the time to replenish the warehouse system is negligible
- The arrival rates of incoming online orders $\lambda_p, \forall p$, are assumed to be given or predicted from historical data
- There is a limited number of AGVs and this constitutes the bottleneck operation
- A dedicated storage policy is considered, where each shelf can hold only a single type of products
- The pick-up/deposit (P/D) time for the AGV to pick up or deposit the product can be ignored, which is justified if the P/D time is fairly small compared to the total latency

- Aisles are wide enough to allow travel of multiple AGVs
- Unlike the traveled distance in y and z directions, the traveled distance in the x-direction is not varying from one product to another and is assumed to be fixed and equal to r (as depicted by Figure (3.1)).

3.2 Model Description

In this section, we describe our proposed model. We assume that there are P product types. The inter-request time (time between two consecutive requests), for every product p, is exponentially distributed with rate λ_p. Our objective is to dispatch the requests for each product in such a way that a weighted sum of mean latency and LTP is minimized. The requests can be assigned to any AGV v, where $v \in \{1, 2, \cdots, V\}$, for service. Further, for the FCFS scenarios, the service for a product is assumed to be non-preemptive so AGVs cannot be interrupted if they are already in service. In order to serve a request of product p, we need to choose one AGV, v, to serve the request. Each AGV has a different horizontal speed, denoted by C_v for AGV v. Moreover, the arm of each AGV has a constant speed, represented by f. To provide prioritized service levels, we propose a prioritized probabilistic scheduling and placement policy as follows. Each order is assigned to one AGV with probability $q_{v,p} \geq 0$ for AGV v. By optimizing $q_{v,p}$, $\forall$ v, p, the load is balanced among all AGVs. For any product p, the following condition has to be satisfied for feasibility of scheduling process, by ensuring that each product can only be assigned to one AGV:

$$\sum_{v=1}^{V} q_{v,p} = 1 \qquad \forall p \tag{3.1}$$

The selection process of AGV v is a challenging task as it needs to take into consideration many factors including the queue of each AGV as well as the current orders that are not fully executed yet. Besides those factors, the policy should also efficiently schedule the products such that the mean latency and/or tail latency of products are minimized. In chapter 4, we explain our proposed scheduling policy and show how this policy is optimized to reduce the latency of the products.

3.3 Retrieval Time

We assume that the retrieval time of an order of a product p at an AGV v follows a shifted exponential distribution with two parameters $(\mu_v, \alpha_{v,p})$ [24]. This distribution of the retrieval time $X_{v,p}(s)$ is given by the following equation:

$$X_{v,p}(s) = \begin{cases} \mu_v e^{-\mu_v(s-\alpha_{v,p})} & s \geq \alpha_{v,p} \\ 0 & s < \alpha_{v,p} \end{cases} \tag{3.2}$$

where the parameter of the distribution represents $S_{v,p}$, which is the retrieval time for product p of an AGV v (denoted by s in the equation for simplicity), μ_v represents the service rate at an AGV v and $\alpha_{v,p}$ gives the fixed minimum time needed to retrieve product p using AGV v. Note that the expected retrieval time of an order of product p is $\alpha_{v,p} + 1/\mu_v$.

We note that the value of $\alpha_{v,p}$ depends on the assignment of product p. Thus, the retrieval time depends on how close or far the product is and, hence, it scales differently according to the product location in the warehouse. Hence, two components contribute to the retrieval time of product p: random part and constant part. The constant part represents the *minimum* time of a request for product p to be successfully received. Further, the exponential part accounts for the variability and reliability of the AGVs and warehouse environment, and captures the randomness that makes the retrieval time non-deterministic. Unlike the service model, which is widely considered in the literature, our shifted exponential model gives more flexibility for better modeling the service in reality[1]. Let $M_{v,p}(t_v) = \mathbb{E}[e^{t_v S_{v,p}}]$ be the moment generating function of the retrieval time of an order for product p at AGV v, $S_{v,p}$. Then, $M_{v,p}(t_v) = \mathbb{E}[e^{t_v S_{v,p}}]$ is given as

$$M_{v,p}(t_v) = \frac{\mu_v}{\mu_v - t_v} e^{\alpha_{v,p} t_v} \qquad \forall v, p \tag{3.3}$$

[1] In fact, we choose a shifted distribution in order to simulate general distributions. The shifted exponential distribution is a two-parameter distribution. When the shift/fixed parameter is much larger than the random part of the retrieval time $(1/\mu_v)$, it can approximate the deterministic models. In contrast, when the shift parameter is much smaller than $(1/\mu_v)$, it approximates the exponential distribution. Hence, ”SExp” distribution includes the exponential and deterministic/general distributions as special cases.

It remains to characterize the minimum retrieval time $\alpha_{v,p}, \forall v, p$. From Figure 1, the horizontal distance at column y, denoted as D_y, traveled by an AGV v is given by

$$D_y = 2[r + m(y-1)] \tag{3.4}$$

where y represents the row index (or width coordinate) and m represents the distance between two adjacent rows, and L represents the fixed distance from the loading zone/depot to the first row.

Next, we calculate the vertical distance D_z traveled by the arm of an AGV v. Similar to D_y, we can write

$$D_z = 2[n(z-1)] \tag{3.5}$$

where z gives the elevation of the shelf (or height coordinate), and n represents the vertical distance between the shelves.

From (3.4) and (3.5), we can write the minimum time needed to retrieve product p using AGV v as follows

$$\alpha_{v,p} = \frac{D_y}{C_v} + \frac{D_z}{f} \tag{3.6}$$

Recall that f is the vertical speed of the AGV arm, and C_v is the horizontal speed of an AGV v. Next, we present our proposed strategies for product placement and AGV scheduling.

CHAPTER 4. PROPOSED PLACEMENT AND SCHEDULING

In this section, our proposed strategies for placement and scheduling the product requests are presented.

4.1 Probabilistic Scheduling

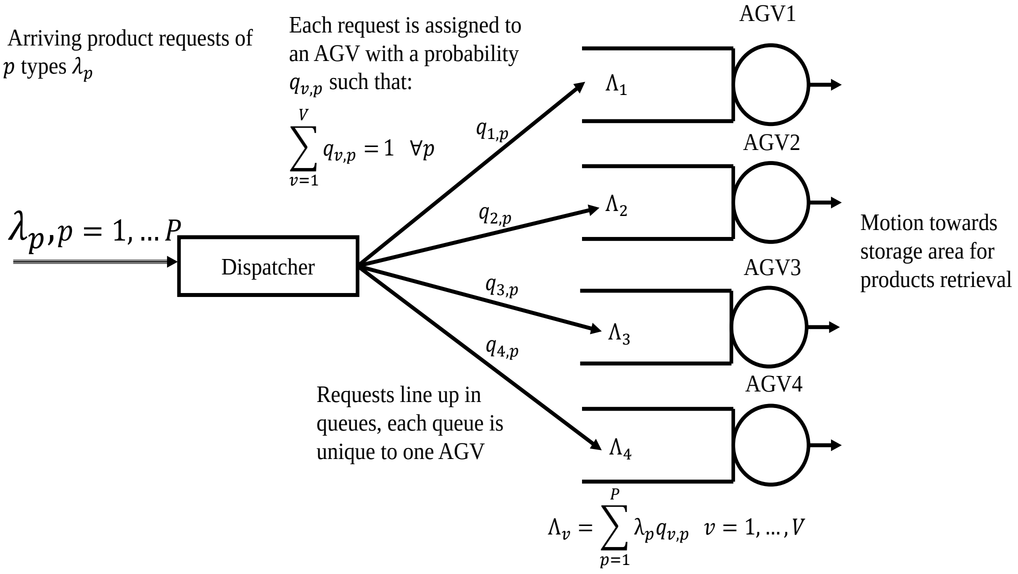
Arriving product requests of p types λ_p
Each request is assigned to an AGV with a probability $q_{v,p}$ such that:
$\sum_{v=1}^{V} q_{v,p} = 1 \quad \forall p$
$\lambda_p, p = 1, \dots P$
Dispatcher
$q_{1,p}$
$q_{2,p}$
$q_{3,p}$
$q_{4,p}$
AGV1
AGV2
AGV3
AGV4
Λ_1
Λ_2
Λ_3
Λ_4
Motion towards storage area for products retrieval
Requests line up in queues, each queue is unique to one AGV
$\Lambda_v = \sum_{p=1}^{P} \lambda_p q_{v,p} \quad v = 1, \dots, V$

Figure 4.1. An illustration of our proposed scheduling policy. Upon arrival of product p, the dispatcher chooses an AGV v with probability $q_{v,p}$.

We now describe our proposed scheduling policy. Upon arrival of orders at the dispatcher (see Figure 4.1), an AGV v is chosen to serve the order. The optimal scheduling policy has to consider the queue state of all AGVs, importance of each order, and all orders that are not fully executed yet. While one can use a Markov decision process with multiple states, this approach is not tractable and will result in, so-called, state explosion problem [25]. Further, this approach does not give expressions that can be optimized to determine the real-time orders assignments and optimal AGVs allocations.

To overcome these issues, we propose a scheduling that jointly considers all different design parameters including the scheduling decisions ($q_{v,p}, \forall v, p$) and the heterogeneity of the AGVs. To provide prioritized service levels, we propose a prioritized probabilistic scheduling as follows. Each AGV v has its own queue and the real-time updates in each queue are served under First Come First Serve. An order of product p, $p \in \{1,2,3,\dots,P\}$, is assigned to the queue of an AGV v, $v \in \{1,2,3,\dots,V\}$, with probability $q_{v,p} \geq 0$.

In order to retrieve a product p, we first probabilistically choose one AGV to pick-up the product. Since the key bottleneck is the limited number of AGVs, orders have to wait in a queue until the AGV is free. Under probabilistic scheduling, the arrival of orders at AGV v forms a Poisson process with rate:

$$\Lambda_v = \sum_{p=1}^{P} q_{v,p}\lambda_p, \qquad \forall v \tag{4.1}$$

which is the superposition of P Poisson processes each with rate $q_{v,p}\lambda_p$. In chapter 6, we carry out, under some simplified assumptions, an analysis to provide closed-form expressions for the scheduling probabilities to gain some insight into the performance of the proposed algorithm and its behavior under different system parameters.

4.2 Products Placement

Let $a_{x,y,z,p}$ be an indicator variable which is equal to 1 if product p is stored at row x, column y and shelf z, and zero otherwise, i.e.,

$$a_{x,y,z,p} = \begin{cases} 1 & \text{if product } p \text{ is stored at row } x\text{, column } y \text{ and shelf } z \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

Since any given location can only store one product and every product is stored at one place only, the following conditions hold true

$$\sum_{x=1}^{X}\sum_{y=1}^{Y}\sum_{z=1}^{Z} a_{x,y,z,p} = 1 \qquad \forall p \tag{4.3}$$

$$\sum_{p=1}^{P} a_{x,y,z,p} = 1 \qquad \forall x, y, z \tag{4.4}$$

From the previous equations, and recalling that $\alpha_{v,p} + \frac{1}{\mu_v}$ is the expected retrieval time of the shifted exponential distribution, the retrieval time of a product p, given that it is assigned to AGV v, is then given by

$$\mathbb{E}(S_{v,p}) = \sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{z=1}^{Z} a_{x,y,z,p} \left(\alpha_{v,p} + \frac{1}{\mu_v} \right) \qquad \forall v, p \tag{4.5}$$

We note that the retrieval time at an AGV v, S_v, is $S_{v,p}$ with probability $\frac{\lambda_p q_{v,p}}{\Lambda_v}$. Hence, the retrieval time at AGV v is given by averaging over all product types using the following equation

$$\mathbb{E}(S_v) = \sum_{p=1}^{P} \frac{\lambda_p q_{v,p}}{\Lambda_v} S_{v,p} \qquad \forall v \tag{4.6}$$

4.3 Latency Tail Probability (LTP) and Mean Latency Characterization

In this section, we quantify the LTP and the average latency of an order for product p, given that it is assigned to AGV v. Let $L_{v,p}$ be the random sojourn time that product p needs if assigned to AGV v. The latency tail probability of an order for product p is defined as the probability that $L_{v,p}$ is greater than or equal to δ_p, for a given δ_p and AGV v. Since evaluating $\Pr(L_{v,p} \geq \delta_p)$ in closed form is challenging for heterogeneous settings with general service time distribution, we derive an upper bound on LTP. This upper bound turns out to be tight as will be shown later.

The total time for an order to be completed (total service time) depends on two components, (i) waiting in the queue of AGV v for service, W_v, and (ii) retrieval time for an order p, $S_{v,p}$, at AGV v. The latency $L_{v,p}$ for an order of product p, served from AGV v is thus given as

$$L_{v,p} = W_v + S_{v,p}. \tag{4.7}$$

Note that the waiting time in the queue of AGV v is the same for all products due to the nature of M/G/1 queue (all products are queued in one queue), however, the retrieval time depends on the product. Hence, the proposed model differentiates between the products according to their type, location and priority.

From Eqn. (3.3) and similar to that in (4.6), we can derive the moment generating function of the retrieval time at an AGV v by averaging over all product types as follows, i.e.,

$$M_v(t_v) = \sum_{p=1}^{P} \frac{\lambda_p q_{v,p}}{\Lambda_v} M_{v,p}(t_v) \qquad \forall v \tag{4.8}$$

Since $\mathbb{E}\left[S_v\right] = M_v^{'}(0)$, by taking the first derivative of the above equation and equating to zero, we arrive at the expected retrieval time at AGV v as follows

$$\mathbb{E}\left[S_v\right] = \sum_{p=1}^{P} \frac{\lambda_p q_{v,p}}{\Lambda_v} \left(\alpha_{v,p} + \frac{1}{\mu_v}\right) \qquad \forall v \tag{4.9}$$

Further, the utilization of AGV v (denoted by ρ_v) is given as follows

$$\rho_v = \sum_{p=1}^{P} \lambda_p q_{v,p} \left(\alpha_{v,p} + \frac{1}{\mu_v}\right) \qquad \forall v \tag{4.10}$$

Having characterized the service time distribution, the moment generating function of the latency $L_{v,p}$ can be characterized using Pollaczek-Khinchine (PK) formula for M/G/1 queues, since the request pattern is Poisson and the service time has a general distribution. This PK formula gives us the MGF of the latency. Hence, we can write

$$\mathbb{E}[e^{t_v L_{v,p}}] = \frac{(1-\rho_v) t_v M_{v,p}(t_v)}{t_v - \Lambda_v (M_v(t_v) - 1)} \qquad \forall v, p \tag{4.11}$$

Since (4.11) is a non-negative, monotonically increasing function, we can apply Markov's inequality to give us an upper bound for the LTP, where the numerator in the R.H.S. represents the PK formula, as follows:

$$\mathbb{P}(L_{v,p} \geq \delta) \leq \frac{\mathbb{E}[e^{t_v L_{v,p}}]}{e^{t_v \delta}} \qquad \forall v, p \tag{4.12}$$

By averaging over the choice of AGVs, we arrive at an expression for the LTP of product p as follows

$$\mathbb{P}(L_p \geq \delta) = \sum_{v=1}^{V} q_{v,p} \mathbb{P}(L_{v,p} \geq \delta) \qquad \forall p \tag{4.13}$$

Next, we derive the average latency of product p. Since $\mathbb{E}\left[L_{v,p}\right] = \frac{d(\mathbb{E}[e^{t_v L_{v,p}}])}{d_{t_v}}|_{t_v=0}$ using equation 4.11, the expected average latency at AGV v for product p is given by taking the first derivative of the PK formula and equating to zero, as follows:

$$\mathbb{E}[L_{v,p}] = \frac{\Lambda_v \mathbb{E}[S_v^2]}{2(1-\Lambda_v \mathbb{E}[S_v])} + \mathbb{E}[S_{v,p}] \qquad \forall v, p \tag{4.14}$$

where $\mathbb{E}[S_v^2]$ is the second moment of the retrieval time and is calculated by averaging over all product types as follows

$$\mathbb{E}[S_v^2] = \frac{\sum_{p=1}^{P} \lambda_p q_{v,p} S_{v,p}^2}{\Lambda_v} \qquad \forall v \tag{4.15}$$

Finally, the average latency of product p is calculated by averaging $L_{v,p}$ calculated in Equation (4.14) over all AGV choices, as follows:

$$\mathbb{E}[L_p] = \sum_{v=1}^{V} q_{v,p} \mathbb{E}[L_{v,p}] \qquad \forall p \tag{4.16}$$

CHAPTER 5. OPTIMIZATION FOR MEAN LATENCY AND LTP

In this section, we first formulate our optimization problem. Then, an efficient algorithm is proposed for solving our formulated problem, that aims to minimize the average latency and the LTP.

5.1 Optimization Problem for Mean Latency and LTP Trade-off

Let $q = \{q_{v,p},\ \forall v, p\}$, $t = \{t_v,\ \forall v\}$ and $\mathscr{S} = \{\mathscr{S}_p,\ \forall p\}$. We consider the following joint weighted mean latency and latency tail probability optimization problem, where the optimization is performed over the scheduling probabilities q, the placement of products $\mathscr{S}$ and auxiliary parameters t. Since this is a multi-objective optimization, the objective can be modeled as a convex combination of the two metrics. Let θ be the trade-off factor that determines the relative significance of tail latency and mean latency in the optimization problem, where $\theta \in [0,1]$. Further, let ω_p reflect the weight (or priority) of an order for product p. Hence, we can write our objective as follows:

$$\min \sum_{p=1}^{P} \sum_{v=1}^{V} \omega_p q_{v,p} \left[\theta \frac{(1-\rho_v) t_v M_{v,p}(t_v) e^{-t_v \delta_p}}{t_v - \Lambda_v (M_v(t_v) - 1)} + (1-\theta) \left(\frac{\Lambda_v \mathbb{E}[S_v^2]}{2(1 - \Lambda_v \mathbb{E}[S_v])} + \mathbb{E}[S_{v,p}] \right) \right] \tag{5.1}$$

$$\text{subject to} \quad \Lambda_v = \sum_{p=1}^{P} \lambda_p q_{v,p} \qquad \forall v \tag{5.2}$$

$$M_{v,p}(t_v) = \frac{\mu_v}{\mu_v - t_v} e^{\alpha_{v,p} t_v} \qquad \forall v, p \tag{5.3}$$

$$M_v(t_v) = \sum_{p=1}^{P} \frac{\lambda_p q_{v,p}}{\Lambda_v} M_{v,p}(t_v) \qquad \forall v \tag{5.4}$$

$$\mathbb{E}(S_v) = \sum_{p=1}^{P} \frac{\lambda_p q_{v,p}}{\Lambda_v} S_{v,p} \qquad \forall v \tag{5.5}$$

$$\mathbb{E}[S_v^2] = \sum_{p=1}^{P} \frac{\lambda_p q_{v,p} S_{v,p}^2}{\Lambda_v} \qquad \forall v \tag{5.6}$$

$$\mathbb{E}[S_{v,p}] = \sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{z=1}^{Z} a_{x,y,z,p} (\alpha_{v,p} + \frac{1}{\mu_v}) \qquad \forall v, p \tag{5.7}$$

$$\rho_v = \sum_{p=1}^{P} \lambda_p q_{v,p} \left(\alpha_{v,p} + \frac{1}{\mu_v} \right) \qquad \forall v \tag{5.8}$$

$$\sum_{v=1}^{V} q_{v,p} = 1 \qquad \forall p \tag{5.9}$$

$$\sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{z=1}^{Z} a_{x,y,z,p} = 1 \qquad \forall p \tag{5.10}$$

$$\sum_{p=1}^{P} a_{x,y,z,p} = 1 \qquad \forall x,y,z \tag{5.11}$$

$$q_{v,p} \in [0,1] \qquad \forall v,p \tag{5.12}$$

$$a_{x,y,z,p} \in \{0,1\} \qquad \forall x,y,z,p \tag{5.13}$$

$$\sum_{p=1}^{P} \sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{z=1}^{Z} a_{x,y,z,p} = P \tag{5.14}$$

$$\rho_v < 1 \qquad \forall v \tag{5.15}$$

$$t_v > 0 \qquad \forall v \tag{5.16}$$

$$t_v > \Lambda_v(M_v(t_v) - 1) \tag{5.17}$$

Constraint (5.2) gives the aggregate arrival rate Λ_v for each AGV under given scheduling $q_{v,p}$ and arrival rates λ_p, Constraints (5.3-5.6) define moment generating function with respect to parameter t_v, service time, and the second moment of the service time of AGV v. Constraint (5.7) gives the retrieval time of product p, if assigned to AGV v. Constraint (5.8) gives the traffic intensity of the AGVs, Constraints (5.9-5.14) guarantee that the scheduling and assignment decisions are feasible, and finally, the moment generating function exists due to the constraint in (5.17). Moreover, $\rho_v < 1$ is ensuring the stability of the warehouse system (i.e., queue length does not blow up to infinity under given arrival rates and scheduling probabilities).

Our policy prioritizes products according to their weights, so products with larger weights are prioritized more to further reduce the objective and thus optimizes the overall system. The placement of the products $\mathscr{S}$ helps in placing the highly prioritized products at closer locations thus reduces the objective. Note that the optimization over the auxiliary variables t gives a tighter bound on the weighted latency tail probability. Finally, tuning $\theta = 1$ to $\theta = 0$, the solution for (5.1) spans the solutions that minimize the LTP to the ones that minimize the mean latency of products orders. While we formulated our model generally, for our results section we have used $\theta = 0$ (strictly mean latency) or $\theta = 1$ (strictly LTP), since in practice only one of the two will be of interest.

Remark: The proposed optimization is non-convex, since constraint (5.17) is non-convex in (q, t). Further, the product placement ($\mathscr{S}$) has integer constraints.

Next, we develop an algorithmic solution by dividing the original problem into sub-problems that are easy to solve, then coming up with an efficient alternating optimization algorithm to solve it.

5.2 Proposed Algorithm

The joint mean and tail latency optimization problem given above is optimized over three sets of variables: scheduling probabilities q, product assignments $\mathscr{S}$, and auxiliary parameters t. Since the problem is non-convex, we propose an iterative algorithm to solve the problem. The proposed algorithm divides the main problem into sub-problems, which are easier to handle. The sub-problems are the following: Product assignment optimization which optimizes $\mathscr{S}$ for any given q and t; scheduling optimization which optimizes q for a given t and the optimal $\mathscr{S}$; and, auxiliary variables optimization which optimizes t for a given q and the optimal $\mathscr{S}$. This algorithm is summarized as follows.

1. **Initialization:** Initialize t and q in the feasible set.

2. **Find Optimal Product Assignment**: Develop an algorithm for finding the optimal product assignment.

3. **While Objective Converges**

   (a) Run Scheduling Optimization using current values of t and the optimal $\mathscr{S}$ to get new values of q.

   (b) Run Auxiliary Variables Optimization using current values of q and the optimal $\mathscr{S}$ to get new values of t.

Next, we will describe the sub-problems along with the proposed solutions for the sub-problems.

5.2.1 Product Assignment Optimization

To solve this problem we have provided an algorithm that works as follows. First, the products are sorted in descending order according to the ratio of (λ_p/ID_p) ratio, where ID_p is a product ID that determines how important/critical a product is. The smaller the product ID is, the more critical it is. Second, product with larger ratio are placed closer to the loading zone.

We prove that this placement algorithm is an optimal policy. Hence, we need only to iterate over the other 2 decision variables, i.e., q and t. To show this optimality, we use the concept of Adjacent Pairwise Interchange (API) on products. Since the proof of this theorem follows directly from the proof of Theorem 11.3.1 in [26], we refer the interested reader to page 301 of the book, for a detailed treatment of this.

To give a brief overview about the proof, we start by noting that there must be at least two adjacent products, say product j followed by product k, such that $\omega_j/T_j < \omega_k/T_k$, where T_j and T_k represents the latency for products j and k, respectively. Further, let $\omega_j > \omega_k$. From Eqn (4.7) and under probabilistic scheduling, we see that all products assigned for an AGV v experience, on average, the same waiting time (e.g., W_v) in the queue, however each product experiences different service time. This service time depends on two parts: random part $1/\mu_v$, and fixed/deterministic part $\alpha_{v,p}$ which depends on the product location in the warehouse. Since $1/\mu_v$ depends only on the AGV v and is independent of the product type, it is easily verified that assigning the products with higher weights closer to the loading zone (and thus lower retrieval time) will result in minimizing the overall weighted mean latency. This is because we can always express the retrieval time of product k as a function of the retrieval time of product j. Consider $D_{v,j,k}$= $\omega_j(W_v+S_{v,j})+\omega_k(W_v+S_{v,k})$=$\omega_j(W_v+S_{v,j})+\omega_k(W_v+S_{v,j}+\Delta_{j,k})$, where $\Delta_{j,k}$ represents the excess retrieval time of product k over product j. Since ω_j is strictly greater than w_k by definition, $D_{v,j,k}$ is minimized by reducing $S_{v,j}$, and thus our statement holds true.Another way of showing that the proposed placement algorithm is optimal is by noting that taking this ratio provides us with the relevant importance of each product. Thus, products with a higher ratio possess greater importance and intuitively should be placed in the nearest locations to the loading zone.

5.2.2 Scheduling Optimization

In order to solve this problem, we can use Successive Upper-Bound Minimization (SUM) algorithm or project gradient descent (PGD) algorithm [27]. The key idea of SUM algorithm is that the non-convex objective function is replaced by suitable convex approximations at which convergence to a stationary solution of the original non-convex optimization is established. SUM solves the approximated function efficiently and maintains feasibility in each iteration. However, in the context of our problem, PGD algorithm is used. While the SUM algorithm provides more accurate results than the PGD algorithm, it is complex and requires long computational times. The PGD, on the other hand, is much simpler with slightly less accuracy. Next, we present the pseudo-code and the optimization model for the PGD algorithm, where q is the decision variable:

Pseudo-code

```
Initialize $q^0$
For t from 0 to $T-1$ do:
$\rightarrow$ Compute the gradient $\nabla_q E(q^t)$
$\rightarrow$Take a step in the negative direction, $\tilde{q}^{t+1} = q^t - \gamma \nabla_q E$
$\rightarrow$Project $\tilde{q}^{t+1}_{i:\lambda}$ to the simplex $\triangle^L$ satisfying $\sum_{\lambda \in L} \tilde{q}_{i:\lambda} = 1$ and $0 \leqslant \tilde{q}_{i:\lambda} \leqslant 1$
$\rightarrow$$q^{t+1} = Proj_{\triangle} L(\tilde{q})$
end
Output :$q^{T-1}$
```

Optimization Model

$$\underset{q}{\text{minimize}} \quad f(q) \quad s.t. \quad q \in C$$

$$y_{k+1} = q_k - t_k \nabla f(q_k)$$

$$q_{k+1} = \underset{q \in C}{\text{argmin}} \|y_{k+1} - q\|$$

5.2.3 Auxiliary Variables Optimization

This sub-problem can be shown to be convex, and thus can be solved by the Projected Gradient Descent Algorithm with guaranteed (linear) convergence. We now show that this sub-problem is convex in t. We first note that inside the summations of the objective function (5.1), only the first term depends on a single value of t_v. Thus, it is enough to show that $\frac{t_v e^{-t_v \delta} M_v(t_v)}{t_v - \Lambda_v(M_v(t_v)-1)}$ is convex with respect to t_v. Since there is only a single index v here, we ignore this subscript for the rest of this proof. We denote

$$\begin{aligned} G(t) &= \frac{te^{-tx}M(t)}{t-\Lambda(M(t)-1)} && (5.18) \\ &= \frac{\alpha te^{(\beta-x)t}}{-t^2+(\alpha-\Lambda)t+\Lambda\alpha-\Lambda\alpha e^{\beta t}} && (5.19) \\ &= \frac{\alpha te^{(\beta-x)t}}{-t^2+(\alpha-\Lambda)t-\Lambda\alpha(e^{\beta t}-1)} && (5.20) \end{aligned}$$

Since the constraints in (5.2)-(5.17) are convex in t and the second derivative of $G(t)$ can be shown to be greater than zero, i.e., $G''(t) > 0$, the objective function is convex in t.

5.3 Proposed Algorithm Convergence

We first initialize $q_{v,p}$, $\mathscr{S}_p$ and t_v $\forall$ v, p , such that the choice is feasible for the problem. The t can be initialized to any value that is greater than zero (we have chosen an arbitrary value of 0.01), whereas the $q_{v,p}$ is initialized such that uniform assignment is achieved for each product among the AGVs. Then, we find the optimal $\mathscr{S}_p$, and do alternating minimization over $q_{v,p}$ and t_v. Since each sub-problem converges and the overall problem is bounded from below, the proposed algorithm converges to a stationary point.

CHAPTER 6. EXTENDED ANALYSIS FOR SCHEDULING

In this chapter, we provide closed-form expressions for the scheduling probabilities, under some simplified assumptions, regardless of the type of products. This will help us have a better understanding of how the system behaves for the proposed algorithm under a variety of system parameters. We aim to balance the load among all AGVs so that AGVs with higher speeds are expected to get larger portion of the total load. Hence, scheduling decisions are assumed to be source agnostic. Below, we focus only on minimizing the LTP by choosing the optimal decisions $(q_v, \forall v)$ so that the load is optimally (or near-optimally) distributed over all AGVs. From (4.11), we have

$$\begin{aligned}
\Pr(L_p \geq \delta) &\leq \sum_{v=1}^{V} \frac{q_v}{e^{t_v \delta}} \frac{(1-\rho_v)t_v}{t_v - \Lambda_v(M_v(t_v)-1)} \left(\frac{\mu_v e^{-\alpha_{p,v}}}{\mu_v - t_v} \right) \\
&\leq \sum_{v=1}^{V} \frac{(1-\rho_v)t_v M_{v,p}(t) e^{-t_v \delta}}{t_v - \Lambda_v(M_v(t_v)-1)} \\
&= \sum_{v=1}^{V} \frac{(1-\rho_v)F_v}{t_v - \Lambda_v R_{v,p}} \qquad (6.1)
\end{aligned}$$

where $F_v = t_v M_{v,p}(t) e^{-t_v \delta}$ and $R_v = (M_v(t_v)-1)$. Note that both $\mathbb{E}[S_v]$ and $M_v(t_v)$ are independent of q_v. To see that, we can write

$$\begin{aligned}
\mathbb{E}[S_v] &= \sum_{p=1}^{P} \frac{q_v \lambda_p}{\sum_p q_v \lambda_p} S_{v,p} = \sum_{p=1}^{P} \frac{\lambda_p}{\sum_p \lambda_p} S_{v,p} \\
&= \frac{1}{\lambda} \sum_{p=1}^{P} \lambda_p S_{v,p} = \frac{1}{\lambda} \overline{S}_{v,p} \qquad (6.2)
\end{aligned}$$

where $\lambda = \sum_{p=1}^{P} \lambda_p$ and $\overline{S}_{v,p} = \sum_{p=1}^{P} \lambda_p S_{v,p}$. Similarly, we can write

$$\begin{aligned}
M_v(t_v) &= \sum_{p=1}^{P} \frac{q_v \lambda_p}{\Lambda_v} \mathbb{E}\left[e^{t_v S_v} \right] \\
&= \sum_{p=1}^{P} \frac{q_v \lambda_p}{q_v \sum_{p=1}^{P} \lambda_p} M_v(t_v)
\end{aligned}$$

$$= \sum_{p=1}^{P} \frac{\lambda_p}{\lambda} M_v(t_v) \tag{6.3}$$

Without loss of generality, we consider two AGVs only, $V = 2$, and later on, we will generalize it to the scenarios where we have $V > 2$. Let $q_1 = q$ and hence $q_2 = 1 - q$. Hence, for $V = 2$, equation (6.1) reduces to

$$\Pr(L_p \geq \delta) \leq \frac{(1 - q\lambda\overline{S}_1)F_1}{t_1 - q\lambda R_1} + \frac{(1 - (1-q)\lambda\overline{S}_2)F_2}{t_2 - (1-q)\lambda R_2} \tag{6.4}$$

It is straightforward to prove that (6.4) is a convex function with respect to q. Therefore, to get the optimal value of q we differentiate and equate to zero. By $\frac{d\Pr(L_p \geq \delta)}{dq} = 0$ and simplifying the expressions, we get

$$\frac{d\Pr(L_p \geq \delta)}{dq} = \frac{F_1R_1\lambda - \lambda\overline{S}_1F_1t_1}{(t_1 - q\lambda R_1)^2} + \frac{\lambda\overline{S}_2F_2t_2 - F_2R_2\lambda}{(t_2 - (1-q)\,\lambda R_2)^2} \tag{6.5}$$

Hence,

$$\sqrt{\frac{F_1}{F_2}\,\frac{R_1 - \overline{S}_1t_1}{R_2 - \overline{S}_2t_2}} = \frac{t_1 - q\lambda R_1}{t_2 - (1-q)\,\lambda R_2} \tag{6.6}$$

By defining $C_{1,2}^{(q)} = \sqrt{\frac{F_1}{F_2}\frac{R_1 - \overline{S}_1t_1}{R_2 - \overline{S}_2t_2}}$, the optimal scheduling probability q^* (i.e., portion of load that goes to the first AGVs) can be written as

$$q^* = \frac{t_1 - C_{1,2}^{(p)}t_2 + C_{1,2}^{(p)}\lambda R_2}{C_{1,2}^{(p)}\lambda R_2 + \lambda R_1} \tag{6.7}$$

We can show that the above formula can be written for any number of AGV as follows

$$q_j^* = \frac{t_j - C_{j,-j}^{(p)}t_{-j} + C_{j,-j}^{(p)}\lambda R_{-j}}{C_{j,-j}^{(p)}\lambda R_{-j} + \lambda R_j} \tag{6.8}$$

where q^* represents the optimal scheduling probability to minimize the tail latency. We note that $C_{1,2}^{(q)}$ is always non-negative. Since F_1 and F_2 are non-negative, it is enough to show that $R_v > \overline{S}_v t_v$, for any v. To do so, we can write
$R_v = M_v(t_v) - 1 = \mathbb{E}[e^{t_v S_v}] - 1 > \mathbb{E}[1 + t_v S_v] - 1 = t_v \mathbb{E}[S_v] = t_v(\frac{1}{\mu_v})$. Hence, $C_{1,2}^{(q)} \geq 0$.

Note that for identical AGVs where $t_1 = t_2$, $R_1 = R_2$, $F_1 = F_2$, and hence $C_{1,2}^{(q)} = 1$, we have $q^* = 1/2$, which is intuitively expected.]

Next, for M/M/1 queues while focusing on the minimizing the mean latency, the following formula can be derived by taking the derivative of $L_{v,p}$ with respect to q and equating it to zero:

$$q^{**} = \frac{\lambda\sqrt{\mu_1} + \mu_1\sqrt{\mu_2} - \mu_2\sqrt{\mu_1}}{\lambda(\sqrt{\mu_1} + \sqrt{\mu_2})} \tag{6.9}$$

where q^{**} represents the optimal scheduling probability to minimize the average latency, $\mu_j, j = 1,2$, is the speed of the AGV and $\lambda = \sum_{p=1}^{P} \lambda_p$. For example, if $\mu_1 = \mu_2$, then $q^{**} = 1/2$ which is intuitively apparent. For general number of servers, (6.9) can be written as (for AGV j)

$$q_j^{**} = \frac{\lambda\sqrt{\mu_j} + \mu_j\left(\sum_{v=1}^{V}\sqrt{\mu_v}\right) - \sqrt{\mu_j}\left(\sum_{v=1}^{V}\mu_v\right)}{\lambda(\sum_{v=1}^{V}\sqrt{\mu_v})} \tag{6.10}$$

CHAPTER 7. NUMERICAL RESULTS

In this section, we evaluate our proposed algorithm for optimizing the two metrics of mean latency and LTP of products. Unless otherwise explicitly stated, we set the number of AGVs to 8, i.e., $V = 8$. The warehouse dimensions are assumed to be $X = 30$, $Y = 20$ and $Z = 2$. We further set $r = 0.01, m = 0.05, n = 0.01$. Moreover, we consider three different priority classes, where each product is assigned to one of the classes uniformly at random. Without loss of generality, a class of lower index is prioritized over those with higher index values (i.e., class 1 is the most important one while class 3 is the least important). The speeds of AGVs are shown in Table 7.1. The speeds are in m/s , and values were chosen to represent realistic speeds of AGVs in warehouses. Modelling the system with the heterogeneous speeds of the AGVs is very useful, and can be representative of real life situations, where a warehouse is transitioning to a new technology, which takes place gradually such that there is a mix between old AGVs with lower speeds and newer AGVs with faster speeds. Further, it can represent a hybrid warehouse system where humans transport loads along with the AGVs.

Table 7.1. *AGV Speeds*

AGV index	1	2	3	4	5	6	7	8
Speed (m/s)	2.15	1.50	1.90	2.25	1.80	2.20	1.90	2.35

We investigate our model under a wide range of parameters. While our simulation uses these specific parameters, our analysis and results remain applicable for any setting as long as the system maintains stable conditions under the chosen parameters.

Next we investigate our placement policy, scheduling policy and the convergence property of our algorithm.

As for the placement policy, keeping in mind that our storage policy is a dedicated one, we consider two baseline systems to compare with as described below:

- *Uniform Assignment (UA) Policy:* In this strategy, product types are assigned to random locations, regardless of how close or far they are to the loading zone. In other words, every product has en equal probability of being assigned to any of the locations, in the dedicated storage setting.

- *Turnover Storage (Greedy) Policy:* This policy assigns the product types with the highest arrival rates to locations that are closest to the loading zone. More details can be found in [8]

With respect to the scheduling policy, we compare our algorithm with the following baselines:

- *Join Shortest Queue (JSQ) Policy:* In this policy, the orders of products are assigned to the AGV with the least queue length. For detailed treatment of this policy, interested reader can refer to [11]

- *Least Work Load (LWL) Policy:* This policy assigns the incoming orders to the AGV that has the least (remaining) load (or processing time) among all the AGVs. Interested reader can refer to [12] for further details

- *Least Work Load-d LWL(d) Policy:* In this policy, a set of d AGVs are chosen at random and then orders are assigned to the AGV that has the least waiting time among those selected

- *Power-of-d Pow(d) Policy*: In this policy, a set of d AGVs are chosen at random and then orders are assigned to the AGV with the least queue length among those selected. In-detail description for this strategy can be found in [10]

- *Random Assignment (RA) Policy:* This policy assigns the scheduling in a random way

- *Proportional-service-rate Assignments (PA) Policy:* Orders of products are assigned in proportion to the service rates of the AGVs

7.1 Evaluation of Placement Algorithm

In this section, we evaluate our proposed placement algorithm and compare it with the baseline polices mentioned earlier.

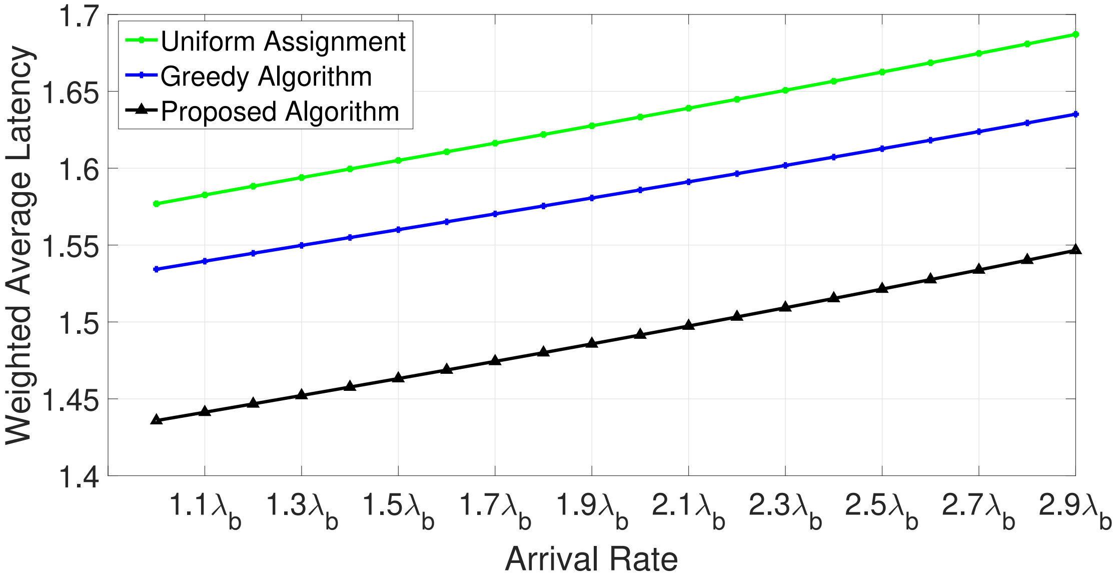

Figure 7.1. Effect of arrival rate on average weighted latency

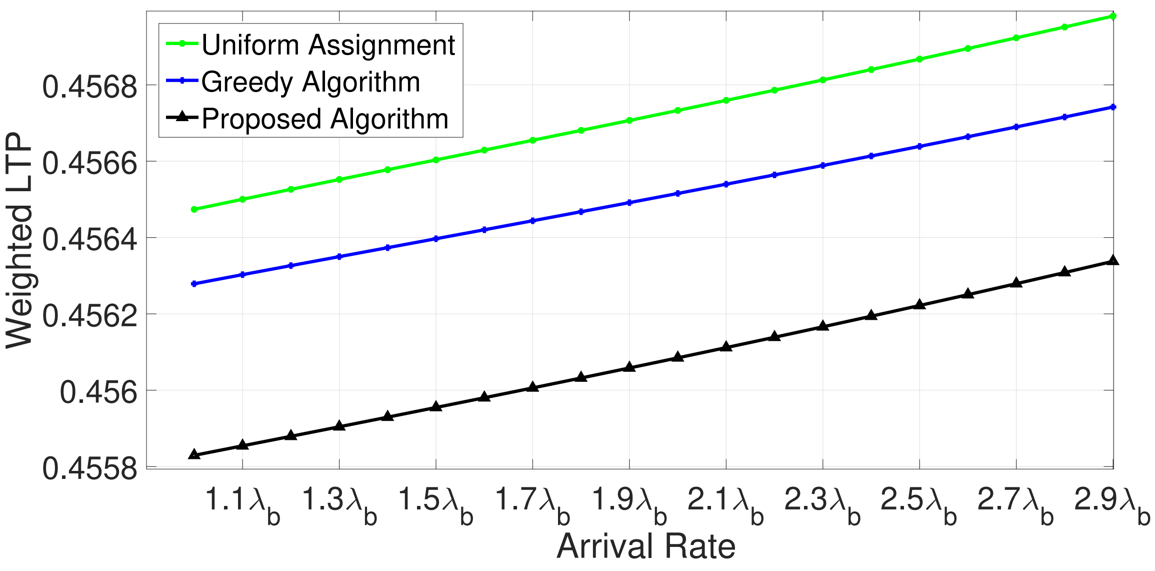

Figure 7.2. Effect of arrival rate on weighted LTP

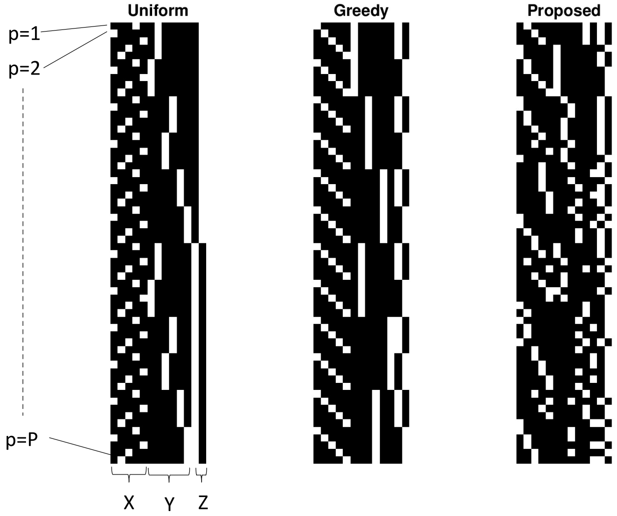

Figure 7.3. Visual representation of the placement of products in the warehouse. A black cell is corresponding to a ”zero” coordinate whereas a white cell is corresponding to a ”one” coordinate. For instance, for $p = 1$, the corresponding location is $(5,2,2)$ for the uniform assignment. However, this product $p = 1$ is stored at location $(1,1,1)$ for the Greedy policy, and in location $(2,5,1)$ for our optimized assignment policy.

Figures 7.1 and 7.2 show the average weighted latency and the weighted LTP for different arrival rates considering three placement policies. Further, a visual representation of the distribution of the product assignment in the warehouse is plotted in 7.3. To visualize the product placement and to better understand the effects of placement, we consider only 60 products in this particular experiment, with a warehouse dimensions of $X = 5, Y = 6, Z = 2$. Several arrival rates with multiple λ_b (where λ_b is the base arrival rate of product p) are also used. Further, we show the optimality of our policy for both the mean latency as well as the Latency Tail Probability. For the purpose of this, we use the optimal scheduling policy demonstrated in Chapter 6. We compare our placement policy with the uniform and greedy placements. In greedy placement, products with higher arrival rates are placed closer to the loading zone. We observe that our policy achieves the lowest latency as compared to the uniform placement and the greedy, for all arrival rates, for both the mean latency and the LTP. Further, our approach obtains 6.8%, 9.8% percentage improvement in average latency over the greedy algorithm and uniform assignment, respectively.

In figure 7.3, we plot a visualization view for the product placement in the warehouse under the three different policies. The X, Y, Z coordinates are spread out horizontally, forming a mesh of 60×13 (i.e., $60 \times (X+Y+Z)$) matrix. The first 5 cells represent the x-coordinate, while cells 6 to 11 represent the y-coordinate and the last two cells (cells 12 and 13) represent the z-coordinate. A black cell indicates that the coordinate is zero whereas a white cell has a value of 1. A certain location is determined by the tuple (x, y, z). For example, for the proposed policy, the first product is placed at $(2, 5, 1)$ while for the greedy policy the first product is assigned $(1, 1, 1)$. The aim of this figure is to provide a clear view of how the locations differ for the different proposed placement policies. Different from the greedy policy that takes into account only the arrival rate when assigning the products, our policy jointly considers more than one aspect when determining the products allocation, including the arrival rates, and the importance of the products.

7.2 Convergence of Algorithm

In this section, we will demonstrate the convergence of our proposed algorithms to a stationary point, with the aid of multiple graphs.

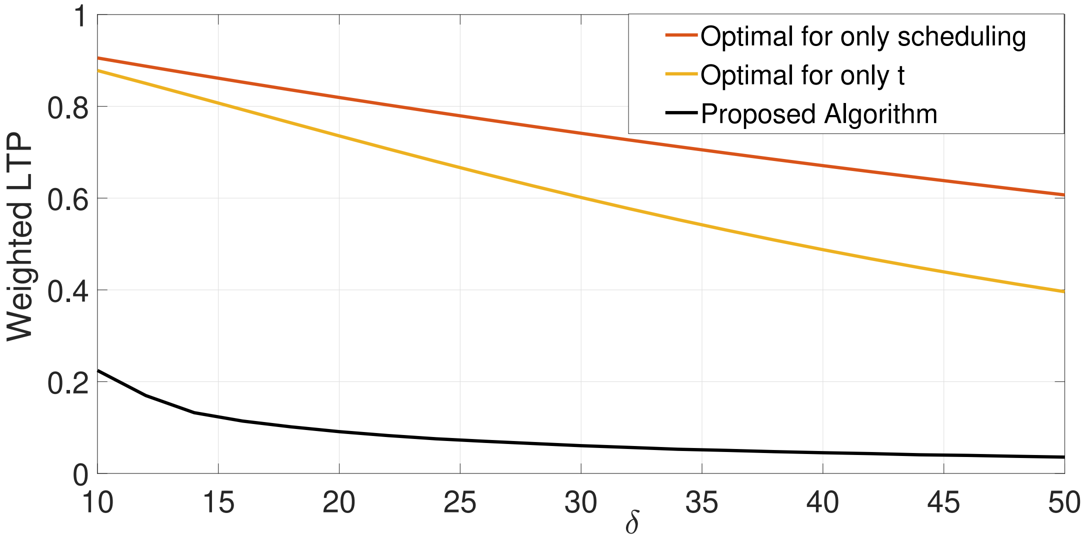

Figure 7.4. Demonstrating convergence of proposed algorithm by plotting δ versus weighted LTP

Figure 7.4 shows the behavior of weighted LTP versus the threshold δ (in seconds). We observe the behavior when only t is optimized, only q is optimized and when both are optimized (which is our proposed algorithm). Our approach finds the optimal weighted LTP by applying the alternative optimization algorithm over our control parameters: q and t, after the optimal placement has been plugged in. We note that this figure also represents the complementary cumulative distribution function (ccdf) of the aforementioned policies. For example, we observe that $\Pr(\delta \geq 20) \approx 0.1$ for our proposed policy which is significantly lower as compared to the other strategies.

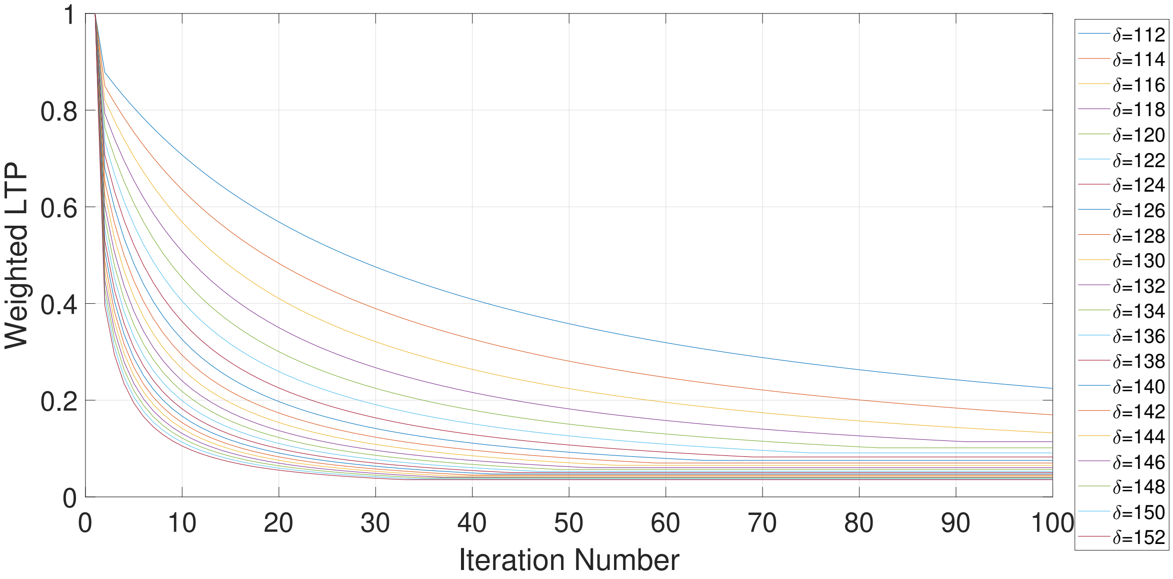

Figure 7.5. Demonstrating convergence of proposed algorithm by plotting iteration number versus weighted LTP

Figure 7.5 shows the convergence of our algorithm to a stationary point. It plots the weighted LTP for different values of δ, ranging from $\delta = 112$ to $\delta = 152$ with increments of 2, while we iterate over t and q with the optimal placement plugged in. This validates the efficiency of the proposed optimization algorithm.

7.3 Evaluation of Scheduling Algorithm

In this section, we evaluate our proposed algorithm for the scheduling, and compare it to some other baseline policies. For this purpose, we run simulations of the system in an online mode, meaning that the products arrive in real time and need to be dispatched on the fly into one of the available AGVs. The total time of our simulation is $T = 2000\ s$. For the following figures, we set $\lambda_p = \lambda_b/(p+1)$, where λ_b is the base arrival rate and is equal to 0.3. So all the next 3 figures use simulations to demonstrate the effectiveness of our scheduling algorithm.

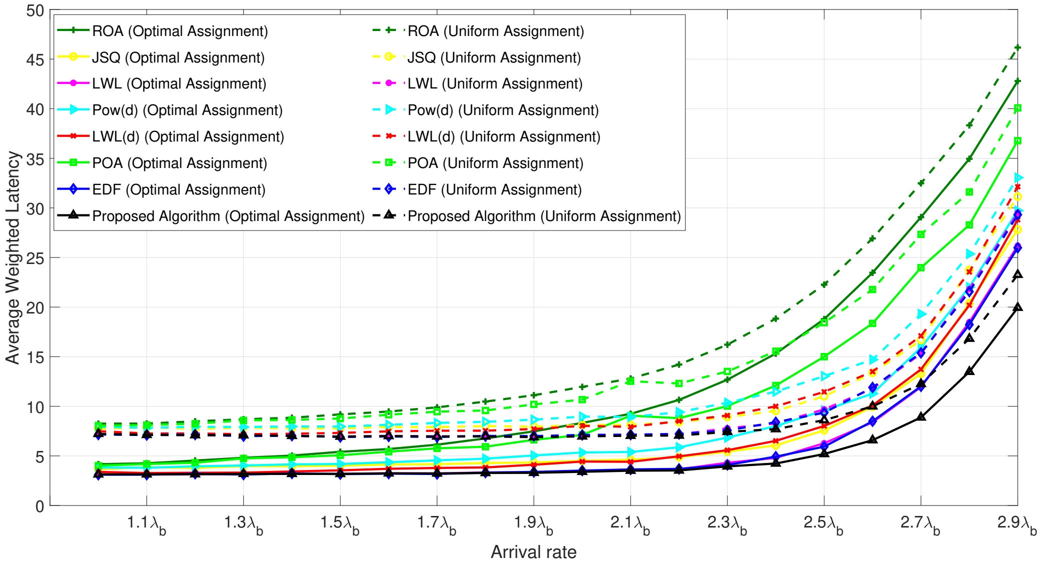

Figure 7.6. Arrival rates versus weighted average latency. Solid lines represent scheduling with the proposed optimal placement of products whereas dotted lines represent scheduling with a uniform placement

Effect of arrival rate of product orders on mean latency: Figure 7.6 shows the effect of increasing the product request rates from $1.0\lambda_b$ to $2.9\lambda_b$ with an increment step of 0.1. In this figure, we compare different scheduling strategies assuming optimal placement (solid lines) and random placement (dashed lines). We first observe that our proposed approach consistently performs the best among all considered approaches. In addition, at higher arrival rates, our approach still maintains low latency as compared to the most competitive baselines, i.e., LWL and JSQ. For instance, at the arrival rate of $\lambda_b = 2.9$, the proposed strategy reduces the latency by around 50% compared to JSQ, and by over 23% compared to the LWL policy. Further, our policy shows an improvement of around 33% as compared to the LWL without optimal placement. Note that, unlike queue-length-based scheduling where only the queue length counts, our policy differentiates among the different classes by prioritizing more the orders with higher weights/priority in order to minimize the overall latency.

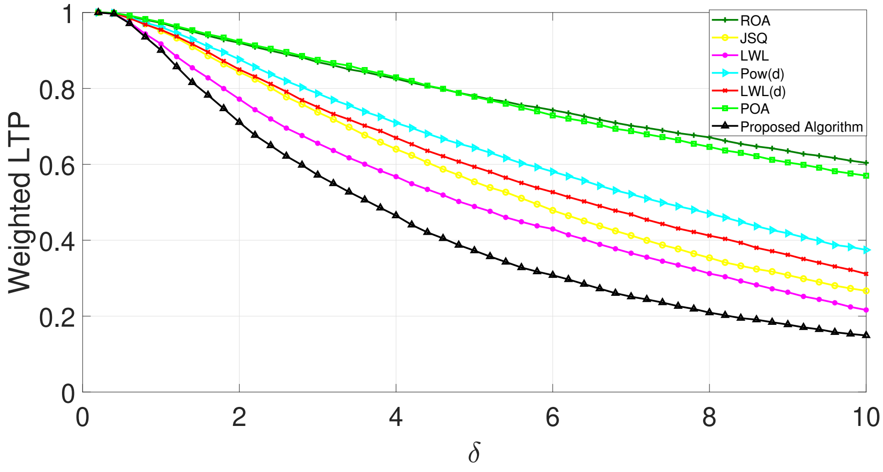

Figure 7.7. δ versus weighted LTP

Effect of varying the threshold for the latency: Figure 7.7 shows the effect of varying the threshold value on the probability of the average latency of products exceeding it. As observed, the greater the threshold value, the less the proportion of products whose latency surpass it. Further, our proposed algorithm consistently performs the best among others. This is because it utilizes the resources better and accounts for both classes and arrival rates. In our policy, higher-priority orders are prioritized, and thus their latency is minimized, resulting in an overall decrease of the weighted latency.

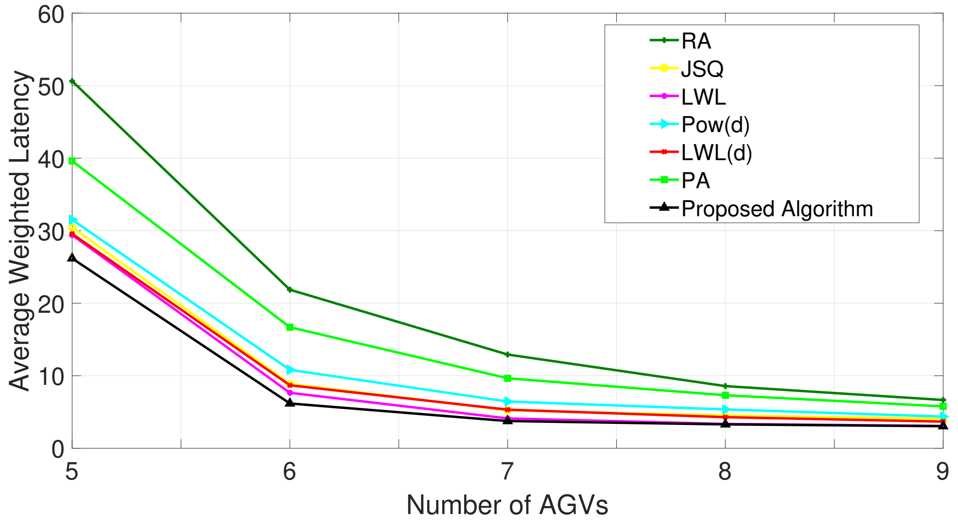

Figure 7.8. Weighted average latency for different number of AGVs

Effect of varying the number of AGVs on the mean latency: Figure 7.8 shows the effect of increasing the number of the AGVs from 5 to 8 with an increment step of 1. As expected, the average weighted latency decreases as the number of AGVs increases since more AGVs are available to serve the incoming online orders. Further, our policy shows an even better performance when the number of AGV is limited, with an 11% percentage improvement compared to the LWL policy. For a less constrained system (higher number of AGVs), our approach still performs best, with somewhat lower percentages. The major gain in this case, however, is the minimum complexity and the lack of need to track the AGV queues. Hence, overall our approach is better when it comes to both the performance and the computational cost.

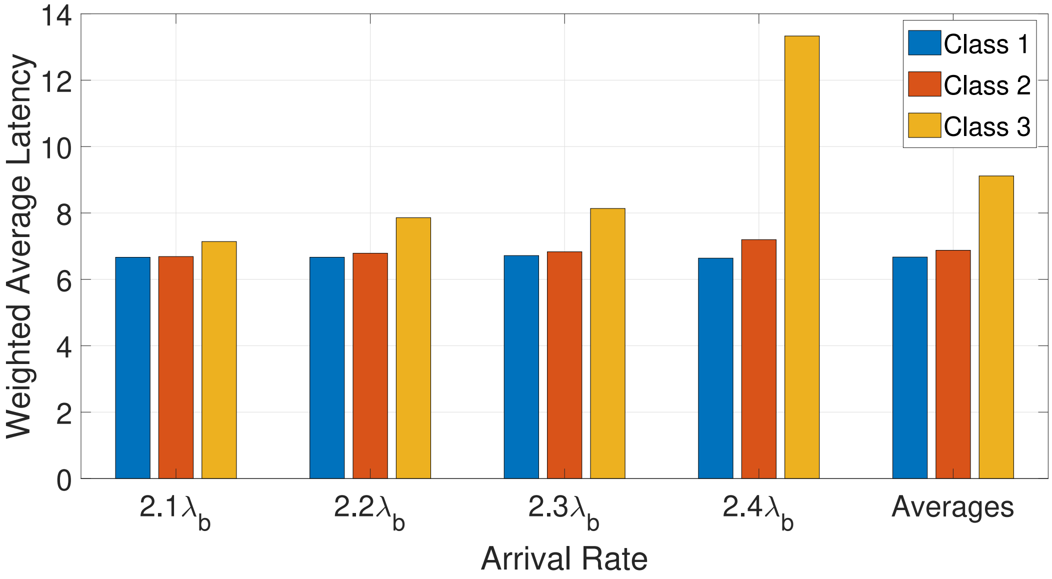

Figure 7.9. Weighted average latency for different arrival rates and classes

Figure 7.9 shows how the different classes of order requests experience different weighted average latency. Products are divided into 3 classes (blue bars represent the first class which is the most prioritized, red bars represent the second class, and yellow bars represents the third class which is the least prioritized). We vary the arrival rates of all products from 2.1λ to 2.4λ and plot the weighted average latency for each group. While weighted mean latency increases as arrival rate increases, our algorithm assigns differentiated latency for different product sets. Class 1 always receives the minimum average latency. Hence, efficiently reducing the latency of the high arrival rate products reduces the overall weighted average latency.

CHAPTER 8. CONCLUSIONS AND FURTURE WORK

We carry out extensive simulations to study the performance of the developed strategy. The system performance is measured by different objectives including the overall weighted sum of mean latency and LTP as well as the trade-off between them. We observe that the higher the difference in priority classes is, the more effective our policy gets. Further, at heavy loads of customer orders and low number of AGVs, our approach achieves a higher percentage system improvement. Further, for a less constrained system, our approach has the advantage of low computational cost. Moreover, joint optimization of storage assignment and LTP results in greater gains than optimizing only one parameter at a time. Hence, our framework gives important design guidelines for designing smart warehouses to provide the desired service to the customers.

Our future work can include an end-to-end delivery to customer houses using drones, with the inclusion of the cost of delivery. An addition that can also be introduced is a warehouse model where the same product type is stored in different locations, and thus one or more AGV(s) can be assigned to serve a particular order. Further, more than one warehouse can be considered to serve the same request. Also, non-uniform unit loads along with AGVs with varying capacities can be considered. Finally, our approach can be extended to include varying sizes of the racks, with more added constraints to ensure products fit on desired shelves.

REFERENCES

[1] Edward Frazelle. *Supply chain strategy: the logistics of supply chain management*. McGrraw Hill, 2002.

[2] Alan Harrison and Remko I Van Hoek. *Logistics management and strategy: competing through the supply chain*. Pearson Education, 2008.

[3] Peter Baker and Marco Canessa. Warehouse design: A structured approach. *European Journal of Operational Research*, 193(2):425–436, 2009.

[4] René de Koster, Tho Le-Duc, and Kees Jan Roodbergen. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2):481–501, 2007.

[5] Charles G. Petersen and Gerald Aase. A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*, 92(1):11–19, 2004.

[6] Venkata Reddy Muppani (Muppant) and Gajendra Kumar Adil. A branch and bound algorithm for class based storage location assignment. *European Journal of Operational Research*, 189(2):492–507, 2008.

[7] Mengfei Yu and René B.M. de Koster. The impact of order batching and picking area zoning on order picking system performance. *European Journal of Operational Research*, 198(2):480–490, 2009.

[8] Yugang Yu and René B.M. De Koster. On the suboptimality of full turnover-based storage. *International Journal of Production Research*, 51(6):1635–1647, 2013.

[9] Jiaxi Li, Mohsen Moghaddam, and Shimon Y Nof. Dynamic storage assignment with product affinity and ABC classification—a case study. *The International Journal of Advanced Manufacturing Technology*, 84(9):2179–2194, 2016.

[10] Nikita Dmitrievna Vvedenskaya, Roland L'vovich Dobrushin, and Fridrikh Izrailevich Karpelevich. Queueing system with selection of the shortest of two queues: An asymptotic approach. *Problemy Peredachi Informatsii*, 32(1):20–34, 1996.

[11] Hwa Chun Lin and C S Raghavendra. An analysis of the join the shortest queue (JSQ) policy. In *Proceedings - International Conference on Distributed Computing Systems*, volume 1992-June, pages 362–366, 1992.

[12] Debankur Mukherjee. *Scalable Load Balancing Algorithms in Networked Systems*. 2018.

[13] Guy De Tre, Axel Hallez, and Antoon Bronselaer. Performance optimization of object comparison. *International Journal of intelligent Systems*, 29(2):495–524, 2014.

[14] Daria Battini, Martina Calzavara, Alessandro Persona, and Fabio Sgarbossa. Order picking system design: The storage assignment and travel distance estimation (SA&TDE) joint method. *International Journal of Production Research*, 53(4):1077–1093, 2015.

[15] Ling Feng Hsieh and Yi Chen Huang. New batch construction heuristics to optimise the performance of order picking systems. *International Journal of Production Economics*, 131(2):618–630, 2011.

[16] Wing Yan Hung, Nouri J. Samsatli, and Nilay Shah. Object-oriented dynamic supply-chain modelling incorporated with production scheduling. *European Journal of Operational Research*, 169(3):1064–1076, 2006.

[17] Md E. Haque, Sameh Elnikety, Yong Hun Eom, Ricardo Bianchin, Yuxiong He, and Kathryn S. McKinley. Few-to-Many: Incremental parallelism for reducing tail latency in interactive services. *ACM SIGPLAN Notices*, 50(4):161–175, 2015.

[18] Leif Gustafsson. Poisson Simulation as an Extension of Continuous System Simulation for the Modeling of Queuing Systems. *Simulation*, 79(9):528–541, 2003.

[19] Venkitasubramony Rakesh and Gajendra K. Adil. Layout Optimization of a Three Dimensional Order Picking Warehouse. *IFAC-PapersOnLine*, 28(3):1155–1160, 2015.

[20] Randolph W. Hall. Distance approximations for routing manual pickers in a warehouse. *IIE Transactions (Institute of Industrial Engineers)*, 25(4):76–87, 1993.

[21] Ek Peng Chew and Loon Ching Tang. Travel time analysis for general item location assignment in a rectangular warehouse. *European Journal of Operational Research*, 112(3):582–597, 1999.

[22] Luis F. Cardona, Diego F. Soto, Leonardo Rivera, and Hector J. Martínez. Detailed design of fishbone warehouse layouts with vertical travel. *International Journal of Production Economics*, 170:825–837, 2015.

[23] Kees Jan Roodbergen, Gunter P. Sharp, and Iris F.A. Vis. Designing the layout structure of manual order picking areas in warehouses. *IIE Transactions (Institute of Industrial Engineers)*, 40(11):1032–1045, 2008.

[24] Sheldon Ross. *A first course in probability*. Pearson, 2014.

[25] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

[26] M L Pinedo. Scheduling: theory, algorithms, and systems. 5-th ed. Cham, 2016.

[27] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013.